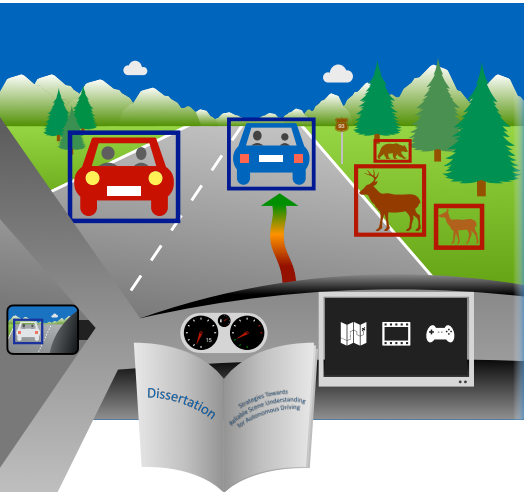




Dissertation

Strategies Towards Reliable Scene Understanding for Autonomous Driving

Stefano Gasperini





Technische Universität München
TUM School of Computation, Information and Technology

Strategies Towards Reliable Scene Understanding for Autonomous Driving

Stefano Gasperini

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Julia Schnabel

Prüfende der Dissertation: 1. Priv.-Doz. Dr. Federico Tombari

2. Prof. Dr. Daniel Cremers

3. Prof. Dr. Bastian Leibe

Die Dissertation wurde am 21.12.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 04.11.2024 angenommen.

Stefano Gasperini

Strategies Towards Reliable Scene Understanding for Autonomous Driving

Dissertation, Version 1.0

Technical University of Munich

TUM School of Computation, Information and Technology

Chair for Computer Aided Medical Procedures

Boltzmannstraße 3

85748 Garching by Munich

Abstract

Autonomous driving has gained significant attention thanks to its potential to revolutionize transportation, offering safety, efficiency, and comfort benefits. Despite substantial progress and the achievement of important milestones such as autonomous rides in geofenced areas, the widespread availability of self-driving cars to the public remains a future goal. This is mainly due to unresolved technical and regulatory challenges, as well as issues around public acceptance.

A key technical challenge in autonomous driving is scene understanding, which is crucial for vehicles to accurately perceive and interpret their surroundings. This computer vision task is made complex by the unpredictability and diversity of real-world scenarios. Since self-driving cars aim to improve safety, reliability is of utmost importance. In this context, reliability translates into consistent performance under varying conditions, including familiar and new environments, different times of day, and weather, while managing sensor artifacts and dynamic, unpredictable elements. However, most works focus on improving performance in well-known and defined settings, e.g., in the daytime with good visibility and known objects, ignoring corner cases and challenging conditions.

This dissertation addresses these challenges through several publications, aiming to enhance the reliability of autonomous driving perception systems. The work investigates reliability from various perspectives in 2D and 3D settings and presents a range of innovative techniques for improving the performance and trustworthiness of state-of-the-art models in both typical and extreme conditions. Such research enhances reliability via output consistency across sub-tasks, generalization to out-of-distribution data and unseen, unknown objects, and robustness against adverse and dynamic conditions. Specifically, it proposes an end-to-end method for panoptic segmentation and domain generalization approaches based on uncertainty estimation and plausible adversarial augmentations. Generalization is then taken to the extreme, enabling a model to segment known regions and unseen objects of completely unknown categories. The dissertation also tackles specific systematic issues in monocular depth estimation, such as depth prediction for dynamic objects and in adverse weather and illumination conditions, offering novel solutions such as weak radar supervision or learned common representations across easy and unfavorable scenarios.

In summary, by uncovering and addressing critical challenges in scene understanding through diverse methodologies, this research provides valuable insights and novel methods, raising awareness of the importance of tackling reliability issues as a necessary step to full autonomy. Therefore, this dissertation represents a step towards safer and more reliable perception systems for autonomous driving, contributing to the ongoing global efforts in this field.

Zusammenfassung

Autonomes Fahren hat aufgrund seines Potenzials, den Transport zu revolutionieren, erhebliches Interesse geweckt, da es Vorteile in den Bereichen Sicherheit, Effizienz und Komfort bietet. Trotz beträchtlicher Fortschritte und wichtiger Meilensteine, wie autonomer Fahrten in abgegrenzten Bereichen, bleibt die flächendeckende Verfügbarkeit selbstfahrender Autos für die Öffentlichkeit ein Ziel der Zukunft. Dies liegt hauptsächlich an ungelösten technischen und regulatorischen Herausforderungen sowie an Fragen der öffentlichen Akzeptanz.

Eine zentrale technische Herausforderung im autonomen Fahren ist das Szenenverständnis, das für eine präzise Wahrnehmung und Interpretation der Umgebung durch Fahrzeuge entscheidend ist. Diese Aufgabe im Bereich des maschinellen Sehens wird durch die Unvorhersehbarkeit und Vielfalt realer Szenarien erschwert. Da selbstfahrende Autos die Sicherheit verbessern sollen, ist Zuverlässigkeit von größter Bedeutung. In diesem Zusammenhang bedeutet Zuverlässigkeit eine konsistente Leistung unter unterschiedlichen Bedingungen, einschließlich bekannter und neuer Umgebungen, verschiedener Tageszeiten und Wetterbedingungen, während gleichzeitig Sensorartefakte sowie dynamische und unvorhersehbare Elemente bewältigt werden. Die meisten Arbeiten konzentrieren sich jedoch auf die Leistungssteigerung in gut bekannten und definierten Szenarien, z. B. tagsüber bei guter Sicht und bekannten Objekten, während Randfälle und herausfordernde Bedingungen oft vernachlässigt werden.

Diese Dissertation adressiert diese Herausforderungen durch mehrere Publikationen mit dem Ziel, die Zuverlässigkeit von Wahrnehmungssystemen für autonomes Fahren zu verbessern. Die Arbeit untersucht Zuverlässigkeit aus verschiedenen Perspektiven in 2D- und 3D-Settings und stellt eine Reihe innovativer Techniken zur Leistungssteigerung und Vertrauenswürdigkeit moderner Modelle sowohl unter typischen als auch unter extremen Bedingungen vor. Die Zuverlässigkeit wird dabei durch Konsistenz der Ausgabe über verschiedene Teilaufgaben hinweg, Generalisierung auf Out-of-Distribution-Daten und unbekannte Objekte sowie Robustheit gegenüber widrigen und dynamischen Bedingungen verbessert. Konkret wird eine End-to-End-Methode für panoptische Segmentierung sowie Ansätze zur Domänengeneralisierung basierend auf Unsicherheitsabschätzung und plausiblen adversarialen Augmentierungen vorgeschlagen. Die Generalisierung wird dann auf die Spitze getrieben, sodass ein Modell bekannte Regionen und unbekannte Objekte völlig unbekannter Kategorien segmentieren kann. Darüber hinaus werden systematische Probleme in der monokularen Tiefenschätzung untersucht, wie beispielsweise die Tiefenvorhersage für dynamische Objekte und unter widrigen Wetter- und Lichtverhältnissen. Hierfür werden neuartige Lösungen vorgestellt, darunter schwache Radar-Supervision oder gelernte gemeinsame Repräsentationen zwischen einfachen und herausfordernden Szenarien.

Zusammenfassend bietet diese Forschung durch die Identifikation und Bearbeitung zentraler Herausforderungen im Szenenverständnis wertvolle Erkenntnisse und neuartige Methoden und macht auf

die Notwendigkeit aufmerksam, Zuverlässigkeitsprobleme als einen essenziellen Schritt hin zur vollständigen Autonomie zu adressieren. Damit stellt diese Dissertation einen Beitrag zu sichereren und zuverlässigeren Wahrnehmungssystemen für das autonome Fahren dar und unterstützt die laufenden globalen Bemühungen in diesem Bereich.

Acknowledgments

This PhD has been a fantastic journey of personal and professional growth. Although there is my name on the cover page of this dissertation, this is the result of the direct and indirect work and support of many people around me. The amount of challenges and stress during these PhD years has been significant (including covid), but I was privileged to have great people by my side to support me by helping me tackle the issues or by distracting me from them. Therefore, thank you to everyone who has been there for me and made this time so unique and memorable!

In particular, I want to thank my supervisor, Federico, who gave me this opportunity in the first place and supported me until the end. I remember when you told me BMW would consider your suggested candidate for their open PhD position. My confusion turned into joy as I realized I was that candidate. Since then, you have become a mentor, a role model, and a friend. I always look forward to sharing time with you in Munich and at the conferences!

Special thanks also to Nassir for creating the one-and-only CAMP and welcoming me to the group, Mohammad, Alvaro, and later Michael for hiring me at BMW for this PhD and supporting me and my work there. To you guys and Federico, thank you for giving me freedom and letting me define my research direction. I really appreciate it; it made me stronger and more passionate about my work.

I owe so much to my co-authors for sharing the projects with me and contributing firsthand to my PhD. The heroes who worked with me through the sleepless nights before the classic 9 a.m. CET deadlines are Alex, Jan, Nils, and Patrick K. In particular, thank you, Alex, for always saying, "Let's do it!" with every experiment I suggested! Thank you, Jan, for one of the wildest adventures! Thank you, Nils, for not giving up even when nothing worked! And thank you, Patrick, for excellently running on cold and partly uncooked rice for weeks. It has been an absolute pleasure to work with you and celebrate the papers' successes together!

Huge thank you to my closest PhD colleagues who quickly became awesome friends and shared this journey and other trips with me: Alex, HyunJun, Lennart, and Shun-Cheng. It has been an absolute pleasure to have you along for joking around, having serious work and life conversations, releasing the pressure, and making these years genuinely fun! Thank you so much!

From CAMP, I want to thank also Magda P for introducing me to research and CAMP itself, giving me a glimpse of PhD already during my Master's, which helped me a lot later during my PhD. A big thank you to the many people at CAMP who have shared slightly or fully their PhD journey with me! Thank you for the interesting discussions and fun conversations at all the lunches, dinners, conferences, CAMPings, meetings, and breaks we shared in the lab and outside. Thank you Alberto, Artem, Ashkan,

Chantal, Dani, Ege, Evin, Felix D, Felix H, Felix T, Guangyao, Heiko, Kamilia, Kristina, Kunyi, Mahdi, Maria, Markus, Mert, Michael, Michela, Niko, Salvo, Sen, Shervin, Tianyu, Tobias, Yanyan, Yuan, Zhongliang, and who joined after I submitted, too: Andrea, Changxuan, Christian, Diego, Giulia, Klara, Magda W, Miriam, and Miruna. Good luck to all of you working on your PhD! Whether you have a few days or years to go - you got this! Thank you, Fabian, Helisa, and Johanna, for your insights and for showing what an excellent PhD looks like. Many thanks to Niko for helping the team left and right, to our seniors Shahrooz, Slobodan, and Ulrich for their positivity and insights, to Amir for the IT support, and to Martina and Leopold for keeping the Chair running.

Big thank you to everyone who embarked on their Master's thesis under my supervision, in particular, Christopher, Frithjof, Jan, Nils, Patrick K, and Tarlan, for growing with me, making me happy as I saw your progress, all the interesting discussions we had, and for directly or indirectly contributing to shaping my PhD and research directions.

Thanks a lot also to my PhD committee, Federico, Daniel Cremers, and Bastian Leibe for reading through so many pages and appreciating my contributions to the field. Thank you to Julia Schnabel and Sabine for the organization.

From my colleagues at BMW, I want to thank also Barbara, Loren, Naveen, Nils B, Sebastian, Thomas, Tim, and Vinzenz for the interesting discussions and their support, and the fellow PhDs of the ProMotion program, Andrea, Aral, Chantal, Giada, Gonzalo, Jiajie, Johannes, Kai, Lukas, Manoj, Maren, Martin, Pia, Robert, Sebastian, and Thomas, for the fun times and the cool perspectives on the automotive world.

Many thanks also to my friends and important people in my life outside the lab in Munich, Bologna, and around the world, who rooted for me and were always there to support or distract me: Alessandro, Alessio, Andrea, Camilla, Carlos, Chiara, Christel, Claudia D, Claudia G, Eunah, Fanny, Farida, Filippas, Francesco C, Francesco F, Francesco P, Gemma, Giacomo, Ignacio, Jin, Johanna, Juliana, Kostas, Krystal, Luca, Magda, Mark, Maria Clara, Marie, Markéta, Michele, Nicola, Riccardo, Roberta, Simona, Tommaso, and William. Thank you so much!

Last but not least, I want to thank my family for their unconditional support and affection and for believing in me throughout these years! It means a lot!

Thank you!

Contents

List of Authored and Co-authored Publications	1
I Introduction	3
1 Introduction	5
1.1 Motivation and Main Objective	5
1.2 Structure of this Dissertation	6
2 Autonomous Driving	9
2.1 The 6 Levels of Autonomy	9
2.2 The Benefits	10
2.3 Brief History	11
2.4 Societal Response	14
2.5 Impact on Regulations	15
2.6 Autonomous Driving Stack	16
3 Scene Understanding	19
3.1 Overview	19
3.2 Applications	20
3.2.1 Robotics	20
3.2.2 Healthcare	21
3.2.3 Augmented Reality	21
3.3 Input Signals for Autonomous Driving	22
3.4 Perception Tasks for Autonomous Driving	23
3.5 Challenges in Scene Understanding for Autonomous Driving	25
II Methodology	27
4 Reliability	29
4.1 Overview	29
4.2 The Facets of Reliability	30
4.3 Robustness and Generalization	31
4.3.1 The Importance of Generalization	31
4.3.2 The Importance of Robustness	32
4.3.3 The Robustness-Generalization Trade-off	32
4.4 The Impact of Data	33
4.4.1 Data Biases	34
4.5 Challenging Conditions	35

4.6	Out-of-Distribution Data and Corner Cases	36
4.6.1	Unknown Objects	37
4.7	Uncertainty Estimation	38
4.8	Domain Adaptation and Domain Generalization	39
4.9	Adversarial Methods	40
5	Summary of Contributions	43
5.1	Consistent Scene Understanding	43
5.1.1	End-to-End Panoptic Segmentation (RA-L 2021)	44
5.2	Generalizing to Out-of-Distribution Data	57
5.2.1	Domain Generalization with Uncertainty Estimation (RA-L 2021)	57
5.2.2	Domain Generalization with Adversarial Augmentations (CVPR 2022)	69
5.2.3	Segmentation of Known and Unknown Objects and Regions (ICCV 2023)	83
5.3	Improving Robustness for Depth Estimation	101
5.3.1	Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)	101
5.3.2	Monocular Depth Estimation in Challenging Conditions (ICCV 2023)	115
III	Conclusion and Outlook	131
6	Summary and Findings	133
7	Future Work and Discussion	135
7.1	Methods to Make Models More Reliable	135
7.1.1	Specific Improvements to the Proposed Methods	135
7.1.2	Foundation Models	137
7.1.3	Best Use of Reliability Information	138
7.1.4	Temporal Models and End-to-End Frameworks	139
7.2	Data Generation and Processing	139
7.2.1	Adversarial Methods for Data Generation	139
7.2.2	Scene Manipulation	140
7.2.3	Synthetic Data	141
IV	Appendix	143
A	Abstracts of Publications not Discussed in this Thesis	145
	3D Adversarial Augmentations for Domain Generalization (IJCV 2023)	147
B	Supplementary Materials of the Publications	149
B.1	Appendix: Domain Generalization with Adversarial Augmentations (CVPR 2022)	149
B.2	Appendix: Segmenting Known and Unknown Objects and Regions (ICCV 2023)	161
B.3	Appendix: Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)	177
B.4	Appendix: Monocular Depth Estimation in Challenging Conditions (ICCV 2023)	185
C	Licenses of the Publications	211
	Bibliography	219
	List of Figures	246

List of Authored and Co-authored Publications

2023

- [85] **S. Gasperini**, A. Marcos-Ramiro, M. Schmidt, N. Navab, B. Busam, and F. Tombari. "Segmenting Known Objects and Unseen Unknowns without Prior Knowledge". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. October, 2023.
- [86] **S. Gasperini***, N. Morbitzer*, HJ. Jung, N. Navab, and F. Tombari. "Robust Monocular Depth Estimation under Challenging Conditions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. October, 2023.
- [163] A. Lehner*, **S. Gasperini***, A. Marcos-Ramiro, M. Schmidt, N. Navab, B. Busam, and F. Tombari. "3D Adversarial Augmentations for Robust Out-of-Domain Predictions". In: *International Journal of Computer Vision (IJCV)*. Published in October 2023 and included in Volume 132, Number 3 of March 2024.

2022

- [162] A. Lehner*, **S. Gasperini***, A. Marcos-Ramiro, M. Schmidt, M.-A. Nikouei Mahani, N. Navab, B. Busam, and F. Tombari. "3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June, 2022.

2021

- [84] **S. Gasperini**, M.-A. Nikouei Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. "Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds". In: *IEEE Robotics and Automation Letters (RA-L)*. February, 2021. Also accepted and presented at: *IEEE International Conference on Robotics and Automation (ICRA)*. May, 2021.
- [82] **S. Gasperini***, J. Haug*, M.-A. Nikouei Mahani, A. Marcos-Ramiro, N. Navab, B. Busam, and F. Tombari. "CertainNet: Sampling-Free Uncertainty Estimation for Object Detection". In:

IEEE Robotics and Automation Letters (RA-L). November, 2021. Also accepted and presented at: *IEEE International Conference on Robotics and Automation (ICRA)*. May, 2022.

- [83] **S. Gasperini***, P. Koch*, V. Dallabetta, N. Navab, B. Busam, and F. Tombari. "R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes". In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. December, 2021.

The symbol * indicates that the authors contributed equally.

Part I

Introduction

Introduction



Figure 1.1. Representations of autonomous vehicles created with DALL·E 3 [23]. The left image shows a fictional autonomous car in the streets of Munich, Germany, equipped with perception sensors such as a LiDAR on the roof. As depicted on the right side, the high autonomy enables all passengers to focus on activities such as working or reading as the vehicle drives itself.

1.1 Motivation and Main Objective

In recent years, autonomous driving has emerged as a promising technology with the potential to revolutionize transportation systems and redefine our relationship with vehicles. Various factors, ranging from safety and efficiency to comfort, have driven the advancements of this technology.

In a futuristic, utopian scenario, vehicles would lack human drivers, be interconnected, and always make optimal decisions. Such a reality promises to majorly reduce road accidents, improve traffic flow and efficiency, reduce the environmental impact of transportation (e.g., by mitigating congestion), and broaden mobility options for people who cannot drive (e.g., children) and even pets. As depicted in Figure 1.1, it could also significantly increase productivity and comfort by enabling multitasking during commutes and encouraging the adoption of on-demand mobility as a service model.

Over the past few decades, the introduction of safety features such as seat belts and airbags has substantially reduced the number of road fatalities. Specifically, since 1970 in Germany, traffic-related deaths have dropped by a factor of seven, reaching approximately 3,000 annually [335]. Yet, road accidents still result in around 1.24 million fatalities globally every year [197], standing as one of the primary causes of death [205]. Autonomous vehicles promise to drastically reduce this number by mitigating human errors, which are the primary contributors to road accidents.

The work behind this dissertation began during a period characterized by immense hype and optimism toward autonomous driving. Early estimates fueled a *race to autonomy*, with diverse stakeholders, including car manufacturers, tech companies, startups, and component suppliers, jumping into the race. This led to significant investments and high expectations among the general public. However,

the challenges in achieving full autonomy have been greater than initially estimated, significantly extending development timelines and increasing costs. This caused a general disillusionment towards the field and a recalibration of objectives by various players, with some focusing more on assisted driving and others altogether abandoning the race. Nevertheless, the benefits of this technology are substantial, so its development has continued.

The progress achieved by the scientific community has been tremendous, reaching significant milestones such as the offer of autonomous rides to selected users [315]. However, substantial challenges remain before autonomous vehicles can safely and reliably operate in the unconstrained real world [147]. Among these challenges is the robust perception of the vehicle's environment through scene understanding. Characterized by accurately perceiving and understanding traffic scenes in real-time, even under challenging, adverse, and dynamic conditions, this ability is essential for autonomous vehicles to navigate while ensuring the safety of their passengers and other traffic participants.

Even though state-of-the-art approaches reached astonishing results on popular benchmarks [35, 90], this does not indicate absolute performance because the same benchmarks cannot evaluate a method under all possible situations. Due to the difficulty of collecting corner cases as they happen in real life, current datasets fail to fully represent the world's diversity. For these reasons, the performance of state-of-the-art methods deteriorates significantly with novel or unexpected scenarios that fall outside their training data distribution. Therefore, deploying state-of-the-art approaches directly to real-world applications raises issues in safety-critical settings, such as autonomous driving. Unlike in other domains, such as mobile applications and chatbots, where dangers are mainly related to the spread of misinformation, in safety-critical scenarios, ensuring reliability is fundamental to mitigating potential hazards that could endanger people's lives.

Particularly concerning are systematic errors exhibited by state-of-the-art approaches whenever their underlying assumptions are violated. These include conflicts among the output of sub-tasks [84], the handling of unknown objects for segmentation models [85], and that of dynamic objects [83] or challenging illumination and weather conditions [86] for self-supervised monocular depth estimators. Specifically, unknown objects are wrongly associated with the limited set of known classes, while low visibility is detrimental to learning to estimate depth as it prevents establishing correct pixel correspondences. Moreover, the depth of dynamic objects is wrongly estimated due to their movement within sequences of images, which are assumed static during training.

This dissertation aims to mitigate these safety-critical issues in scene understanding by designing new techniques that increase the models' reliability. Thus, this thesis addresses the applicability of state-of-the-art methods in challenging scenarios, taking a step towards the futuristic and utopian driver-less setting described above. While most of this work focuses on scene understanding for autonomous driving, it also touches upon indoor and general environments.

1.2 Structure of this Dissertation

This section briefly describes the structure of this dissertation. Chapter 2 introduces the fundamentals of autonomous driving, including the levels of autonomy (Section 2.1) and the history of it (Section 2.3). Chapter 3 provides an overview of scene understanding, with a focus on autonomous driving and its input signals (Section 3.3), tasks (Section 3.4), and challenges (Section 3.5).

Chapter 4 is about reliability in the context of scene understanding for autonomous driving, looking into its facets (Section 4.2), the impact of data (Section 4.4), common causes of reliability issues, such as challenging conditions (Section 4.5) and out-of-distribution data (Section 4.6), and measures to improve reliability, including uncertainty estimation (Section 4.7) and adversarial methods (Section 4.9).

Then, Chapter 5 delves into the contributions of this dissertation, addressing various aspects of reliability in the context of scene understanding for autonomous driving, such as the consistency of the outputs of panoptic segmentation (Section 5.1.1), the generalization to out-of-distribution data with uncertainty estimation (Section 5.2.1), adversarial augmentation (Section 5.2.2), or the segmentation of completely unseen and unknown objects (Section 5.2.3). Moreover, Section 5.3 focuses on reliability in the context of monocular depth estimation, proposing techniques to improve on dynamic objects (Section 5.3.1) and adverse illumination and weather conditions (Section 5.3.2).

Chapter 6 provides a summary of the findings, and Chapter 7 describes possible future works to improve reliability from the perspective of the methods (Section 7.1) and data (Section 7.2). Finally, the Appendix includes the abstract of a publication not discussed in this thesis (Appendix A), the supplementary materials of the works described in Chapter 5 (Appendix B), and their licenses from the publisher (Appendix C).

Autonomous Driving

Contents

2.1	The 6 Levels of Autonomy	9
2.2	The Benefits	10
2.3	Brief History	11
2.4	Societal Response	14
2.5	Impact on Regulations	15
2.6	Autonomous Driving Stack	16

Autonomous driving, or self-driving, is about enabling a road vehicle to navigate and operate without manual intervention. This chapter provides a brief overview of autonomous driving, from its benefits to its history and regulatory landscape.

2.1 The 6 Levels of Autonomy

Depending on the degree of autonomy of the system and that of involvement necessary for the human driver, the Society of Automotive Engineers (SAE) has defined a standard that distinguishes six different levels of driving automation [295]. They range from Level 0 (no automation, e.g., a classic car) to Level 5 (full unconditional automation, i.e., no driver nor steering wheel required) and are represented in Figure 2.1.

Up to Level 2, the vehicle offers various degrees of driving assistance features, which support the driver, who retains the responsibility. This means that the driver is offloaded from taking complete care of specific driving sub-tasks but still needs to carefully monitor the road and be ready to intervene at any time. Specifically, while Level 1 provides either steering or braking/acceleration support, Level 2 provides both at the same time [295]. Steering assistance keeps the vehicle centered in its lane. Braking and acceleration support include the adaptive cruise control, which maintains the speed and the distance from the car in front. Nowadays, many manufacturers offer Level 2 functionalities on various vehicles (e.g., BMW, Mercedes-Benz, Tesla, Toyota, and Volvo). These features significantly offload the driver's fatigue during long journeys on a freeway or stop-and-go traffic.

Starting from Level 3, the driver does not retain the responsibility of the driving tasks whenever the autonomy functionalities are engaged [295]. While Level 5 means full autonomy in all conditions, the automation of Levels 3 and 4 is constrained to specific situations or locations. For Level 3, the autonomous features can request the driver to intervene. An example of Level 3 functionalities is the automation in case of traffic jams and speeds up to 60 km/h, which recently started to be rolled out to customers and is available under limited conditions (Section 2.3). Instead, an autonomous taxi has

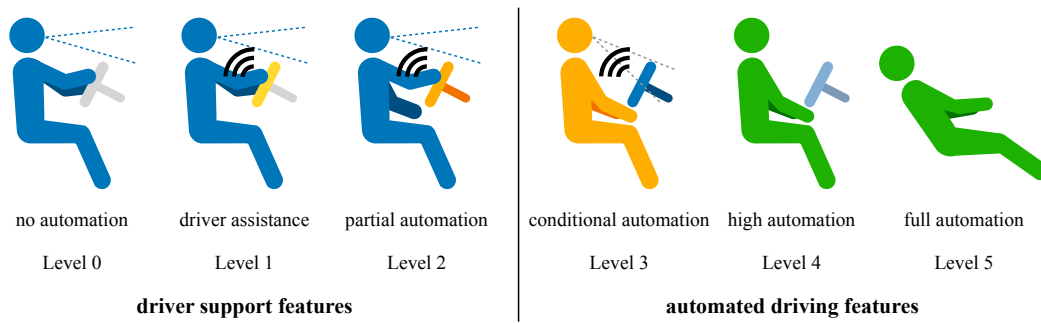


Figure 2.1. Representation of the 6 levels of autonomy [295]. At Level 0, no automation is in place. Level 1 includes lateral or longitudinal support. Level 2 features both kinds of support, i.e., adaptive cruise control and lane following. At Level 3, the responsibility shifts from the driver to the vehicle. With Level 4, no driver is required under certain conditions, e.g., geofenced. Level 5 indicates full and unconstrained automation.

Level 4 capabilities and does not need any controls to operate the vehicle. Level 4 autonomous taxis are already available in constrained areas for selected public users [315].

2.2 The Benefits

Autonomous driving promises numerous advantages that strongly motivate its exploration. As introduced in Section 1.1, a significant motivation to pursue autonomous driving is the increase in safety and the reduction of road fatalities [68] that are still one of the primary causes of death in the world [205]. This improvement would be possible by eliminating human factors such as distraction, fatigue, and even intoxicated driving [134].

Self-driving cars can also reduce traffic congestion by communicating with each other and the infrastructure, optimizing traffic flow [317]. This would improve efficiency and help save time spent in traffic. However, this would require a significant share of autonomous vehicles on the road and connected traffic lights and street sensors. Furthermore, it was found that cars searching for parking in major cities account for an average of 34% of traffic [106]. Self-driving vehicles could help mitigate that by autonomously reaching further parking structures or via autonomous ride-hailing and shared mobility concepts. Moreover, autonomous vehicles could substantially lower greenhouse gas emissions by reducing the need for stop-and-go driving and allowing for optimized path planning.

Thanks to full autonomy, people who cannot drive due to disabilities or age restrictions could get driven by the vehicle, improving accessibility [6]. This would provide greater independence, enhancing the quality of life of many people. Moreover, the technology would improve comfort and productivity by allowing all passengers to engage in various activities, such as working or video streaming, as imagined by DALL·E 3 [23] in Figure 1.1. Autonomous driving would also revolutionize the trucking industry, enabling longer drives by not having to undergo driving time restrictions and significantly improving efficiency via platooning [305]. Logistics costs could be reduced depending on the equipment costs for autonomy (e.g., sensors), as the driver would not be needed. Similarly, other sectors could change significantly towards improved services, including taxi, delivery, and public transportation.

Car ownership in larger cities could change dramatically in favor of car sharing. Since fewer vehicles would be needed, this would reduce the need for parking and revolutionize urban planning. Neverthe-

less, car sharing is already available in many cities, yet car ownership is still widely established. A study conducted in Germany in 2018 [219] suggests that private cars will remain the preferred mode of transportation. While the preferences may change, the need for mobility will persist, so the popularity of one mode or the other may depend on the costs and availability of autonomous car sharing compared to other solutions such as car ownership, public transportation, or biking infrastructures [2]. Such preferences will likely differ between cities and rural areas [53].

Overall, autonomous driving technology has the potential to have a major impact on our society, thereby motivating significant efforts in its research and development.

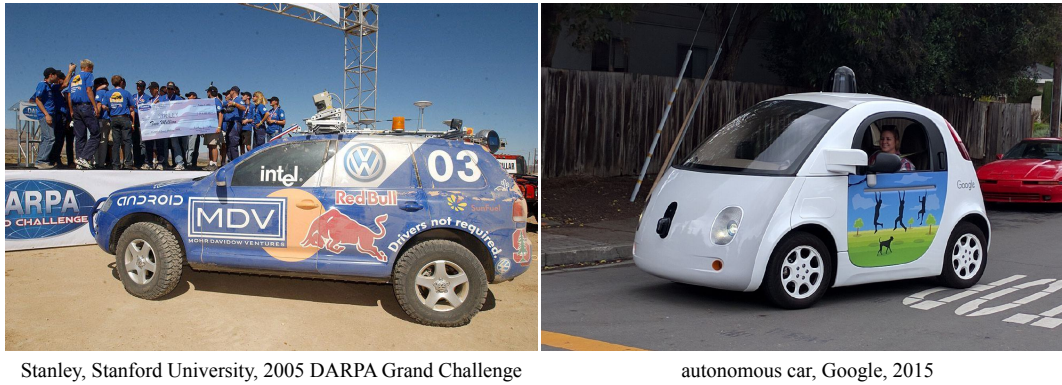
2.3 Brief History

The concept of autonomous vehicles dates back to the early 20th century. The idea of an autonomous passenger car was already envisioned in 1918 [226]. One of the earliest prototypes of an autonomous vehicle was a radio-controlled car demonstrated in the late 1920s. While these vehicles were not autonomous in the modern sense, they marked the beginning of this field. Modern cruise control, i.e., the system that automatically controls the speed of a vehicle, was invented in 1948 and patented in 1950 [296]. This technology, which found its way into a production car in 1958 with Chrysler, is considered to have laid the foundations of autonomous driving. A significant leap was the Stanford Cart, a remote-controlled moon rover repurposed to navigate terrestrial terrains. Developed at Stanford University in the 1960s, by 1971, the cart used computer vision to slowly navigate through obstacle courses [207].

Adaptive cruise control can be traced back to the late 1980s and early 1990s [128]. On top of maintaining the speed, adaptive cruise control keeps the distance from the vehicle driving in front. The first commercial version of the adaptive cruise control was introduced by Mitsubishi in Japan in 1995 with the model Diamante [333], which measured the distance with a laser. However, this technology was not connected to the breaks and was limited to 67 mph (i.e., 107.8 km/h). In 1999, with the S-Class, Mercedes-Benz introduced the first radar-based adaptive cruise control, operating at higher speeds and connected to the brakes. The same year, Subaru introduced the first adaptive cruise control based on cameras.

In 1989, researchers at Carnegie Mellon University developed ALVINN [238], an autonomous vehicle equipped with a camera and a laser range finder capable of driving along real roads under certain conditions. ALVINN was built upon the NAVLAB project [297], which produced a series of minivans equipped with hardware to execute autonomous functions. This project led to the first coast-to-coast autonomous trip across the USA in 1995, from Pittsburgh, Pennsylvania, to San Diego, California [131, 237]. Remarkably, the vehicle drove autonomously for 98.2% of the 2849 miles (i.e., 4585 km).

In 2004, the first DARPA Grand Challenge marked a turning point in autonomous driving. While no vehicle successfully completed the course, the event sparked significant interest and investment in the field [33]. The subsequent 2005 challenge saw a dramatic improvement, with Stanley from Stanford University becoming the first vehicle to complete a challenging off-road course of 132 miles (i.e., 212.4 km) in the Mojave Desert [298]. Stanley is shown on the left of Figure 2.2. The 2007 Urban Challenge required vehicles to navigate a simulated urban environment while obeying traffic laws.



Stanley, Stanford University, 2005 DARPA Grand Challenge

autonomous car, Google, 2015

Figure 2.2. Examples of successful earlier efforts towards autonomous driving. The left picture shows the autonomous desert racing vehicle Stanley [298] by Stanford University. The Stanford team won the 2005 DARPA Grand Challenge as Stanley autonomously drove through the course in the Mojave Desert. The vehicle is a Volkswagen Tuareg with various sensors, including LiDARs on the roof. The right image shows the first autonomous vehicle built by Google around 2015. The car is a custom-built two-seater without driving controls fitted with perception sensors such as the roof-mounted LiDAR. Earlier, starting in 2009, Google was testing on retrofitted Toyota Prius and Lexus RX. Then, in 2016, Google’s Self-Driving Car Project became Waymo. The right image is taken from Wikipedia under the license <https://creativecommons.org/licenses/by-sa/4.0/>.

Carnegie Mellon’s Boss won, showing that autonomous driving could be feasible in more complex scenarios [309].

Google started a driverless car project in 2009 [2]. Around 2015, as shown in Figure 2.2, Google developed a small autonomous car without driving controls. In the same year, Google became the first to achieve autonomous drives on public roads with an emergency driver [274]. By the end of the 2010s, countless players worldwide followed various strategies to push self-driving technology. This includes efforts from carmakers such as BMW [284], Mercedes-Benz [75], Toyota [101], and Tesla, automotive suppliers such as Bosch, Valeo [264], and Continental, tech firms such as Google/Waymo [290], Baidu, Nvidia [12], and Intel, ride-hailing services such as Uber [306], DiDi, and Lyft [120], as well as rising startups and autonomous driving companies such as Cruise, Mobileye, Pony.ai, Zoox [95], and Wayve [144]. Many academic institutions also contributed to the progress. These include the Carnegie Mellon University [309], the Stanford University [298], the ETH Zurich [258], the Technical University of Munich [4], and the Karlsruhe Institute of Technology [90].

In the early 2020s, a period of disillusionment began, fueled by the COVID-19 pandemic, alongside missed estimates, rising costs, and unclear business models, leading various companies to revisit their goals in the context of autonomous driving. Many players abandoned this race for autonomy and completely shut down their self-driving efforts, including Uber. Others shifted their focus to driving assistance functionalities (Levels 2 and 3), reducing or even canceling their investments in autonomous vehicles (Levels 4 and 5).

As of late 2023, as shown in Figure 2.3, various passenger cars and trucks on sale are equipped with Level 2 functionalities (Section 2.1). Cameras are already widely available onboard, enabling various features, such as emergency braking, lane keeping, and the monitoring of blind spots. Similarly, radar sensors are commonly used for adaptive cruise control. Furthermore, a few specific recent models already feature integrated LiDAR sensors for improved safety and assistance features (e.g., Volvo EX90 and Xpeng P5). In particular models and markets, e.g., on BMWs in North America or China,

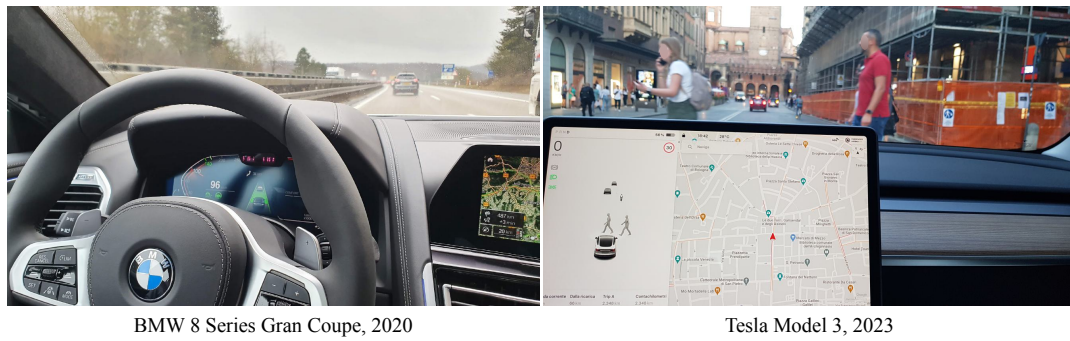


Figure 2.3. Examples of assisted driving features already available in production cars. Both vehicles shown offer Level 2 functionalities, including adaptive cruise control and lane following. BMW equips its vehicles with radars, cameras, and ultrasonic sensors (soon also LiDARs), while Tesla advocates a camera-only approach (although it may soon add back radars). The BMW 8 Series was captured in February 2020 on a rainy day in the German Autobahn. The controls for the driving assistance features are shown on the left spoke of the steering wheel, including the dial for adjusting the speed, the buttons for the distance to keep from the vehicle in front, and the button to activate lane following. BMW features lights on the steering wheel (above the buttons on the spokes) to signal the engagement of the assistance features and call for the driver's attention with yellow or red flashes alongside audible warnings. The picture of the Tesla Model 3 was taken in October 2023 at a pedestrian crossing in the center of Bologna, Italy. Both cars provide live feedback of what the vehicle is perceiving. BMW indicates this in a simplified view in the instrument cluster and head-up displays, while Tesla provides more information in the central display, including pedestrians, traffic signs, and traffic lights. The images have been anonymized, hiding the faces of the pedestrians.

the driver's attention is monitored via cameras or sensors in connection to the engagement of the driving assistance features. As long as the driver looks at the road ahead, the vehicle does not require any inputs and operates at Level 2. When the driver gets distracted, the car demands inputs from the driver (e.g., via light flashes and acoustic warnings), such as the touch of the steering wheel. If the driver does not interact, the vehicle reaches a safe stop and calls the emergency services, assuming medical intervention is needed. In other cases, the driver attention monitoring is performed only to warn about fatigue, e.g., BMWs in Europe, and the driver interaction is required a priori every few seconds, when it is typically sufficient to touch the steering wheel. The different availability of these features depends on the regulations governing each market.

A few years ago, in 2021, Honda became the first to offer Level 3 autonomy on public Japanese roads, with a limited edition of 100 cars, marking a remarkable achievement. These features were made available under certain conditions, such as with congested traffic on a freeway. Recently, in early 2023, Mercedes-Benz became the first to offer Level 3 functionalities on a non-limited series car, the S-Class. Such Level 3 features from Mercedes are initially available only in Nevada, USA, on suitable freeway sections with dense traffic and speeds up to 40 mph (i.e., 64.4 km/h). Since then, as of late 2023, other manufacturers have been obtaining or have already undergone the certifications for Level 3. For example, BMW is set to enable its 5 and 7 Series to drive hands-free on German and American freeways up to 130 km/h (i.e., 80.8 mph) and is the first to release Level 3 functionalities even at night.

Before that, in 2022, Mercedes-Benz was also the first carmaker to offer Level 4 functionalities, together with Bosch, albeit only for parking and restricted to a single parking lot in the Stuttgart Airport, equipped with various sensors. Starting in early 2022, a handful of autonomous driving companies, including Waymo, Cruise, Motional, and Zoox, have been granted permission to offer the public an autonomous ride-hailing service in a limited number of cities, including San Francisco



autonomous car, Waymo, 2023

autonomous testing vehicle, Zoox, 2023

Figure 2.4. Vehicles by Waymo and Zoox equipped with autonomous driving sensors and features. The pictures were captured in the streets of San Francisco, California, in June 2023. The Waymo car on the left has no driver and can offer autonomous rides to selected users, operating as a robot taxi. Vehicles like the retrofitted white Jaguar I-Pace shown are equipped with many sensors in various housings, especially around the roof area. Zoox is developing a mini-bus without driving controls on board and is testing on public roads with retrofitted Toyota Highlander SUVs equipped with multiple sensor rigs featuring numerous sensors, especially around the roof and front of the vehicle. While the sensor configuration between the two is different, both companies opted for multiple LiDARs, cameras, and radars. The images have been anonymized, hiding license plates and vehicle identifiers.

and Las Vegas. These were the first publicly available services with Level 4 functionalities without a safety driver. Examples of such vehicles captured in San Francisco are shown in Figure 2.4. In January 2023, Waymo reached over 1M miles (i.e., 1.6M km) without any driver and reported no injuries [315], which is an incredible result.

These astonishing achievements were made possible thanks to a significant joint effort from multiple players and the sharing of research outcomes. While the technology has progressed tremendously, autonomous driving is not a solved problem, as several challenges still need to be overcome to achieve full autonomy and expand or even lift the geofences currently constraining autonomous operations to limited areas. For example, following a severe accident dated October 2023, the authorities revoked Cruise's license to offer autonomous rides amid safety concerns about the readiness of their technology, highlighting the still current need for substantial developments. In particular, this demand for improved reliability, especially towards unseen scenarios, motivated the research of this dissertation.

2.4 Societal Response

The introduction of autonomous driving has led to a wide range of reactions from society, with a mix of enthusiasm and concerns [2, 74]. For example, with respect to safety, some people are concerned about the transition period when human drivers and autonomous cars share the same roads. Other concerns revolve around how autonomous machines make moral decisions and who should judge which one to implement [10, 98]. Moreover, questions surrounding liability in accidents involving autonomous vehicles are also problematic [114, 195, 196].

While some believe this technology will create new job opportunities, especially in the tech industry, others are worried that it will lead to unemployment in sectors such as trucking, taxi driving, and delivery services [77]. These economic implications form a significant friction point in the societal debate around autonomous driving. Furthermore, there are concerns about data privacy [49]. With

collecting and processing large amounts of data, concerns regarding user privacy and data security are substantial. The potential misuse of data is an issue that feeds into the broader societal response about the role of technology in daily life.

Many of these concerns are similar to those on artificial intelligence and technology adoption in general. In this regard, the popularity of products based on artificial intelligence, such as the widespread interaction with large language models through ChatGPT [31, 215] beyond tech enthusiasts, may have improved the acceptance of artificial intelligence by the broader public, thus that of autonomous vehicles.

2.5 Impact on Regulations

The advent of autonomous driving has brought changes not only in the way society conceptualizes mobility but also in how it regulates it. The transition from human-operated to machine-operated vehicles presents significant novel challenges that require rethinking existing legal frameworks and creating new ones. This section provides an overview of the complex interplay between autonomous driving technologies and regulatory measures.

Traditional traffic laws are predicated on the notion of human decision-making and responsibility. As autonomous vehicles have started to become a reality, adjustments to these laws are needed to accommodate non-human agents on the road [281]. To comprehend the ongoing changes in global regulations, one must first recognize the established norms. A critical historical point is the Vienna Convention on Road Traffic of 1968, signed by 77 countries, including all members of the European Union. The convention states in its Article 8 that "Every moving vehicle or combination of vehicles shall have a driver" and "Every driver shall at all times be able to control his vehicle" [229, 314]. Since the convention defines the driver as a person who drives a vehicle or guides cattle, it does not account for the development of vehicles that could operate without human intervention.

Amendments to this convention have been required to make room for autonomous vehicles. In 2014, a critical amendment was approved, enabling autonomous driving technologies as long as they can be overridden or switched off by the driver, thereby opening up a legal framework for autonomous driving on an international level [229]. While this amendment came into force in 2016, it still demanded that every vehicle must have a driver. Therefore, highly and fully automated cars were still not allowed by the 2014 amendment [229]. A new amendment from 2022 expanded the convention to include fully automated vehicles, allowing their use in the EU.

In the USA, autonomous vehicle regulation falls under federal and state jurisdictions. While the National Highway Traffic Safety Administration (NHTSA) has issued multiple iterations of guidelines for autonomous vehicles, these are voluntary recommendations rather than binding rules [64]. States like California and Nevada have taken proactive steps to permit the testing and deployment of autonomous vehicles within their jurisdictions, leading to a patchwork of state-specific regulations [6].

Countries like China, Singapore, and Japan are also making considerable steps forward both in technology and regulatory landscapes. Regardless of the countries, these efforts are often being approached through pilot projects that allow limited-scale testing of autonomous vehicles under specified conditions [68].

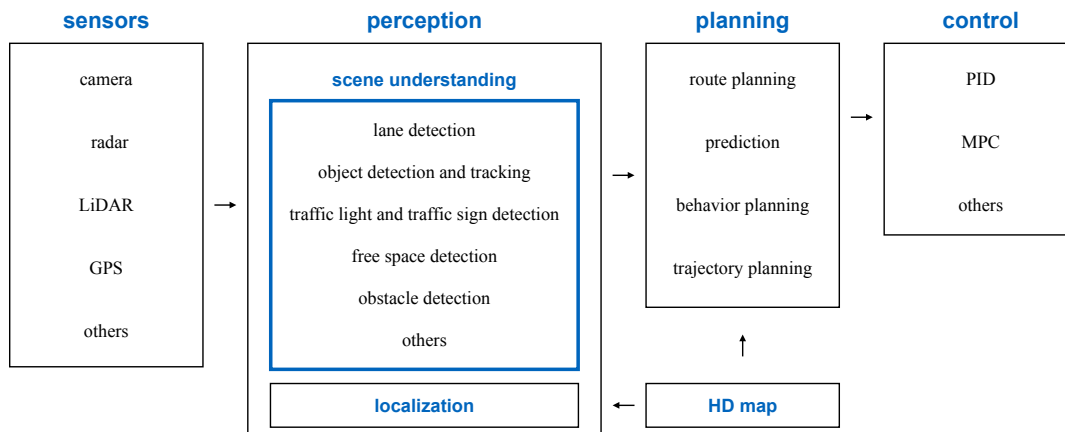


Figure 2.5. Representation of the standard autonomous driving software stack. Sensory inputs and an HD map are processed by the perception module, which performs scene understanding and localization. This feeds the planning component responsible for estimating the ego trajectory given that of others. The control module finally executes the computed trajectory. The focus of this dissertation is on scene understanding, highlighted in the diagram by the blue box.

While technology advances, regulations also need to consider ethical and legal dimensions. Questions regarding liability, data privacy, and cybersecurity remain significant concerns for policymakers [93]. The operation of autonomous vehicles involves collecting and processing vast amounts of data, raising substantial concerns about data governance and cybersecurity. Regulations are needed to determine who owns this data, how it will be used, and how to secure it against potential cyber-attacks [93]. Ethical guidelines for decision-making algorithms in autonomous vehicles are being developed but are yet to become part of mainstream regulation [28]. Traditionally, driver liability has been a cornerstone of traffic law. However, the shift to autonomous vehicles is moving the focus from driver liability to product liability, making insurance companies reevaluate their frameworks [195].

With the international nature of automotive markets, regulations need to address the interoperability of vehicles between different jurisdictions. Ensuring that vehicles produced and used in one country can safely and effectively operate in another country is essential for the mass adoption of autonomous vehicles [190]. This legal aspect is related to the technical ability of the system to generalize robustly across different scenarios, which is the core topic of this dissertation.

2.6 Autonomous Driving Stack

Beyond ethical and regulatory challenges, one of the most crucial components to reach full autonomy is the software stack that governs everything from perception and decision-making to control systems. This thesis focuses on improving reliability in scene understanding, which is an integral part of the perception layer. This section offers a structured overview of the typical layers in an autonomous driving stack.

While a recent trend explores end-to-end autonomous driving solutions, where a unified model takes sensory data in input and outputs a driving trajectory [26, 144], the standard approach to autonomous driving divides the problem across multiple components, forming the so-called autonomous driving stack, illustrated in Figure 2.5.

The foundation of the stack is built upon sensors such as cameras, LiDAR (Light Detection and Ranging), radar (radio detection and ranging), and others that collect real-time data from the vehicle's surroundings [50, 90]. Section 3.3 provides more details on the sensors. The complexity and volume of the data make the subsequent layers crucial for real-time, reliable decision-making.

At the perception layer, raw sensor data is processed to identify and understand key elements in the surrounding environment. The perception component performs scene understanding to extract all the information from the driving scene that is useful for driving. Identifying objects, their positioning, and semantics are core tasks at this layer [71, 369], e.g., through object detection and semantic segmentation (Section 3.4). This thesis contributes to this layer, enhancing the reliability of scene understanding methods (Chapter 3).

Within the perception layer, the stack includes localization mechanisms that situate the vehicle precisely within a map and mapping functions that continuously update internal maps of the environment [111, 112, 272]. Accurate localization and mapping are essential for reliable navigation and path planning.

Together with the localization data, the perception information is abstracted (e.g., object locations and semantic classes) and forwarded to the planning module, which predicts the trajectory of the other traffic participants [188, 266]. Then, considering the positioning, the surroundings, and the local traffic rules, the planning layer generates a safe, legal, and efficient trajectory for the ego vehicle to follow [58]. It decides on aspects like lane-changing, obstacle avoidance, and route optimization.

The final layer is the control systems that translate planning decisions into real-world actions. Using algorithms like PID controllers and Model Predictive Control (MPC), it regulates vehicle acceleration, braking, and steering via actuators [69, 334]. As the ego vehicle constantly perceives the updated surrounding environment, it computes the latest optimal trajectory and follows it autonomously.

Different challenges are associated with each layer. For the perception component, reliability is a significant one. The primary contribution of this thesis lies in strengthening the perception layer against a myriad of real-world variabilities, thereby augmenting the reliability and safety of the entire autonomous driving stack.

Scene Understanding

Contents

3.1	Overview	19
3.2	Applications	20
3.2.1	Robotics	20
3.2.2	Healthcare	21
3.2.3	Augmented Reality	21
3.3	Input Signals for Autonomous Driving	22
3.4	Perception Tasks for Autonomous Driving	23
3.5	Challenges in Scene Understanding for Autonomous Driving	25

3.1 Overview

Humans can interpret complex scenarios with remarkable precision. This ability is crucial for understanding the environment and interacting with it. Visual perception is fundamental to activities such as picking up an object or moving. For a computer-based system, this is possible thanks to interpreting the sensory inputs, such as vision and touch, and translating the acquired information into motion and action (e.g., grasping the object). Similarly to humans, autonomous agents perceive the environment through sensory inputs and interpret them via scene and object understanding [115].

Scene understanding is the ability of a system to comprehend and interpret visual information gathered from the surrounding environment. In a standard perception-action model, the abstract information extracted by the scene understanding component is then used to make decisions in the form of actions. For an autonomous system, these actions include crucial aspects such as navigation and interaction with the environment. To safely navigate through the environment, scene understanding aims to analyze the content of the inputs and extract relevant information, such as the presence of obstacles in the surroundings.

The components of scene understanding receive visual information in input, which can be captured by various sensors. In indoor settings, standard sensors include RGB or RGB-D cameras. Instead, the most typical sensors for autonomous driving are RGB cameras, LiDARs, or radars (Section 3.3). While RGB cameras output 2-dimensional data, RGB-D cameras, as well as LiDAR and radars, provide data that is 3-dimensional and can be represented as point clouds. Reliable 3D information about the surroundings of an autonomous agent is of utmost importance for its safe operations. Furthermore, additional sensors can be available, such as GPS and inertial measurement units (IMU), as well as extra odometry information, e.g., speed and steering angle. All this data is typically available over time in the form of sequences.

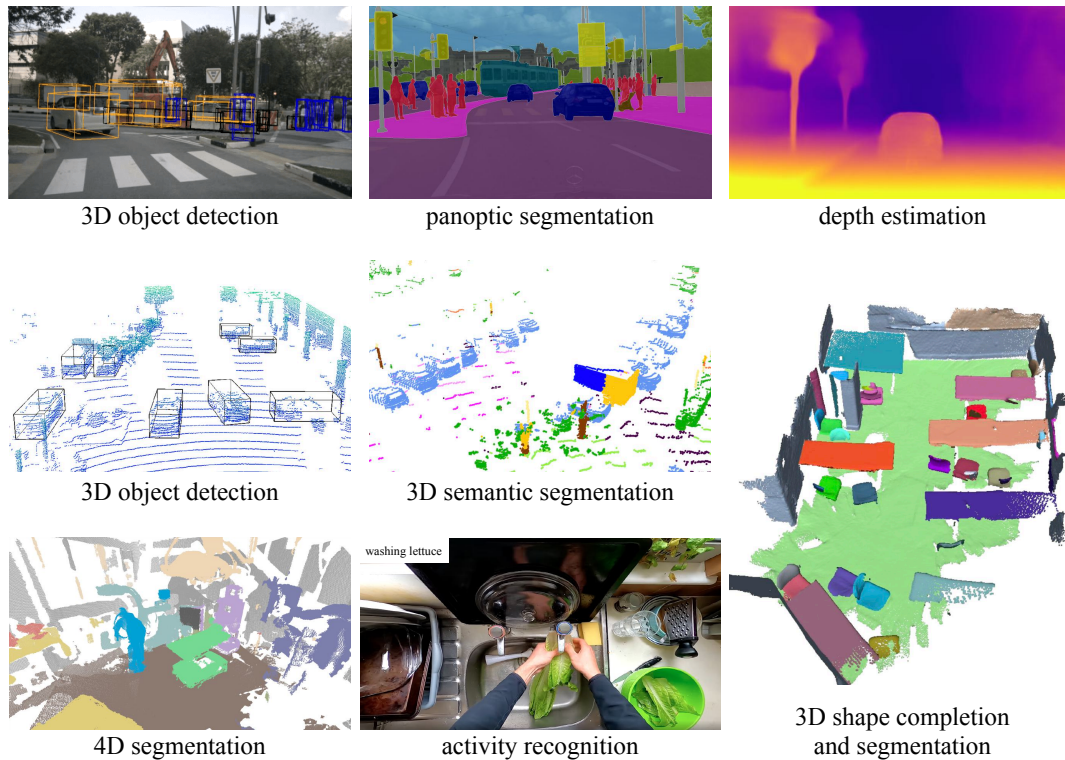


Figure 3.1. Examples of scene understanding tasks and applications, from outdoor to indoor. Autonomous driving is a prominent application for scene understanding, both in 2D and 3D. Robotics, augmented reality, and healthcare are also popular applications in 2D and 3D as well. Image sources: 3D object detection (top) from nuScenes [35], panoptic segmentation from Cityscapes [50], depth estimation from our md4all [86], 3D object detection (middle) from our 3D-VField [162], 3D semantic segmentation from our [163], 4D segmentation across space and time from SegmentOR [15], activity recognition of videos from ARGO1M [233], 3D shape completion and panoptic segmentation through incremental 3D scene graphs from SceneGraphFusion [339].

3.2 Applications

As shown in Figure 3.1, scene understanding is crucial to various applications, such as robotics, healthcare, augmented reality, and autonomous driving. While autonomous driving is described in greater detail in Chapter 2, this section provides an overview of other applications. Additional use cases not discussed here include monitoring traffic [136], detecting anomalies in surveillance footage, agriculture and environmental monitoring [192], and understanding customer behavior in the retail industry.

3.2.1 Robotics

Robotic systems are progressively becoming integral to various industries, from manufacturing to healthcare. Scene understanding plays a fundamental role in enhancing the capabilities of autonomous systems, enabling them to interact with and navigate through complex environments [218]. A brief overview of the applications of scene understanding is presented in the following paragraphs. This touches upon object understanding, which, unlike scene understanding, is aimed at comprehending individual objects instead of an entire visual scene or environment. At the object level, the goal is to understand the object’s properties and, in particular, its position and orientation when aiming to

interact with it, e.g., via grasping. Depending on the rigidity [289, 358] or deformable nature of the object [17, 66, 259], different strategies exist to relate objects' position, rotation, and shape with one another.

Robotic arms equipped with vision sensors use scene understanding to identify objects, estimate their 6D poses [62, 142, 193, 251, 323], and understand how to grasp them [115, 358]. This is crucial for tasks like picking and placing objects in manufacturing processes, warehouses, and assembly lines [22]. Scene understanding is also important to avoid collisions and adjusting grip forces based on the properties of an object, such as its material [224].

Whether in industrial settings such as robot arms [158], in household environments such as robot vacuum cleaners [220], or outdoors as self-driving cars [90], autonomous systems need to plan their trajectory and execute it by navigating the environment and avoiding obstacles [321]. Towards this end, scene understanding allows an autonomous agent to identify obstacles in real-time [40], reconstruct [338], and create maps of its surroundings [113, 171], used plan collision-free trajectories. This is particularly relevant in those environments where the layout changes over time [319], such that the agent needs to adapt to the new obstacles in the scene.

Autonomous vehicles are essentially robots. They constitute a crucial application of scene understanding and they are described in Chapter 2. Drones are also autonomous vehicles when they fly autonomously. Thanks to a wide range of sensors and computer vision systems to process their data stream, drones can interpret their surroundings, identify obstacles [40, 199, 355], find ideal landing points [11, 228], and even reconstruct the environment [221]. This enables them to navigate autonomously through challenging scenarios [59], e.g., forests and racing courses [141, 169], monitor terrains, e.g., agricultural fields, conduct search and rescue operations, and even deliver packages [104].

3.2.2 Healthcare

In the context of medical applications, scene understanding can be used, for example, to analyze surgical rooms [15, 329]. Activity recognition in the operating room identifies, analyzes, and categorizes the actions performed by practitioners during surgeries [14, 54]. Thanks to this analysis, the surgical workflow can be summarized by a sequence of events [217] or a scene graph [116, 216]. This builds the foundation for analyzing and optimizing the workflow, leading to context-aware intelligent systems and improving patient care, thereby enhancing the quality of surgical procedures. Scene understanding in the operating room can enable real-time feedback to the surgeons, anomaly detection, and post-operative analysis of the surgical outcomes. Furthermore, the surgical workflow analysis can provide textual descriptions of the surgical images and anonymization [16], preparing the data for education and documentation.

3.2.3 Augmented Reality

Augmented reality (AR) is about overlaying digital information, such as text, images, videos, or 3D objects, onto the view of the real world from the user's perspective [273, 288]. Scene understanding is important for integrating these digital elements into the user's view. In particular, such digital

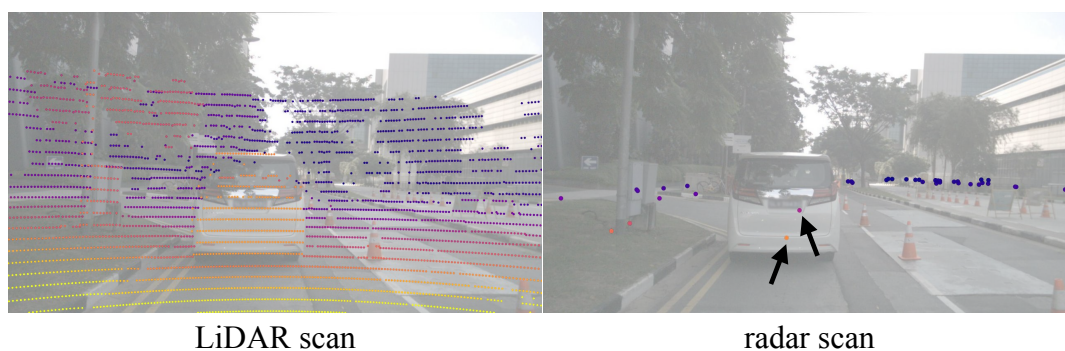


Figure 3.2. Example of LiDAR and radar scans projected to the corresponding image. The data was captured on a clear day in Singapore and is part of the nuScenes dataset [35]. Radars offer sparse measurements with significant levels of noise due to multi-path effects. The points are colored depending on their distance. The arrows indicate two radar points projecting on the same car but measured to be over 5 meters apart [83] [©2021 IEEE]. This is due to the radar projecting along a horizontal plane, so higher points correspond to further measurements, and the further point is due to the radar waves reaching beyond the minivan passing underneath it. LiDARs are less noisy and denser but suffer from other issues, such as artifacts with adverse weather (Section 5.3.2). The LiDAR scan shown is not particularly dense as it was captured with a 32-beam sensor. Other sensors reach 64 [90, 290] or 128 beams, offering higher resolutions.

elements are the augmentations on top of the real video stream from a camera. As the technology improves and the amount of use cases increases [308], AR has found applications not only in the gaming [222] and social media [129] domains but also in industrial [332], educational [29], and medical applications [210]. AR-ready devices include head-mounted displays, smartphones, and even head-up displays from the automotive sector. Scene understanding helps AR by providing information on where to anchor the virtual objects in the real world. Augmented objects can be placed on surfaces, e.g., tables, and interact realistically with the physical environment. Furthermore, with virtual try-on [350], users can see how clothes and accessories would look on them in real-time using AR.

Egocentric vision involves interpreting visual data from a first-person perspective via wearable cameras [234]. Especially if head-mounted, such a point of view is inherently dynamic and aligns closely with the user's interactions with the scene [55], enabling the recognition of the user's activities from videos [233, 235]. In particular, egocentric vision is important for augmenting the views of head-mounted displays or AR glasses, where 3D localization becomes essential [231]. Understanding the user's perspective enables AR features such as visual navigation, instruction-based aids, or visual clues to support the activity [232]. From such a perspective, hand-object reconstruction can become relevant for AR, too [165].

3.3 Input Signals for Autonomous Driving

A wide range of complementary sensors are commonly used for autonomous driving: RGB cameras, radars, and LiDARs. Cameras output a video stream of RGB images. They are typically placed at the top of the windshield, on the roof, and around the vehicle. Specific setups offer stereo vision with a left-right camera system [90], and others have cameras positioned all around to provide 360° coverage [35]. Cameras are the most inexpensive vision sensors and are sensitive to changing illumination conditions. For example, cheaper cameras tend to be particularly noisy at night [35] due to the small sensor size and low-quality optics. To mitigate this, they are often paired with other sensors, such as radars and

LiDARs, which provide native 3D measurements. Thus, these sensors add a level of redundancy that is beneficial for safety. Radar and LiDAR are active sensors that emit a signal and detect the same signal coming back to the sensor after being reflected by a surface. By measuring the elapsed time between emission and detection, they measure the distance to the obstacles, provided in the form of a point cloud.

Spinning LiDARs are typically roof-mounted and the most expensive sensors for autonomous driving perception, with a five-digit cost in USD. This is due to the precise movement they need to achieve to provide centimeter-level accuracy at 100-meter distances. This and high-frequency waves ensure higher accuracy than radars, which employ longer radio waves. Common automotive radars [35] tend to be more cost-efficient, albeit providing highly sparse detections and suffering from high noise levels [83]. Solid-state LiDARs are an evolution of traditional LiDAR systems without moving parts, relying instead on technologies like optical phased arrays or electronically steered beams [223, 241]. Their reduced mechanical complexity increases durability and reliability and reduces costs (mid-three-digit). The smaller form factors and forward-facing capturing enable more straightforward integration into vehicle designs.

Each sensor comes with its challenges. For example, cameras do not provide 3D information, radars are sparse and noisy as shown in Figure 3.2, and LiDARs are sparse at further distances [84] and unreliable with adverse weather conditions [86]. While solid-state LiDARs have a significantly lower price tag than their mechanical counterparts, their resolution and range tend to be inferior. However, these sensors offer complementary benefits: cameras are dense and relatively inexpensive, radars are reliable in adverse weather, and LiDARs offer precise 3D measurements. This can be exploited via sensor fusion strategies [47, 83]. While radars are already commonly available for assisted features such as adaptive cruise control, LiDARs just found their way into passenger cars (e.g., Volvo EX90). In the future, high-resolution radars and solid-state LiDARs might combine the advantages of each in a single, robust 3D sensor offered at a mid-high three-digit cost.

This dissertation proposes methods working with cameras [82, 85, 86], LiDARs [84, 162], and radars with cameras [83]. Additional sensors can be present in a self-driving vehicle, but they are less common and out of the scope of this work. This is the case of ultrasonic sensors for close distances, spinning high-resolution radars [13], thermal cameras [209], event-based cameras [194], and gated cameras [248, 320].

3.4 Perception Tasks for Autonomous Driving

Perception is the vision system of an autonomous vehicle (Section 2.6). It comprises one or more computer vision algorithms performing scene understanding. The goal of perception is extracting from the sensor data everything relevant for driving to fully understand the surroundings. As shown in Figure 3.3, this includes identifying obstacles, the driving surface, and the positioning of the ego vehicle with respect to the other traffic participants. Therefore, a scene-understanding approach is typically developed to solve a specific task.

Object detection is the task of identifying the position in 2D or 3D of objects of interest [82, 162, 208]. In autonomous driving, such objects are other traffic participants (e.g., cars and pedestrians), traffic signs, cones, animals, and other obstacles. The category of the objects is also estimated since each

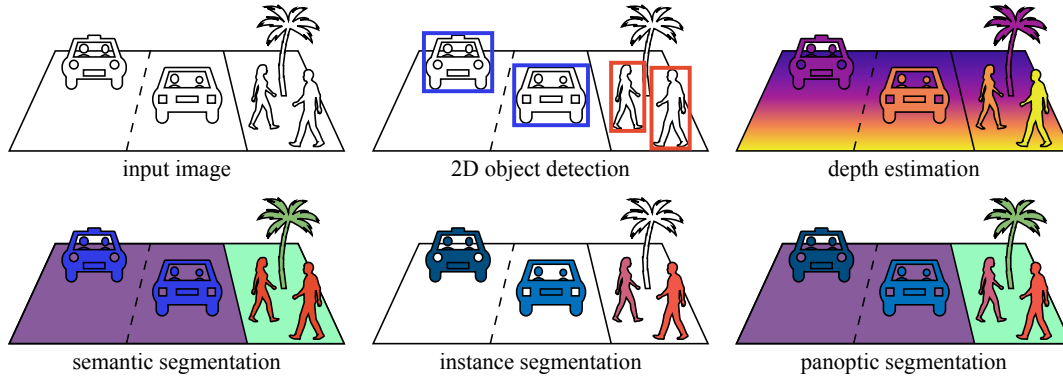


Figure 3.3. Representation of various 2D scene understanding tasks for autonomous driving. Object detection aims to find the presence and location of objects of interest (e.g., cars). Depth estimation is about predicting a distance measurement for each pixel in input. Semantic segmentation assigns a semantic class to each pixel. Instance segmentation identifies the individual pixels of an object. Panoptic segmentation combines semantic and instance segmentation. Object detection can also be performed in 3D from 2D or 3D data. Segmentation tasks can also be performed on 3D data, e.g., assigning semantic estimates to each point of a 3D point cloud instead of a pixel.

type behaves differently on the road. These objects can be static or dynamic, and they are tracked to estimate their future trajectories via object tracking [164, 187].

Semantic segmentation is the dense task of assigning to each unit in input (e.g., pixel) a semantic class of interest [236]. Therefore, as displayed in Figure 3.3, the input is divided into a set of regions (i.e., segments) according to their category. These classes include other traffic participants, roads, sidewalks, buildings, and trees. Depending on the sensor being used, this can be done in both 2D and 3D. Similarly, in the image domain, instance segmentation identifies all pixels belonging to an individual object (i.e., instance). The same can be done in 3D on the points of a LiDAR point cloud. The combination of semantic and instance segmentation is called panoptic segmentation [150]. In this dissertation, we introduce the first end-to-end panoptic segmentation method [84], delivering semantic classes and instance IDs directly from the network’s output (Section 5.1.1).

Furthermore, depth estimation predicts the distance from the camera of what is projected at every pixel [94, 154, 156]. This is important since LiDAR data is rarely available due to the high cost of the sensor. If LiDAR data is available, then the task of making it denser by exploiting image data is called depth completion [347].

The contribution of this thesis touches upon most of these tasks. Other perception tasks relevant for autonomous driving and not performed in this work include localization [111, 133, 261, 262, 263], mapping [112], simultaneous localization and mapping (SLAM) [30, 139, 172, 294], visual odometry [181], lane detection [1], pedestrian intent detection [176] for which human pose estimation is highly relevant [244], vehicle intent detection [126], traffic sign recognition [57], traffic light recognition [56], and free space detection [70]. Furthermore, tasks such as pedestrian trajectory estimation [249] or vehicle trajectory estimation [170] are more related to the planning module (Section 2.6). Still, they may be done in the perception component if carried out directly from the sensory data instead of abstractions of the objects’ position and prior trajectories.

To compare the effectiveness of methods performing a task, a standard set of metrics is computed on specific data, which serves as a benchmark. Depending on the task, different benchmarks exist, such

as the KITTI dataset [90] for 3D object detection or SemanticKITTI [18] for LiDAR semantic and panoptic segmentation.

3.5 Challenges in Scene Understanding for Autonomous Driving

Many open challenges prevent the widespread of autonomous vehicles. These obstacles are related to a variety of domains, including regulations, public adoption, costs, and, of course, technical difficulties. While not all technical issues are due to perception, this thesis focuses on scene understanding so this section presents a brief overview of the challenges in this domain.

For scene understanding in autonomous driving, various technical challenges undermine reliability. These can be seen from two main perspectives: data and methods. In terms of data, it is difficult and expensive to collect interesting and valuable data. Learning-based methods require training with large amounts of data, which should cover all possible scenarios. However, such comprehensive data is hardly available in autonomous driving and other domains such as medical imaging [225, 312, 313] and surgical settings [15].

When driving, we mostly experience similar basic scenarios and rarely encounter dangerous situations and extreme cases that require immediate emergency maneuvers or particular caution (e.g., a deer crossing the road or a child suddenly appearing between parked cars). Analogously, when collecting data, most scenes are similar and do not bring any added value to the ones collected previously.

Collecting the long tail of the data distribution (i.e., the rare scenarios) is extremely difficult [85]. For example, it is problematic to record scenes with a heavy hail storm, not only because hail storms occur rarely but also because they may damage the expensive sensors (e.g., LiDAR). Through simulations, it is possible to generate long tail samples artificially, but that introduces new issues due to the difficulty of rendering adverse conditions realistically, which leads to a domain gap, requiring domain generalization or adaptation strategies [121, 162, 313].

If long tail data is unavailable or scarce, then scene understanding methods must be robust and generalize well to completely unseen scenarios. Evaluating such ability is challenging, too, since data is needed. Beyond the data collection issues, the reliability of the methods becomes of utmost importance when a system is deployed in the real world.

First, the algorithms need to operate in real-time, using the computational resources available on the vehicle. Then, there is a general need to handle any unforeseen scenarios. This is a significant challenge, given the unpredictable nature of such situations. Objects belonging to unseen categories are particularly problematic [85]. There is a strong interest in robust generalization to unknown objects since autonomous vehicles need to handle them safely.

The variability in environmental conditions poses significant issues, too. Since the reliability of a sensor changes across different weather, seasons [160], and illumination conditions [86], it is challenging to deliver robust predictions across all adverse scenarios, such as with sun glare, rain, snow, or at night.

Furthermore, additional challenges occur from a methodological perspective depending on the input signal being used and the task of interest. With cameras, it is difficult to extract precise 3D information. This is relevant for various tasks, such as monocular 3D object detection and monocular depth estimation. Instead, with LiDARs, it is challenging to retain good performance at further distances, where the point clouds are sparser [163], or deal with the sensor artifacts in adverse weather [86]. Sensor fusion is also challenging since the modalities are heterogeneous [83].

Ambiguities in the predictions should be considered when predicting the trajectory of other traffic participants and planning that of the ego vehicle. Such ambiguities can be captured by estimating the uncertainty of the model associated with every prediction [82]. However, how to best use such information remains an open problem.

Additionally, given the expensive data collection issue, it is unrealistic to record large amounts of data worldwide to train location-specific models. Therefore, to a certain extent, there is a need to generalize robustly or adapt effectively to driving scenes from other countries, where street layouts and vehicles are different [162]. For the same reason, collecting data all over again may be overly complicated whenever the sensor configuration changes (e.g., due to a technology update). Therefore, another challenge is that of generalizing or adapting to different sensors [163].

Scene understanding methods must be capable of operating robustly even outside their training specifications. Instead, training and test sets often belong to the same data distributions [18, 35, 90, 290], making it problematic to test such abilities. In this dissertation, we propose to evaluate the robustness of a method by training it on one dataset and evaluating it on other datasets collected differently, thereby assessing the generalization capabilities of the model across extreme changes.

Moreover, the system needs to improve over time, becoming more robust thanks to the data collected by the vehicle itself and others in the fleet. Therefore, there are challenges regarding how to update the models. Continuous and active learning fields investigate this difficult problem [79, 268, 270, 271, 357].

Many methods focus on solving one task for one or more datasets at a time to achieve the best scores and claim state-of-the-art performance [24, 46, 94, 159, 367, 368]. However, they ignore critical issues that need to be solved to safely deploy an autonomous vehicle on open roads. Through a series of works focusing on various problems, this thesis aims to uncover and address some of these challenges via investigations and new methodologies that improve reliability with corner cases and challenging scenarios. Towards this end, new settings and experimental setups are proposed to test the models' reliability and ease the comparability for future works.

Part II

Methodology

Reliability

Contents

4.1	Overview	29
4.2	The Facets of Reliability	30
4.3	Robustness and Generalization	31
4.3.1	The Importance of Generalization	31
4.3.2	The Importance of Robustness	32
4.3.3	The Robustness-Generalization Trade-off	32
4.4	The Impact of Data	33
4.4.1	Data Biases	34
4.5	Challenging Conditions	35
4.6	Out-of-Distribution Data and Corner Cases	36
4.6.1	Unknown Objects	37
4.7	Uncertainty Estimation	38
4.8	Domain Adaptation and Domain Generalization	39
4.9	Adversarial Methods	40

4.1 Overview

In recent years, learning-based approaches, especially those employing deep neural networks, have achieved astonishing performance in various fields, ranging from computer vision [94, 109] and natural language processing [31] to autonomous systems [90] and healthcare [204]. However, a critical challenge undermines their widespread adoption, especially in safety-critical contexts: reliability.

Reliability in learning-based systems indicates the method’s capacity to consistently perform its intended functions under specified conditions and over time. This property is fundamental to avoid dangerous consequences in any safety-critical setting, such as medical diagnosis, medical robotics, and autonomous driving. Reliability is also important for less critical setups like augmented reality and mobile applications to deliver pleasant user experiences. Reliability encompasses multiple dimensions to ensure safe operations across various scenarios.

The crucial aspects of safety and operational efficacy determine the importance of reliable scene understanding in autonomous driving. Wrong predictions concerning the surrounding environment can lead to erroneous navigation, dangerous behaviors, and, in the worst cases, catastrophic outcomes. The societal, legal, and ethical implications of these errors demand an in-depth exploration and reinforcement of reliability in scene understanding for autonomous driving, motivating the research of this thesis.

This dissertation proposes various methods to improve the reliability of scene understanding models for autonomous driving, especially regarding robustness and generalization. The approaches introduced in Chapter 5 investigate this fundamental aspect across tasks and input signals, such as object detection, panoptic segmentation, and depth estimation, with cameras, LiDAR, and radar.

This chapter provides an overview of reliability with a focus on safety-critical applications and autonomous driving, thereby contextualizing the contributions of this work. Instead, specific related works are presented within each publication contributing to this thesis (Chapter 5).

4.2 The Facets of Reliability

Reliability in scene understanding can be seen from several perspectives, including accuracy, generalization, robustness, safety, and ethical assurance.

Accuracy is about the correctness and precision of a system in identifying, classifying, predicting, and ultimately understanding its inputs. Ensuring high accuracy in perception models is crucial for autonomous driving as it lays the foundations upon which decision-making algorithms such as path planning operate, directly impacting safety. Improving accuracy involves refining the methods delivering such predictions, including their architecture and training procedures. Incorporating diverse data from multimodal sensors, such as cameras, LiDARs [84], and radars [83] in the context of autonomous vehicles, enhances the system's ability to disentangle complex scenarios [83]. Accuracy is deeply intertwined with generalization, which is the ability to perform adequately with unseen inputs [316], an essential skill for any learning-based system (Section 4.3.1). In a multi-task setup [84], output consistency across the sub-tasks is crucial to ensure the stability of downstream components (Section 5.1).

Robustness denotes a model's ability to maintain its performance under various perturbations and uncertainties [100] (Section 4.3.2). In the context of autonomous driving, this includes variable lighting, weather, and dynamic scenarios. Due to the nature of learning-based systems and their dependency on the training data (Section 4.4), changes in appearance and anomalies pose substantial challenges to perception methods [135]. Both generalization and robustness are aspects of reliability that are of utmost importance for any safety-critical application, such as autonomous driving, to guarantee safe operations in all conditions. Section 4.3 explores the two characteristics and their relation.

For an autonomous car, safety means minimizing the occurrences that pose a risk to the vehicle, its occupants, and its surroundings. It is intrinsically tied to the reliability of scene understanding and subsequent decision-making algorithms, e.g., trajectory prediction and path planning. Integrating redundancy into perception systems and employing probabilistic models can mitigate risks associated with wrong predictions. Sensor fusion also plays an important role in this context [83]. Since autonomous vehicles move within the environment, perceiving their surroundings in 3D [84, 162] is crucial to avoid scaling and other issues due to monocular or camera-only setups. LiDARs and radars introduce redundancy to prevent dangerous consequences. Additionally, safety can be enhanced via formal methods to systematically verify and validate the behavior of perception methods under various scenarios [3, 4].



Figure 4.1. Examples of data from KITTI [90], BDD100K [354], nuScenes [35], and the Indian Driving Dataset [311]. The pioneering KITTI dataset includes mainly average cases around German residential areas. In contrast, the other, more recent datasets feature more challenging scenarios, such as difficult illumination conditions, large construction sites, and unstructured scenes. The differences highlight the domain gaps between the datasets and the diversity of traffic scenarios.

The ethical effects of perception models are commonly associated with their impact on privacy and algorithmic bias. The design of scene understanding systems must safeguard against unintentionally perpetuating biases and privacy violations [10]. Removing biases from the data before using it for training the models is crucial not only to address ethical concerns but also to improve accuracy and generalization [36] (Section 4.4).

4.3 Robustness and Generalization

4.3.1 The Importance of Generalization

Generalization is a central goal in machine learning. The success of a learning-based model is based on its ability to extend the knowledge acquired on the training set to the unseen examples from the same data distribution. This ability is crucial to use such a model in the real world, where variations from the training data are to be expected. In particular, generalization is highly influenced by the training data quality and the model’s design choices. The data should capture the entire distribution associated with the task (e.g., all types of cars when aiming to detect cars). However, that is difficult to achieve (Section 4.4). If the training data fails to represent the underlying distribution (e.g., it is missing high-performance sports cars), the model will likely face issues when dealing with missing or under-represented inputs (e.g., may not detect such sports cars). Examples of different data distributions are shown in Figure 4.1, highlighting the importance of generalizing to a wide range of autonomous driving scenarios.

Furthermore, depending on the complexity of the task, a model needs to be designed to understand the nuances of the data. While relatively accessible tasks such as handwritten digit recognition [161] can be addressed by relatively small models, more complex tasks such as panoptic segmentation [84, 150] need larger models. However, as model size and complexity increase, the risk of overfitting rises. Overfitting occurs when models memorize the training data rather than learn meaningful and transferable abstractions of it. Standard techniques such as data augmentation [212, 279], regularization [283], cross-validation [152], and early stopping [242] are essential to mitigate overfitting, thereby improving generalization.

In this dissertation, we focus on the extreme case of generalization to out-of-distribution data, which we aim to improve (Section 5.2). Out-of-distribution data is, by definition, outside the training data distribution, e.g., inputs captured by another sensor or in different settings (Section 4.6).

4.3.2 The Importance of Robustness

Robustness takes into account the model’s resilience to variations of the input [100]. These include noise, adversarial attacks, and distributional shifts. Similarly to generalization, robustness is crucial in real-world settings, especially in safety-critical applications, where models often encounter unseen and unexpected scenarios that deviate from their training data [110]. In a broader sense, a model is considered robust if it can withstand significant variations in its input. Robustness can be improved by overcoming the limitations of a model arising with certain types of input. Hence, methods exist to test the performance boundaries of a system and expand its capabilities by making it more robust.

Learning-based approaches are vulnerable to adversarial attacks [100], where small, carefully crafted perturbations of the input called adversarial examples lead to incorrect predictions. Even state-of-the-art models are not immune to these alterations [100]. Adversarial training [80] augments the training data with adversarial examples to improve the robustness of a model. Instead, certification methods [362] provide guarantees on the behavior of a model.

In this thesis, we investigate robustness against significant variations of the input as out-of-distribution data (Section 5.2). Additionally, we explore the root causes behind systematic failures of monocular depth estimation methods. We propose new techniques to overcome such limitations, rendering the models more robust across various conditions in input, e.g., at night or with dynamic scenes (Section 5.3).

4.3.3 The Robustness-Generalization Trade-off

A trade-off exists between robustness and generalization [286, 304]. Robustness techniques may penalize generalization by making the model overly conservative, leading to sub-optimal predictions on the original data [287]. Similarly, excessively regularizing a model for better generalization might weaken its robustness by narrowing its decision boundaries [361]. Addressing this trade-off requires careful design choices, delving into the underlying mechanisms that drive generalization and robustness. This leads to new hybrid approaches that balance these two critical aspects [191].

In this dissertation, we propose a set of techniques that strive to improve both robustness and generalization or to improve one without penalizing the other (Sections 5.2.1 and 5.2.2).

4.4 The Impact of Data

Data plays a pivotal role in the performance of every learning-based model. Specifically, the effectiveness of a machine learning method heavily depends on the diversity and representativeness of its training data. Datasets aim to represent a particular data distribution by including a variety of exemplary samples that should cover all scenarios. While non-trivial, harvesting the data generated with automatic simulations makes it possible to encompass a wide range of conditions that could cover most, if not all situations [63]. However, simulations introduce new issues around synthetic to real gaps, requiring additional techniques, e.g., domain adaptation [37, 121, 122, 230, 265, 313] (Section 4.8). When considering the variability of the real world, datasets often fall short of capturing its wide diversity [85, 162, 269], leaving a substantial part of it as out-of-distribution (Section 4.6).

Many researchers have been focusing on achieving state-of-the-art performance [130, 322, 330, 349], measured on one or more benchmarks comprising training and test data from the same distribution. However, there is a substantial gap between the data of these benchmarks [89, 90] and the real world, so there are no guarantees on the performance of these powerful models in real-world applications [91, 300]. This is due to the difficulty of capturing corner cases and rare occurrences as they naturally happen in reality [163]. For this reason, datasets tend to contain primarily average cases, with a limited number of corner cases [85, 162, 269].

Theoretically, this would not be an issue if the training data could fully capture the underlying distribution and the model could absorb and abstract its patterns. Foundation models have been gaining traction thanks to unprecedented generalization capabilities [31, 39, 151, 243]. They achieve this by harvesting extremely vast amounts of data with large models via highly complex training procedures. The training data is expected to be diverse enough to cover more scenarios than standard datasets [90, 175], enabling better generalization than average models.

Beyond the complicated methodological design, this raises challenges in collecting such comprehensive datasets and training the models. This requires significant data and hardware resources available to a limited number of institutions [31, 39, 151, 243]. Nevertheless, in dynamic and evolving real-world settings, such as traffic scenarios, a dataset is unlikely to adequately sample the entire long tail of the distribution (i.e., rare cases). Furthermore, datasets are collected at a certain point, so radically new designs of buildings, clothing, and vehicles from the future cannot already be part of it. Active and continual learning are techniques to help mitigate this problem [79, 357], but they have their own set of challenges as open fields of research.

Even powerful foundation models have limitations due to their training data not fully representative of the real world, so they struggle with out-of-distribution and long tail samples too [243]. The key difference is that the out-of-distribution data of foundation models is likely a subset of that of standard models trained on common datasets. In particular, the latter exhibit issues sooner when moving towards the long tails of the distribution.

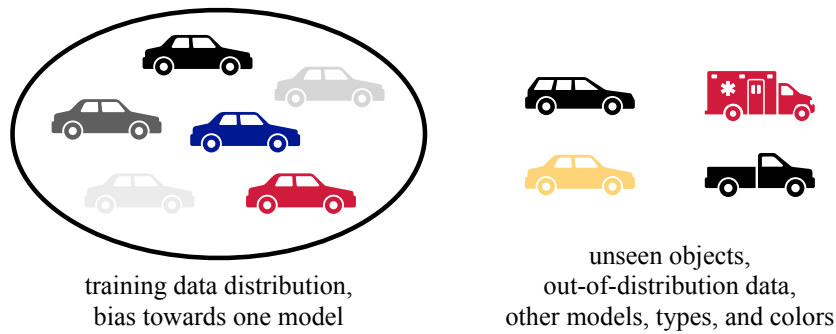


Figure 4.2. Example of possible data bias due to the vehicles present in a dataset. If the training data is not diverse enough for the model to abstract the characteristics of the objects and scenes of interest (e.g., contains only one car model), it may develop a bias towards the training data distribution, failing with other scenarios that are not part of it (e.g., with unseen colors and models).

This data issue is particularly severe for safety-critical applications, where the unrepresentative training data cannot be an excuse for dangerous behaviors. Therefore, it is crucial to design methods that work reliably not only with the training data distribution but also outside of it [85, 132, 269]. This enables robust predictions in all scenarios, regardless of the training data being a limited representation of the world. Towards this end, this dissertation proposes a set of techniques to expand the capabilities of existing methods beyond their training specifications (Section 5.2).

4.4.1 Data Biases

Data biases in machine learning refer to imbalanced distributions of data, which can lead to unfair, unethical, or inaccurate model predictions [148, 200]. Data biases can result from various factors and manifest in numerous ways, influencing different stages of the model's development and deployment [353]. An example of data bias is depicted in Figure 4.2.

If the data is not collected randomly or does not represent the actual distribution of the underlying population, it introduces sampling bias. Wrong or inconsistent labeling introduces errors and bias, too. Unequal representation of classes in the data can make models biased towards the majority classes, e.g., perform well on the common *car* class and underperform on the more rare *tram* class. This problem is very common for semantic segmentation. If certain features unfairly represent specific groups, classes, or outcomes, it can lead to biases. When certain demographic groups (e.g., based on gender, ethnicity, or age) are underrepresented in the dataset, it may lead to skewed model predictions towards overrepresented groups [105]. This is particularly concerning for medical imaging, where bias can lead to unfair diagnosis [65]. Furthermore, when trained on historical data, models may perpetuate existing biases from old outcomes. Other biases can arise by capturing the data with specific sensors.

Bias is particularly concerning in the context of datasets [105]. Therefore, researchers and engineers collecting and processing the data need extra care to iron out biases. In autonomous driving, research has been preferred toward addressing the tasks of seminal datasets, such as KITTI [89, 90], which contained biases towards relatively simple scenarios. For example, KITTI consists mainly of static and residential scenes captured during the day with good weather. However, considering all datasets are limited by definition, even when sampling dynamic scenes under various weather conditions and times

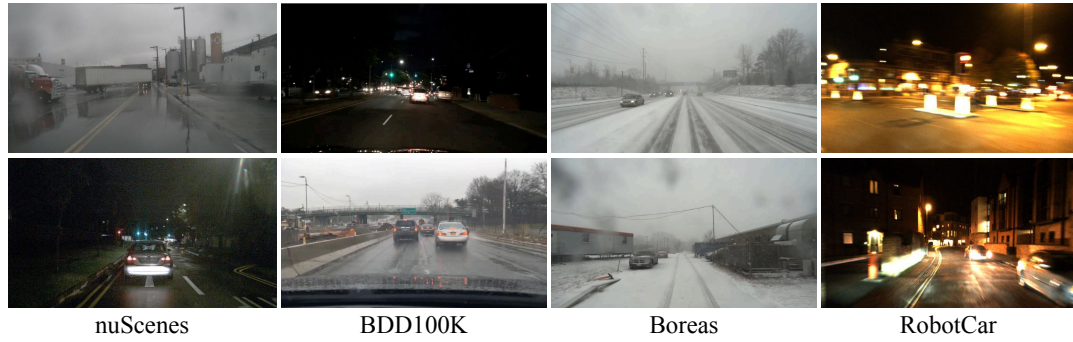


Figure 4.3. Example of challenging weather and illumination conditions from nuScenes [35], BDD100K [354], Boreas [34], and RobotCar [189]. Compared to standard illumination settings in clear or cloudy scenarios, the conditions shown pose a substantial challenge due to noise, blurriness, reflections, lack of texture, and hidden information in the dark or under snow. Additionally, with adverse weather, LiDAR sensors introduce artifacts that need to be accounted for (Section 5.3.2).

of the day, e.g., as in nuScenes [35], bias is inevitable towards the geographic regions in which the data was captured (e.g., Boston and Singapore for nuScenes).

Without extreme care and effort [105, 213], driving datasets collected by individual institutions [35, 89, 90] tend to be biased towards their geographical sphere of activity. The risk associated with this is that an autonomous vehicle is safe in specific regions where extensive data was collected, but it is unreliable and exhibits dangerous behaviors in other areas where the manufacturer had less interest or capacity to collect data. This raises significant concerns that need to be mitigated.

While taking care of biases during data collection and preparation is crucial, it is difficult to eliminate all data biases given the uncontrolled environments and geographic dependency in the context of autonomous driving. Therefore, the mitigating factor and last resort in terms of unbiasing must come from the methods learning from such data [353]. For example, improving the performance on out-of-distribution data, e.g., captured in other regions, helps to mitigate bias [82, 85, 162, 163, 269].

4.5 Challenging Conditions

Challenging conditions encompass a broad spectrum of scenarios that deviate from the well-defined settings often encountered during training. Methods may exhibit systematic failures with certain types of inputs due to their limitations in terms of training data or learning assumptions (Section 5.3). In safety-critical applications, such as autonomous driving, it is fundamental to design robust models to both standard and challenging conditions.

In autonomous driving, challenging conditions comprise a variety of scenarios that are not necessarily outside the training data distribution. For scene understanding, these conditions include adverse weather [35, 247], low illumination [189], and unstructured environments [311]. As shown in Figure 4.3, weather variations such as rain, snow, fog, and direct sunlight with glare can limit visibility, making it difficult to interpret the visual inputs adequately. As described in Section 3.3, adverse weather can severely disrupt the signals of a sensor, such as LiDAR [86]. Due to the overall darkness and noise,

low illumination (e.g., at nighttime) can be detrimental to the extraction of reliable visual features [86]. Highly unstructured environments present challenges as the models cannot find useful patterns in the training data. To a certain degree, out-of-distribution data and corner cases are problematic for every model (Section 4.6). Additionally, conditions considered challenging vary significantly depending on the task being addressed, the model, the input, and the training signal being used.

In this dissertation, we explore robust methods to overcome inherent issues of monocular depth estimation (Section 5.3). In this context, self-supervised methods learn based on assumptions, such as the camera moving in a static world. Violations of such training assumptions lead to systematic failures, preventing the safe applicability of such methods. For example, the presence of dynamic objects in the scenes violates the static assumption, leading to wrong depth estimates [83]. Furthermore, self-supervised depth estimation approaches suffer from severe performance degradation at nighttime, as darkness and noise prevent establishing correct pixel correspondences [86]. Instead, supervised monocular approaches learn artifacts from their ground truth sensor signal [86].

Moreover, we also explore new ways to mitigate the limitations of state-of-the-art approaches with unknown objects (Section 5.2.3) and inputs captured by different sensors in different countries (Sections 5.2.1 and 5.2.2), which are considered out-of-distribution data, as discussed in Section 4.6.

4.6 Out-of-Distribution Data and Corner Cases

Corner cases and out-of-distribution data are correlated as they refer to edge cases or outliers, representing samples that deviate significantly from the average observed during training. These cases tend to lie towards the extremes of the input space distribution (i.e., at the long tail), and they are helpful to test a model's ability to generalize. Specifically, they are characterized by their rarity or unique properties, making them less present in the training data than the average cases, which lie in the middle of the data distribution and are therefore widely explored.

In autonomous driving and other safety-critical applications, such as medical diagnosis, ensuring robustness against out-of-distribution data is particularly important to avoid dangerous behaviors. Still, the state-of-the-art often ignores out-of-distribution samples and corner cases and thus is likely to fail when exposed to inputs beyond the range of its training data [46]. Instead, a model that can deliver accurate predictions with such extreme cases is considered robust as it can adequately deal with a wide range of input variations (Section 4.3.2).

Out-of-distribution data can take many different forms. For example, it can be an input captured by another sensor, compared to that of the training data, or with different objects and settings, including unseen, unknown objects (Section 4.6.1). Nighttime inputs are out-of-distribution for a model trained on daytime data [82]. Analogously, inputs captured in the USA are out-of-distribution for a system trained on data collected in Germany [162, 331] because the street layouts and vehicles found in the USA are not part of the same distribution as the German ones.

The first step to tackle this problem is identifying the outliers at inference time. This enables flagging a limitation of the model and demands extra attention from a human operator or a downstream module of the same system (e.g., the planning component for autonomous driving). Identifying out-of-distribution

data is the task of outlier detection [118, 123, 184], which can be tackled via uncertainty estimation (Section 4.7).

Beyond identifying the issue, a different line of work investigates strategies to mitigate the problem directly [132, 179]. Apart from expanding the training data distribution by collecting more corner cases, various techniques aim to improve robustness against corner cases [163]. These robust methods are resilient to distributional shifts thanks to techniques such as domain generalization and domain adaptation (Section 4.8), adversarial training (Section 4.9), and zero-shot learning [32, 341, 342].

This is related to open-set approaches, which are purposely designed to work with potentially any scenarios [41, 85, 132, 269]. Open-set methods are designed to operate reliably both within the training data distribution and outside of it. For example, an open-set object detector could detect any object, including unknowns (Section 4.6.1).

This thesis proposes various methods to tackle this problem by improving the generalization to out-of-distribution data (Section 5.2). We do so by assessing the domain gap via uncertainty estimation (Sections 5.2.1 and 5.2.3) and addressing it with adversarial augmentations (Section 5.2.2) and shared latent representations (Section 5.2.3).

4.6.1 Unknown Objects

Unknown objects are a prominent example of out-of-distribution data. In this context, an unknown object can be any object that is not part of the learned categories. While some works consider an object unknown if it belongs to a category that is unlabeled yet part of the training data [127], others regard as unknown only those belonging to completely unseen categories [132, 179]. In this dissertation, we follow the latter.

Unknown objects pose significant challenges as they demand incredibly high levels of generalization, which is particularly difficult for learning-based approaches. Unconventional road users or unusual environmental elements can potentially lead to dangerous scenarios for an autonomous vehicle. Unknowns include unidentified, damaged parts lying on the road following an accident or animals that are not part of the training data.

For autonomous driving, due to the wide variability of the surrounding environment, it is important to design a method that can operate safely in all conditions, including in the presence of completely unseen, unknown objects [85, 269]. This is crucial because datasets cannot include all possible scenarios (Section 4.4), so there will always likely be objects that are not part of the training data.

Zero-shot [32, 42] and open-set [132, 356] approaches can work with unknown objects. Nonetheless, zero-shot methods often depend on external knowledge, e.g., a language model [243], which is employed to establish shared frameworks encompassing both familiar and unfamiliar object categories. Specifically, this external knowledge comes from a system that is familiar with the unknown objects, making them essentially known [360]. A foundation model is suitable for providing such external knowledge as it has been trained on vast data. For example, a system that has never seen a zebra may be able to identify it by knowing the concepts of horses and stripes and being injected with the information that zebras look like horses with stripes. Open-vocabulary approaches identify unknown

objects with vision-language models [293], which have learned associations between text and image tokens, including those for such unknown objects.

Anomaly detection [20, 21] and uncertainty estimation [132, 179] have been shown as promising strategies to identify the presence of unknown objects and regions in the input. However, it is still unclear how unknown individual objects in a scene can be found without prior knowledge, leaving this highly difficult problem relatively unexplored.

In this dissertation, we propose a new setting to segment completely unseen, unknown objects and known areas without prior knowledge about unknowns (Section 5.2.3). We term this setting holistic segmentation and propose a first method to address it [85].

4.7 Uncertainty Estimation

Uncertainty estimation plays a crucial role in enhancing the reliability and safety of any learning-based model [82, 117, 119, 211, 239, 254, 269, 310]. It does so by providing a measure of confidence in the decisions made by the system. In autonomous driving, uncertainty estimation is commonly regarded as a fundamental pillar for making informed and safe driving decisions [88, 143]. Uncertainty in machine learning is called *epistemic* when related to the model itself or *aleatoric* if connected to the input.

Scene understanding tasks, such as object detection and semantic segmentation, inherently involve ambiguities due to occlusions, varying illumination conditions, sensor noise, and the presence of unknown objects. By providing adequate uncertainty estimates to downstream tasks, autonomous vehicles can make better decisions when encountering challenging scenarios or unfamiliar objects, such as opting for more conservative maneuvers in high-uncertainty settings.

While various learning-based methods provide confidence values alongside their predictions, these do not reflect the actual probability of delivering a correct answer [88]. This behavior is related to the objective functions employed during training. Commonly used for classification and segmentation models, the *softmax* and cross-entropy loss lead to a model that outputs wrong predictions with high confidence. This is due to *softmax* highly promoting the probability of the highest *logit* at inference time and the cross-entropy loss strongly penalizing predictions with low confidence at training time. Therefore, specific methods have to be developed to deliver adequate confidence values.

Gawlikowski et al. [88] proposed a categorization of uncertainty estimation techniques, which we follow to provide an overview. Single deterministic approaches use a deterministic model to deliver both predictions and uncertainty estimates [275], or they compute uncertainty externally [245, 310]. An intuitive way to do so is by measuring the inference sample deviation from the training data [310]. Instead, Bayesian methods infer the probability distribution of the parameters of models that are not deterministic [78, 88].

Ensemble approaches consist of multiple models trained with different initializations or architectures [157]. The uncertainty is estimated by combining their predictions. Test-time augmentation methods also compute the uncertainty through the combination of multiple outputs. Still, in this case, a single deterministic model delivers multiple predictions given different augmented versions of the same input [324].

In this dissertation, we employ uncertainty estimation for domain generalization to unseen illumination conditions, countries, driving scenarios, and sensor configurations (Section 5.2.1). Furthermore, we use uncertainty to improve generalization and robustness against unseen, unknown objects, serving to differentiate between known and unknown (Section 5.2.3).

4.8 Domain Adaptation and Domain Generalization

The ability to generalize to unseen scenarios is one of the pillars defining a successful machine learning model and, ultimately, an intelligent agent (Section 4.3.1). Building reliable and safe autonomous driving systems comprises the challenge of ensuring that the algorithms can perform robustly and effectively across various environmental and traffic scenarios. This problem is closely linked to the concepts of domain adaptation and domain generalization. While autonomous driving models are typically trained on large-scale annotated datasets, the real-world conditions often vary significantly from the statistical properties of their training sets (Section 4.4). This discrepancy can lead to catastrophic failures in model performance and, consequently, safety risks.

Domain adaptation involves adapting a model trained on a source domain to perform well on a different yet related target domain [121, 122, 256, 331]. This adaptation is enabled by access to helpful information about the target domain, and the domain adaptation method needs to make the best use of this information. In autonomous driving, the source domain may consist of data collected in a specific geographic area or under particular weather conditions. In contrast, the target domain may involve a new city, different lighting conditions, or varying traffic densities. Furthermore, source data may be synthetic, and target data may be real. When performing object detection, the available information about the target domain could include, for example, the size of the vehicles compared to that of the source domain [331].

A straightforward approach to mitigate the domain gap is fine-tuning the model on a small subset of annotated data from the target domain [331]. Compared to training entirely on the target data, this allows for labeling fewer samples by bootstrapping a model that has already learned the task in related data. Yet, annotating data is problematic, and it may be challenging to estimate how many samples from the target domain are needed for a seamless transfer of the model.

To circumvent this, unsupervised domain adaptation focuses on adapting the model without the need for annotating the target data [121, 122, 138, 265, 313], making it particularly attractive. These unsupervised techniques align the feature distributions of the source and target domains such that a model trained on the source domain generalizes well to the target domain. Generative Adversarial Networks (GANs) can be used to generate target-like source data, thereby bridging the domain gap [99]. Self-ensembling leverages the model itself by using its predictions on the target domain as pseudo-labels to guide the adaptation [76]. Other techniques include optimal transport [51], Domain-Adversarial Neural Networks [80], and maximum mean discrepancy [348].

Unlike domain adaptation [121], domain generalization aims to build models that are inherently robust to domain shifts without requiring any target domain data [103, 166, 167, 260, 326]. The objective is to build models capable of generalizing to new, unseen environments. Therefore, domain generalization uses no information about the target domain. This makes it more challenging than domain adaptation yet more applicable, as target data may be unavailable until the model is deployed in the real world.

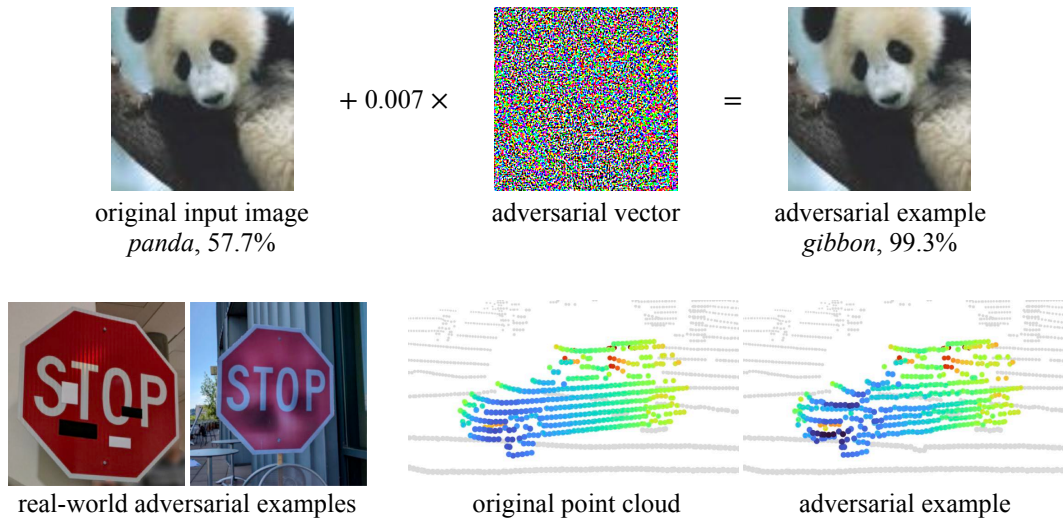


Figure 4.4. Adversarial examples on images and point clouds. At the top, an adversarial pattern is added to the image of a panda, fooling the classifier into predicting a gibbon with high confidence despite the images looking identical to the human eye (adapted from [100]). At the bottom left, physical adversarial patches and paint are applied on stop signs [67], disguising the stop signs as other signs, such as speed limits, according to a model trained to classify signs. At the bottom right, the points of the LiDAR point cloud of a car are altered adversarially while preserving plausibility [163], hiding the car as a different object for a 3D semantic segmentation model.

This is possible by enabling a model to learn domain-invariant features that remain informative across different domains.

Domain generalization strategies include meta-learning techniques, which learn on multiple source domains [72], invariant risk minimization, which learns invariant features [8], and data augmentation [162].

In this thesis, we propose a method using uncertainty estimation to assess and mitigate the domain gap for domain generalization (Section 5.2.1). Then, we introduce an approach performing adversarial data augmentation to increase the diversity of the training data for domain generalization (Section 5.2.2). Furthermore, we use a GAN to aid the unsupervised adaptation of a model across challenging illumination and weather conditions (Section 5.3.2).

4.9 Adversarial Methods

Adversarial methods have emerged as a powerful tool in developing machine learning models, specifically to address robustness-related issues.

Adversarial attacks manipulate inputs in a manner designed to fool machine learning models [100, 191, 214, 292]. As in Figure 4.4, such manipulations can be imperceptible to humans yet lead to significant misclassifications or incorrect scene interpretations, constituting a substantial safety concern in autonomous driving. The new inputs created by these perturbations are called adversarial examples. Adversarial attacks expose the weaknesses of a model, which can then be addressed to improve its robustness.

GANs [99] consist of a generator and a discriminator that play a min-max game. The generator tries to produce indistinguishable data from real data, while the discriminator aims to tell them apart. GANs have been used for data augmentation [7], creating synthetic scenarios that help in domain adaptation [80].

Adversarial training, where the model is trained on both the original and adversarial examples, can build resilience against adversarial attacks and improve robustness in general [276, 336]. However, this procedure is particularly expensive as it requires training a model and generating adversarial examples against it iteratively. Adversarial augmentation is less costly, as it requires training only two models: one to generate the adversarial examples and the other to be trained with such adversarial examples as augmentations [163, 364]. For autonomous driving and other safety-critical applications, this is crucial for expanding the training data with scenarios that may not be sufficiently present due to the data collection limitations (Section 4.4).

In autonomous driving, significant concerns exist around adversarial methods. While attackers may not have access to the models deployed in autonomous vehicles needed for crafting adversarial examples, it has been observed that adversarial examples are often not strictly limited to the models against which they were generated [185, 292, 301]. In settings where the task and the training data are identical, adversarial examples can also frequently fool other models. More surprisingly, there is evidence to suggest that adversarial examples may have a degree of transferability even when the models are trained for different tasks or on different datasets. It was shown that it is possible to generate universal adversarial examples that can fool networks across different tasks [206].

While attackers may not be able to alter the digital inputs of autonomous vehicles, they could manipulate the real scene to fool perception models [38, 67]. This is shown, for example, in Figure 4.4 by applying stickers on stop signs, disguising them into different signs [67] or 3D printing adversarial cones filled with cement fooling camera and LiDAR systems to look invisible [38]. Such perturbations could lead to dangerous consequences, raising significant concerns about the safety of learning-based models. Therefore, it is of utmost importance to understand and address the vulnerabilities of a model before it is deployed to the real world. Designing robust models is crucial for the safe applicability to autonomous driving settings.

In this dissertation, we propose an adversarial augmentation technique to improve a model's robustness and generalization to out-of-distribution data, including damaged and rare vehicles (Section 5.2.2).

Summary of Contributions

Contents

5.1	Consistent Scene Understanding	43
5.1.1	End-to-End Panoptic Segmentation (RA-L 2021)	44
5.2	Generalizing to Out-of-Distribution Data	57
5.2.1	Domain Generalization with Uncertainty Estimation (RA-L 2021)	57
5.2.2	Domain Generalization with Adversarial Augmentations (CVPR 2022)	69
5.2.3	Segmentation of Known and Unknown Objects and Regions (ICCV 2023)	83
5.3	Improving Robustness for Depth Estimation	101
5.3.1	Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)	101
5.3.2	Monocular Depth Estimation in Challenging Conditions (ICCV 2023)	115

This chapter summarizes the main contributions of this dissertation and provides the publications associated with each work. Specifically, Section 5.1 introduces a method to improve the consistency across tasks and outputs. The proposed approach is the first end-to-end panoptic segmentation method, delivering consistent predictions from a single model. Section 5.2 presents three techniques to address out-of-distribution data. Each focuses on a different task and aspect of generalization and robustness, from generalizing to different countries and weather conditions to unknown objects. Then, Section 5.3 focuses on monocular depth estimation, proposing two methods to overcome critical inherent limitations of existing approaches, namely the difficulties with dynamic objects and challenging weather and illumination conditions, which violate common training assumptions.

5.1 Consistent Scene Understanding

As discussed in Chapter 3, scene understanding is a fundamental pillar for any autonomous agent. Specifically, for the agent to interact with the environment, 3D understanding of the surroundings is of utmost importance. So, in this section, we focus on 3D perception.

In multi-task settings, reliability issues may arise when related outputs contrast with one another, e.g., object detection with instance segmentation or semantic with instance segmentation. Panoptic segmentation merges two important perception tasks typically treated separately: semantic and instance segmentation [150]. When separate models are dedicated to predicting related masks, conflicts in the output easily arise due to their contrasting predictions, posing a challenge to downstream modules and undermining the reliability of the entire system. Panoptic segmentation is helpful as it merges the two tasks, but it has been addressed only by combining the outputs of several sub-tasks, heavily relying on heuristics-based merging modules. To mitigate this problem, in this section, we take a step towards coherent and consistent 3D scene understanding, proposing the first end-to-end method for panoptic

segmentation, delivering consistent outputs from a single model, eliminating the need for complex merging modules.

5.1.1 End-to-End Panoptic Segmentation (RA-L 2021)

Panoptic segmentation [150] unifies the tasks of semantic and instance segmentation. By doing so, panoptic segmentation imposes to rethink the two tasks and leads to more efficient perception systems with a more complete understanding of their surroundings. Instead, if the two tasks are addressed separately with dedicated models [186, 252], as done previously, their outputs could easily lead to inconsistencies, e.g., for parts of instances not considered objects by the semantic model.

In theory, panoptic segmentation mitigates this problem. However, at the time of this work, existing solutions addressed panoptic segmentation by solving a series of sub-tasks distinctly and merging their outputs, inherently introducing conflicts and artifacts. As shown in Figure 5.1, approaches could be distinguished into two categories: proposal-based, also called top-down, or proposal-free, also known as bottom-up. The former group is multi-stage: methods find instance masks with Mask R-CNN [108], perform semantic segmentation, and then have to solve the conflicts between the two outputs to merge them into the panoptic masks [345]. Instead, proposal-free approaches are single-stage: they segment the scene semantically and cluster the instances within the predicted object segments [46, 351].

While proposal-free methods do not have conflicts to solve, at the time of this work, they all relied on a cumbersome subsequent grouping stage to associate input pixels or points into instances, as highlighted in Figure 5.1. The state-of-the-art for the imaging domain at the time, Panoptic-DeepLab [46], predicts three outputs: semantic segmentation, instance center prediction, and instance center regression. Nevertheless, the fundamental step to transition from detected object centers to segmented instances consists of grouping the object pixels to the closest instance centers. With overlapping instances, this is prone to introduce artifacts in the final output. Other works relied on different proxy tasks to achieve instance segmentation, such as detecting bounding boxes [149].

Instead, in the 3D domain, panoptic segmentation had just been enabled by adding instance-level annotations to the SemanticKITTI dataset [18, 19]. A handful of approaches extended popular methods from the image domain to point clouds [202], but none could outperform baselines consisting of an object detector and a semantic segmentation model [19]. This is due to LiDAR point clouds being unstructured, unordered, sparse, and irregularly sampled, unlike images, demanding new solutions. For these reasons, we applied the proposed approach to LiDAR point clouds, achieving state-of-the-art performance.

In this work, we propose Panoster, the first end-to-end panoptic segmentation method. In this context, as shown in Figure 5.1, end-to-end means providing instance segments with IDs as well as semantic segments directly from the same neural network. By avoiding costly grouping stages, sub-tasks, or conflict resolutions, Panoster delivers extremely fast and consistent panoptic predictions through a single forward pass, only 5% slower than an equivalent semantic segmentation model.

Essentially, we achieve this end-to-end by instilling the grouping capabilities of a clustering technique directly into the model weights. This enables the network to output instance segments directly without further steps. Towards this end, we employ a combination of loss functions targeted at learning to

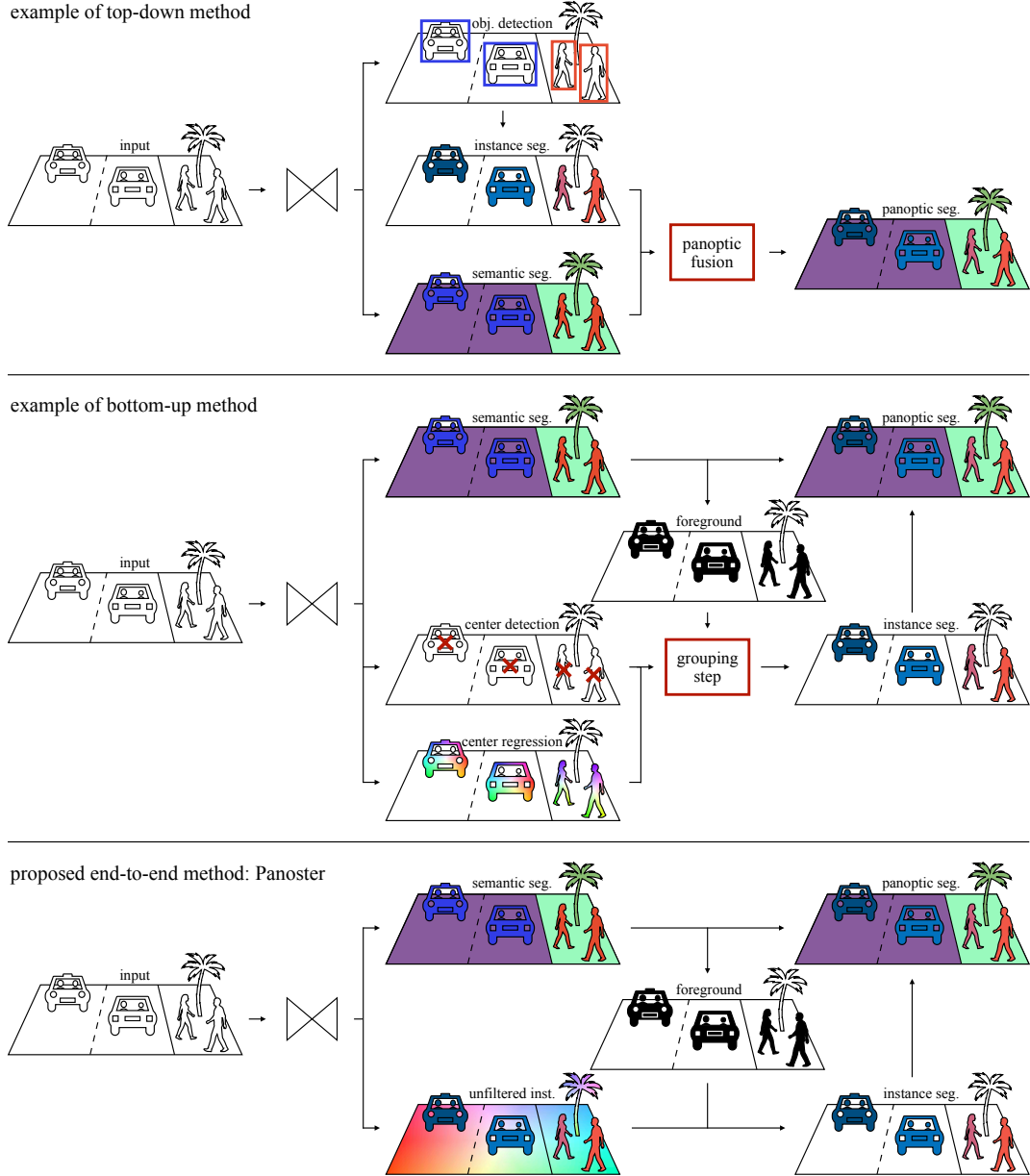


Figure 5.1. Schematic comparison of existing panoptic segmentation approaches with the proposed Panoster [84]. Both top-down [345] and bottom-up [46] approaches break down the panoptic task into sub-tasks, such as object detection or center regression. In the diagrams, red boxes highlight critical modules needed to aggregate such sub-tasks. Panoptic fusion aims to resolve conflicts between the intermediate predictions for top-down methods. For bottom-up approaches, the grouping step is responsible for the instance segmentation output, turning intermediate estimates into instance masks. Instead, Panoster simplifies the bottom-up paradigm in an end-to-end fashion. Panoster eliminates the need to resolve conflicts or take critical grouping steps since the instance IDs are delivered directly from the network.

cluster the points into instances. We initially proposed these losses to cluster radar pulses according to their emitting source [87] in the context of Stefano’s Master thesis. We repurpose them here for instance segmentation within our end-to-end panoptic framework.

Panoster computes instance IDs for each point in input, including those belonging to non-object classes (i.e., *stuff*). Simultaneously, the model outputs a semantic class for each point. The semantic

information is then used to discard the instance ID for the *stuff* points by masking according to the object classes. As done by a few later works, this masking step could be skipped by considering valid only the instance segments corresponding to object classes (i.e., *things*) without changing the essence of the method.

The clustering method we proposed in [87] is the first to enable a neural network to do clustering not only for feature extraction but also for grouping the inputs directly, making it a good candidate for end-to-end panoptic segmentation. It works by optimizing two contrasting aspects of clustering performance: non-fragmentation and purity of the clusters. Specifically, we use a dedicated loss function for each aspect and optimize them jointly. Conversely, prior learning-based clustering works used a model to extract features and fed these features to a traditional data analysis clustering approach.

At the time, we associated Panoster with bottom-up approaches as they were the most similar. The difference is the immediacy with which we obtain instance IDs, i.e., directly from the network. Instead, we could have placed Panoster in its own end-to-end category, as later works did [325]. However, it should be noted that the terms end-to-end and unified had been used by earlier panoptic segmentation methods to indicate that the output was produced by a single model instead of two, yet they relied on costly grouping stages or sub-tasks executed in series [178, 345].

Nevertheless, Panoster is the first panoptic segmentation approach delivering instance and semantic segments directly from the network in a single forward pass, hence the term end-to-end. Other methods have been proposed as end-to-end or unified, including highly successful methods in the imaging domain, such as the concurrent MaX-DeepLab [325], released about a month later. MaX-DeepLab achieves this with a transformer to output non-overlapping masks and semantic labels optimized with a loss maximizing the reference metric (i.e., PQ).

In the LiDAR domain, Panoster is also the first to consume 3D points directly, avoiding projections to 2D, and the first to outperform the strong baselines proposed by Behley et al. [19] which combine an object detector with a semantic segmentation model, thereby pushing the task forward.

The following includes the letter, while the manuscript's license can be found in Appendix C.

Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds

Stefano Gasperini^{1,2} Mohammad-Ali Nikouei Mahani² Alvaro Marcos-Ramiro²
Nassir Navab^{1,3} Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

IEEE Robotics and Automation Letters (RA-L), February 2021

Summary. Panoptic segmentation has recently unified semantic and instance segmentation, thus enhancing reliability by merging previously distinct tasks, which can lead to contrasting outputs. However, existing panoptic segmentation approaches still handle the tasks separately while training for them jointly. This leads to the necessity of several steps to group pixels or points into objects or the resolution of conflicts between the different outputs. In this paper, we address this by presenting Panoster, a novel proposal-free panoptic segmentation method that outputs semantic and instance estimates directly from the network. We achieve this with a simplified framework incorporating a learning-based clustering solution to identify instances. We apply Panoster on LiDAR point clouds. At inference time, this acts as a class-agnostic segmentation, allowing Panoster to be fast while outperforming prior methods in terms of accuracy.

Contributions. I initiated this project and executed it. I proposed to perform instance segmentation directly with the model via integrating a learning-based clustering, reusing the insights from my Master's thesis [87], and I implemented it. I then saw the advantage this end-to-end solution could bring to panoptic segmentation, so I designed and implemented the proposed Panoster. I defined and ran the experiments, drove the analysis and discussions, wrote the manuscript, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: <https://doi.org/10.1109/LRA.2021.3060405>.

Copyright Statement. ©2021 IEEE. Reprinted, with permission, from Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari, 'Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds', IEEE Robotics and Automation Letters (RA-L), February 2021.

The license can be found in Appendix C.

Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds

Stefano Gasperini^{1,2}, Mohammad-Ali Nikouei Mahani², Alvaro Marcos-Ramiro²,
Nassir Navab^{1,3}, and Federico Tombari^{1,4}

Abstract—Panoptic segmentation has recently unified semantic and instance segmentation, previously addressed separately, thus taking a step further towards creating more comprehensive and efficient perception systems. In this paper, we present Panoster, a novel proposal-free panoptic segmentation method for LiDAR point clouds. Unlike previous approaches relying on several steps to group pixels or points into objects, Panoster proposes a simplified framework incorporating a learning-based clustering solution to identify instances. At inference time, this acts as a class-agnostic segmentation, allowing Panoster to be fast, while outperforming prior methods in terms of accuracy. Without any post-processing, Panoster reached state-of-the-art results among published approaches on the challenging SemanticKITTI benchmark, and further increased its lead by exploiting heuristic techniques. Additionally, we showcase how our method can be flexibly and effectively applied on diverse existing semantic architectures to deliver panoptic predictions.

Index Terms—Deep Learning for Visual Perception, Computer Vision for Transportation, Panoptic Segmentation, LiDAR Point Clouds.

I. INTRODUCTION

SCENE understanding is a fundamental task for autonomous vehicles. Panoptic segmentation (PS) [1] combines semantic and instance segmentation, enabling a single system to develop a more complete interpretation of its surroundings. PS distinguishes between *stuff* and *thing* classes [1], with the former being uncountable and amorphous regions, such as *road* and *vegetation*, and the latter standing for countable objects, such as *people* and *cars*. The two have been addressed separately for decades, leading to significantly different approaches: *stuff* classifiers are typically based on fully convolutional networks [2], while *thing* detectors often rely on region proposals and the regression of bounding-boxes [3]. Tackling semantic and instance segmentation jointly with PS allows to save costly and limited resources [4].

Since the introduction of PS, several approaches have been proposed to address this new task. While most focused on

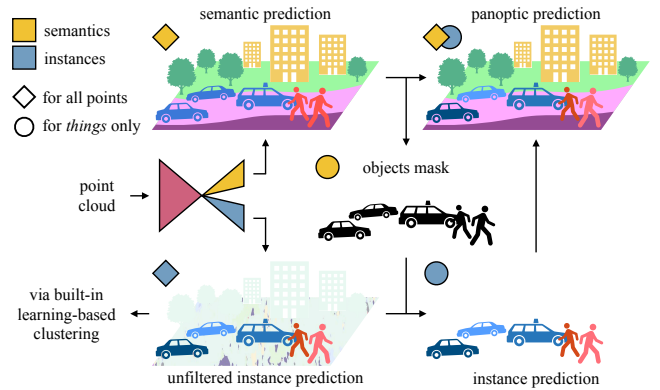


Fig. 1. Proposed Panoster framework. A semantic class and an instance ID are predicted for every point, then filtered and combined for panoptic segmentation.

images [4–6] or RGBD data [7, 8], only a handful of methods used LiDAR point clouds [9, 10], none of which has managed to outperform systems combining separate networks for semantic segmentation and object detection [11]. LiDAR sensors have proven to be particularly useful for self-driving cars, capturing precise distance measurements of the surrounding environment. Segmenting LiDAR point clouds is a crucial step for interpreting the scene. Compared to image pixels, LiDAR point clouds pose various challenges: they are unstructured, unordered, sparse, and irregularly sampled.

In this paper, we introduce a novel proposal-free approach for panoptic segmentation of LiDAR point clouds. We name our method Panoster, standing for panoptic clustering. Panoster brings panoptic capabilities to semantic segmentation networks by adding an instance branch trained to resolve objects with a learning-based clustering method adapted from [12]. Unlike existing proposal-free approaches [6, 10], Panoster, thanks to its clustering solution integrated in the model, does not need further groupings to form objects, as it outputs instance IDs straight away from the network. The contributions of this paper can be summarized as follows:

- We introduce a new proposal-free approach for panoptic segmentation, fully end-to-end, flexible and fast, with a small overhead on top of an equivalent semantic segmentation architecture.
- To the best of our knowledge, for the first time in panoptic segmentation, we exploit an entirely learning-based end-to-end clustering approach to separate the instances, resulting in a fully end-to-end panoptic method.

Manuscript received: October, 15, 2020; Revised December, 30, 2020; Accepted January, 28, 2021.

This paper was recommended for publication by Editor Cesar Cadena Lema upon evaluation of the Associate Editor and Reviewers comments.

¹ Stefano Gasperini, Nassir Navab, and Federico Tombari are with the Faculty of Computer Science, Technische Universität München, 85748 Garching bei München, Germany. stefano.gasperini@tum.de

² Stefano Gasperini, Mohammad-Ali Nikouei Mahani, and Alvaro Marcos-Ramiro are with the BMW Group, 80788 München, Germany.

³ Nassir Navab is also with the Department of Computer Science, Johns Hopkins University, 21218 Baltimore, MD, USA.

⁴ Federico Tombari is also with Google, 8002 Zürich, Switzerland.

Digital Object Identifier (DOI): 10.1109/LRA.2021.3060405.

- Our novel instance segmentation acts as class-agnostic segmentation, has no thresholds, and delivers instance IDs directly, without requiring further grouping steps.

We apply Panoster on the challenging task of panoptic segmentation for LiDAR 3D point clouds, outperforming in a fraction of the time state-of-the-art methods. Furthermore, we show the flexibility of Panoster by deploying it on both projection-based and point-based architectures, making it the first panoptic method to consume LiDAR 3D points directly.

II. RELATED WORK

In this section, we provide a brief overview of existing approaches, and highlight overlaps and differences with Panoster.

A. Semantic Segmentation

Depending on the input representation, semantic segmentation methods can be grouped into projection-based and point-based. Projection-based approaches eliminate the challenging irregularities of 3D points by projecting them into an intermediate grid. This grid representation can be made of voxels [13, 14] or pixels [15, 16], allowing the use of 3D or 2D convolutions respectively. While the former projection tends to be highly inefficient for this task, the latter can count on state-of-the-art methods from image segmentation, albeit discarding valuable geometric information by transitioning from 3D to 2D.

PointNet [17] pioneered point-based methods and deep learning on point clouds. They used multilayer perceptrons (MLP) to learn features from each point, then aggregated with a global max-pooling. Additional MLP-based architectures were proposed in [18–20]. More recent approaches, such as KPConv [21] and SpiderCNN [22] have focused on creating new convolution operations for points. KPConv [21] extended the idea of 2D image kernels to 3D points, by defining an operation which derives new 3D points from the multiplication of input 3D point features with flexible kernel weights. Recently, hybrid approaches have been developed [23, 24], combining point and projection-based solutions.

Our method Panoster extends semantic segmentation approaches to panoptic segmentation. In this paper we apply Panoster on KPConv [21] and SalsaNext [16], representative of point-based and projection-based categories respectively.

B. Instance Segmentation

Instance segmentation methods can be divided into detection-based and clustering-based. The former was established in the image domain by Mask R-CNN [25] which has two stages: region proposals are extracted, refined into bounding boxes, and segmented with a pixel-level mask to identify the objects. Mask R-CNN has catalyzed a variety of methods, some of which address panoptic segmentation [4, 9, 26], as described in Section II-C. Clustering-based approaches learn an embedding space in which instances can be easily clustered [27, 28]. Therefore, the segmentation is typically done by deploying a data analysis clustering technique, such as DBSCAN [29], on the latent features, which were learned

to aid the separation. These methods were extended to point clouds in [30–32].

Panoster operates differently, by clustering object instances thanks to loss functions optimizing the network outputs directly. Although our instance segmentation is learning-based, it does not explicitly rely on the optimization of a latent space via embeddings, differing from clustering-based methods described above. Furthermore, Panoster performs panoptic segmentation, resulting in a more comprehensive approach.

C. Panoptic Segmentation

PS methods can be divided into two categories: proposal-based and proposal-free. The vast majority are from the image domain, of which the larger stake are proposal-based or top-down methods (e.g. UPSNet [26], AdaptIS [33], Seamless [5], EfficientPS [4]): two-stage approaches, which are usually based on the well-known Mask R-CNN [25]. They rely on the detection of *things* first, followed by a refinement step and a semantic segmentation of *stuff*. Their main drawbacks are the needs of resolving mask overlaps and conflicts between *thing* and *stuff* predictions. Instead, proposal-free or bottom-up methods are single-stage and follow the opposite direction. They segment the scene semantically, and cluster the instances within the predicted *thing* segments. DeeperLab [34] was the first of this kind, regressing objects center and corners from images. SSAP [35] used a cascaded graph partition module to group pixels into instances according to a pixel-pair affinity pyramid. Panoptic-DeepLab [6] simplified this concept by predicting the instance center locations, and using a Hough-Voting scheme to group the pixels to the closest center.

While PS received a lot of attention in the image domain, PS on LiDAR point clouds is yet broadly unexplored, due to the only recent introduction of a suitable public dataset, i.e. SemanticKITTI with panoptic labels [11]. In [11] strong baselines combined state-of-the-art semantic segmentation methods KPConv [21] or RangeNet++ [15], with the object detector PointPillars [20]. PanopticTrackNet [9] followed the approach of EfficientPS [4] and extended it to consume LiDAR point clouds: it is top-down, based on Mask R-CNN [25] to resolve the instances. Additionally, a new tracking head delivers temporally consistent instance IDs. The only existing bottom-up method for LiDAR point clouds is [10], which we denote as LPSAD. It is based on semantic embeddings and, as in Panoptic-DeepLab [6], the regression of offsets to the object centers, then instances are extracted by an iterative clustering procedure, grouping points according to their center offset prediction. To date, no method managed to outperform the combined baselines proposed in [11].

Panoster is significantly different from existing panoptic approaches. It is the first to use an entirely learning-based clustering technique to resolve the instances. As LPSAD [10], our method is bottom-up and applied on LiDAR point clouds, but compared to it, our instance branch outputs instance IDs directly, without requiring any subsequent grouping steps, rendering it fully end-to-end trainable.

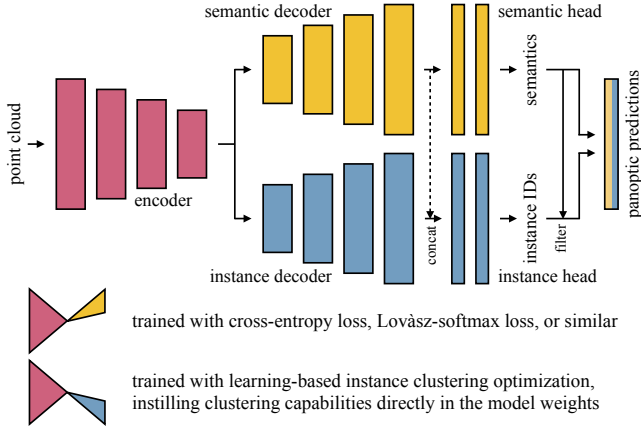


Fig. 2. Panoster simplified architecture. The input is raw 3D points or a 2D spherical projection. During training, gradients do not back-propagate through the dashed line. At inference time, both decoders and heads process data in the same way.

III. PANOSTER

A. Overview

As described in Section II-C, current proposal-free panoptic approaches [6, 10] pair a network with an external grouping technique to form object instances. The latter exploits the network predictions, and is required in order to output any meaningful instance IDs. With the proposed Panoster, we simplify by removing this extra step, and incorporating the clustering in the network itself. We achieve this with an entirely learning-based approach, adapted from the signal processing domain [12], to output instance IDs directly, for each point given in input (Section III-C). To obtain panoptic predictions, it is then sufficient to filter out these IDs for all *stuff* points (Section III-D), as shown in Fig. 1. Thanks to custom loss functions, our integrated clustering method instills grouping capabilities right in the model weights. At inference time, these layers process data in the same way as their semantic segmentation counterparts, while pursuing the rather different task of instance segmentation. In Fig. 2 we showcase Panoster general architecture, highlighting how each part is trained.

B. Architecture

Panoster consists of the following modules: a shared encoder, decoupled symmetric decoders and heads for the two tasks, a mask-based fusion and optional post-processing.

Motivated by [6, 9, 10], we opt for a shared encoder with two separate decoders, each developing its task starting from common features. Following Fig. 2, the backbone, the semantic decoder and head follow the design of a standard semantic segmentation network, consuming a full 360 degree LiDAR point cloud. In this paper we apply Panoster on KP-Conv [21] and SalsaNext [16]. We refer to the two variants as PanosterK and PanosterS, respectively. Therefore, depending on the configuration, our framework can be fed with raw 3D points or with a 2D spherical projection.

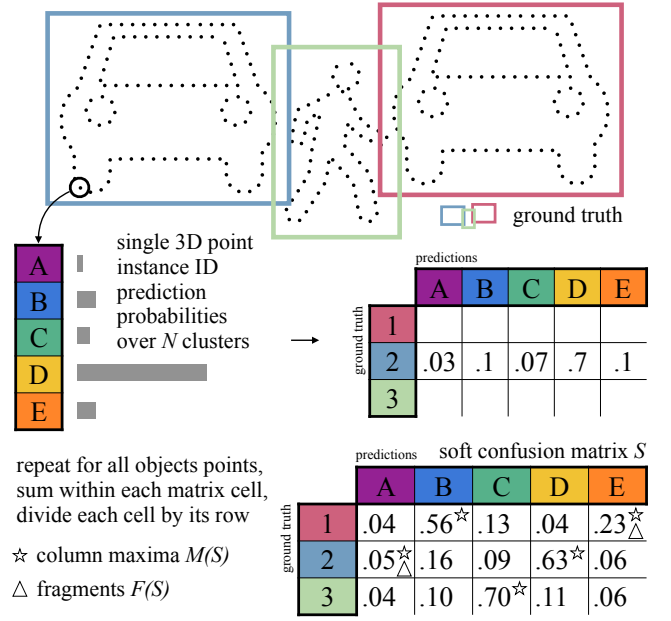


Fig. 3. Construction of the matrix used for the computation of the instance clustering loss functions. The matrix S is based on the instance branch segmentation probabilities, which are the cluster assignment probabilities. The figure shows a simplified scene with 3 instances and a model with $N = 5$. Fragments $F(S)$ and all non-column maxima cells $Q(S)$ are minimized by fragmentation and impurity losses respectively.

As shown in Fig. 2, our method has a second decoder and head, dedicated to instance segmentation. From the architecture perspective, the two branches are identical, except for the size of the last layer: the semantic branch output size depends on the number of semantic classes, while the output of the instance branch is sized N , which is the number of predictable instances (Section III-C). Additionally, the input of the instance head is the concatenation of both decoders outputs. Apart from the input and output sizes of the heads, the two branches differ as each is optimized with its own loss function, which is designed to tackle its task.

We train the semantic branch to predict both *stuff* and *thing* classes. We use the same objective functions as in [16], namely weighted cross-entropy, and Lovász-softmax loss [36], optimizing the Jaccard index. We weight their sum 0.7 for PanosterK and 1.0 for PanosterS. Moreover, following [6], to improve the accuracy on small or distant *thing* objects, we triple the cross-entropy weight on instances smaller than 100 points for PanosterK. We found this to be not beneficial for PanosterS.

C. Learning-based Instance Clustering

Our instance branch predicts an instance ID for each point, including those belonging to *stuff* regions, for which such ID has no meaning and will be later filtered out (Section III-D). This instance ID is the output of the network after an *argmax* operation over the instance branch *logits*. We address instance segmentation as a class-agnostic segmentation task, with the underlying classification based on instance IDs, instead of the usual semantic classes. However, these IDs are interchangeable, similarly to clusters labels, since the essence of instance

	sample prediction	instances purity	fragmented instances
1		3 ok	3 ok
2		2 wrong	2 wrong
3		1 wrong	3 ok
4		12 ok	3 wrong

Fig. 4. Sample instance segmentation predictions. In each row, colors represent predicted clusters for 3 cars. Apart for 1, 2-4 show sub-optimal results: in 2 objects are mixed and split in parts; in 3 the number of cars was underestimated, with impurity on the blue cluster, which merges two instances; in 4 the number of cars was overestimated, with severe fragmentation, as each object is split in four parts.

segmentation is grouping inputs to resemble object instances, e.g. instance clustering. In this Section we describe how to predict such IDs.

Towards this end, we adapt a method originally proposed for separating pulses of radar signals by source, for aircraft applications [12]. In our LiDAR panoptic task, cluster elements are 3D points, and each ground truth cluster is an object instance, e.g. a car. The idea is achieving instance segmentation by guiding the training optimization to minimize two complementary loss functions, which are extracted from a confusion matrix. However, since confusion matrices are inherently non-differentiable, being based on *argmax* operations over the class *logits*, we construct suitable matrices by considering the probability distribution over the *logits softmax* instead. In particular, as depicted in Fig. 3, we build this soft confusion matrix S by collecting the instance branch *softmax* probabilities over the N predictable instances, for all 3D points belonging to ground truth objects, and summing these probabilities together as follows.

For each LiDAR sweep, the *softmax*-based confusion matrix S is sized $G \times N$, with G being the number of ground truth objects. As shown in Fig. 3, the matrix S gathers the instance predictions made by the network over the input 3D points. The notation in this Section refers to sets of cells, rows and columns of the matrix S , rather than single 3D input points. The row S_i corresponds to the i -th object, while the column S^j represents the j -th predicted cluster. The cell S_i^j value is the sum over the probability of each point of the i -th object to belong to the j -th predicted instance, divided by the sum of the row S_i (i.e. the amount of 3D points constituting object i). Unlike in [12], where the matrix values are unbounded and depend on the clusters dimensions, we introduce this division transforming the matrix in percentage-based, to render Panoster robust against the imbalance of instance sizes. The construction of S is shown in Fig. 3.

This matrix S is fully differentiable and can drive the network optimization. We deploy two loss functions computed from the matrix S , namely impurity and fragmentation. As shown in Fig. 4, by aiming at both pure and non-fragmented instances, we can achieve the best outcome.

The **impurity loss** L_{imp} in Eq. 1 tries to shift from 2 and 3 to 1 and 4 in Fig. 4, by optimizing instances purity:

$$L_{imp} = \frac{\sum Q(S)}{\sum S} \quad (1)$$

$$Q(S) := \{S_i^j\} : S_i^j \notin M^j(S) \quad \forall i, j$$

where $M(S) := \{\max(S^j) \forall j\}$ is the set of column maxima of S , and $Q(S)$ is the set of all S cells that are not part of $M(S)$. Column maxima $M(S)$ define each predicted instance by linking it with a ground truth object. By minimizing $Q(S)$, this loss increases the correspondence between ideal and predicted instances.

The additional **fragmentation loss** L_{fr} reduces fragmentation. It limits overestimating the amount of objects, by penalizing the S cells responsible for fragments $F(S)$, which occur when two or more column maxima $M(S)$ fall within the same row:

$$L_{fr} = \frac{\sum F(S) \odot F(S)}{N} \quad (2)$$

$$F(S) := \{S_i^j\} : S_i^j \in M_i(S) \wedge S_i^j \neq \max(M_i(S)) \quad \forall i, j$$

$$M_i(S) := S_i \cap M(S)$$

In the equation, the numerator is the differentiable equivalent of counting the fragments $F(S)$, thanks to the Hadamard element-wise division $\odot$. Therefore, this loss enforces that each ground truth object corresponds to only one predicted instance, bringing use case 4 of Fig. 4 to 1.

Our instance branch is trained with a combination of L_{imp} and L_{fr} , weighted 0.2 and 0.05 respectively.

D. Panoptic Segmentation

The task of PS requires a semantic class for each point, as well as instance IDs for *thing* objects. Both branches output a prediction for each point of *stuff* and *thing* classes. However, instance IDs are meaningless for *stuff* points, and will be filtered out as follows. As shown in Fig. 1, we extract an object mask from the semantic prediction, distinguishing *things* from *stuff* points. We use this mask to discard instance predictions for *stuff*, and retain instance IDs for *things*. Finally, PS predictions are obtained by stacking semantic classes and instance IDs.

Optional post-processing. As described in Section III-C, unlike previous works [6, 10], our instance predictions do not necessarily require further grouping or post-processing. Nevertheless, we explored three different heuristic strategies, as plausibility checks, exploiting 3D information and the synergy between the two tasks. To improve detection and purity, DBSCAN [29] targets the third use case of Fig. 4. We call this strategy *post_splitter*: it splits instances which are too large in space w.r.t. their corresponding predicted semantic classes. Furthermore, to reduce fragmentation, *post_merger* can be deployed to fix the fourth use case of Fig. 4. It uses DBSCAN to cluster the 3D centers of those instances of the same semantic class, which are too close to each other. Finally, with *post_cyclists*, we turn *bicyclists* who have no *bicycle* nearby, but a *motorbike*, into *motorcyclists*.

IV. EXPERIMENTS

A. Experimental Setup

Dataset. We evaluate on the challenging Semantic-KITTI [37], which was recently extended [11] by providing

TABLE I
PANOPTIC SEGMENTATION COMPARISON ON TEST SET RESULTS OF SEMANTICKITTI. ALL VALUES IN [%] WITH St FOR *stuff* AND Th FOR *thing*.

Method	PQ	PQ [†]	SQ	RQ	PQ Th	SQ Th	RQ Th	PQ St	SQ St	RQ St	mIoU
RangeNet++ [15] + PointPillars [20]	37.1	45.9	75.9	47.0	20.2	75.2	25.2	49.3	76.5	62.8	52.4
LPSAD [10]	38.0	47.0	76.5	48.2	25.6	76.8	31.8	47.1	76.2	60.1	50.9
KPConv [21] + PointPillars [20]	44.5	52.5	80.0	54.4	32.7	81.5	38.7	53.1	79.0	65.9	58.8
PanosterK [Ours]	45.6	52.8	78.1	57.0	32.4	77.1	41.6	55.1	78.8	68.2	60.4
PanosterK + <i>post_*</i> [Ours]	52.7	59.9	80.7	64.1	49.4	83.3	58.5	55.1	78.8	68.2	59.9

TABLE II
DETAILED CLASS-WISE PQ RESULTS OF SEMANTICKITTI TEST SET. ALL VALUES IN [%].

Method	PQ	car	truck	bicycle	motorcycle	other vehicle	person	bicyclist	motorcyclist	road	sidewalk	parking	other ground	building	vegetation	trunk	terrain	fence	pole	traffic sign
RangeNet++ + PointP.	37.1	66.9	6.7	3.1	16.2	8.8	14.6	31.8	13.5	90.6	63.2	41.3	6.7	79.2	71.2	34.6	37.4	38.2	32.8	47.4
KPConv + PointP.	44.5	72.5	17.2	9.2	30.8	19.6	29.9	59.4	22.8	84.6	60.1	34.1	8.8	80.7	77.6	53.9	42.2	49.0	46.2	46.8
PanosterK	45.6	49.6	13.7	9.2	33.8	19.5	43.0	50.2	40.2	90.2	62.5	34.5	6.1	82.0	77.7	55.7	41.2	48.0	48.9	59.8
PanosterK + <i>post_*</i>	52.7	84.0	18.5	36.4	44.7	30.1	61.1	69.2	51.1	90.2	62.5	34.5	6.1	82.0	77.7	55.7	41.2	48.0	48.9	59.8

instance IDs for all LiDAR scans of KITTI odometry [38]. It contains 23201 full 360 degree scans for training and 20351 for testing, sampled from 22 sequences from various German cities. We use training sequence 08 as validation split, following the convention. Annotations comprise point-wise labels within 50 meters radius across 22 classes, 19 of which evaluated on the test server: 11 *stuff* and 8 *thing*.

Model training. Experiments are focused mainly on PanosterK, based on KPConv [21], while showing the flexibility of Panoster by applying it also on SalsaNext [16] with PanosterS. Unless otherwise noted, we used the same training parameters as in [21] for PanosterK and [16] for PanosterS, with the addition of our instance clustering loss functions. To better fit the PS task, we used input point clouds covering the entire scene, instead of small regions as in [21]. At each training iteration, we fed PanosterK with at most 80K points. PanosterK used rigid KPConv kernels. For PanosterS we used batch size 4 and learning rate 0.001. We implemented our method in PyTorch and trained the models on a single NVIDIA Tesla V100 32GB GPU. We trained PanosterK from scratch for at least 200K iterations. PanosterS was pretrained on semantic segmentation, and fine-tuned on panoptic segmentation for 100K iterations. The PanosterK model submitted to the test server was trained on the entire training set, including the validation set.

Inference. Unlike KPConv [21], at inference time we fed the whole 50 meters radius scene in a single forward pass, without any test-time augmentation.

Evaluation metrics. We report mean IoU (mIoU) and panoptic quality (PQ) [1] to evaluate semantic and panoptic segmentation, both averaged over all classes. PQ can be seen as the multiplication of segmentation quality (SQ) and recognition quality (RQ) [1], with the former being the mIoU of matched segments and the latter representing the F_1 score, commonly adopted for object detection. Additionally, we report the alternative PQ[†] [5], which ignores RQ for *stuff*

classes. Since our main focus and contribution within PS is on instance segmentation, among the various metrics we concentrate on RQTh, i.e. the F_1 score for *thing* classes, more appropriate than PQ to compare instance segmentation results.

B. Panoptic Segmentation

We compared Panoster with other published PS approaches for LiDAR point clouds, and submitted PanosterK to the SemanticKITTI challenge. Two strong baselines were proposed in [11] combining KPConv [21] or RangeNet++ [15] with PointPillars [20], semantic segmentation and object detection methods. Furthermore, we compared with PanopticTrackNet [9] and LPSAD [10], top-down and bottom-up approaches respectively.

Table I shows results on the test set. Overall, PanosterK delivered superior performance compared to previously published approaches. Despite obtaining worse semantics with a lower SQ, and significantly lower SQTh which penalized PQTh, PanosterK’s strengths are its finer instance segmentation abilities, shown on *thing* classes, resulting in a notable 2.9 increase in RQTh on top of state-of-the-art KPConv and PointPillars, albeit sharing the same backbone. Considering individual classes, Table II shows that PointPillars [20], which is a strong object detector, performed better than our instance clustering approach on the classes *car*, *truck* and *bicyclist*, while ours prevailed on *person*, *motorcycle* and *motorcyclist*. Nevertheless, compared to the combined approaches, our PanosterK was able to deliver a better panoptic segmentation, while using a single network in a multi-task fashion. Interestingly, as shown in Table II, although PanosterK performed comparably to previous approaches on most amorphous *stuff* classes in terms of PQ, it outperformed them on all those *stuff* classes which could be split into instances, such as *traffic sign*, *pole*, and *trunk*. This could be due to our instance losses, which enforced learning the concept of object, despite not

TABLE III
PANOPTIC SEGMENTATION ON THE VALIDATION SET OF SEMANTICKITTI.
ALL VALUES IN [%].

Method	PQ	SQ	RQ	PQ Th	RQ Th	mIoU
RangeNet++ + Pt.P.	36.5	73.0	44.9	19.6	24.9	52.8
LPSAD	36.5	-	-	-	28.2	50.7
PanopticTrackNet	40.0	73.0	48.3	29.9	33.6	53.8
KPConv + Pt.P.	41.1	74.3	50.3	28.9	33.1	56.6
PanosterS	43.5	68.6	55.2	32.6	42.3	57.8
PanosterS + <i>post_*</i>	51.1	76.9	62.1	50.5	58.6	57.8
PanosterK	48.4	73.0	60.1	39.5	50.0	60.4
PanosterK + <i>post_*</i>	55.6	79.9	66.8	56.6	65.8	61.1

TABLE IV
ABLATION STUDY ON THE VALIDATION SET OF SEMANTICKITTI, WITH *c* FOR *car*, *b* FOR *bicycle* AND *p* FOR *person*. ALL VALUES IN [%].

Method	PQ	RQ Th	RQ ^c	RQ ^b	RQ ^p	mIoU
KPConv + instance	45.9	45.2	53.3	42.4	51.2	58.3
+ extra decoder	45.2	43.5	65.4	37.5	60.8	59.0
+ % conf. matrix	47.6	49.3	75.5	47.1	67.1	60.0
+ Lovász-softmax	48.3	49.6	74.2	38.9	70.1	61.1
+ small inst. 3x	48.5	49.4	75.0	40.0	75.1	60.2
+ skip= PanosterK	48.4	50.0	79.5	50.6	74.1	60.4
+ <i>post_splitter</i>	50.4	54.4	87.4	60.1	85.4	60.4
+ <i>post_merger</i>	54.7	62.9	95.3	60.9	87.6	60.4
+ <i>post_cyclists</i>	55.6	65.8	95.3	60.9	87.6	61.1

being applied on *stuff* regions. Furthermore, LPSAD [10], the only other bottom-up method to date, despite a more complex architecture and a necessary grouping step, achieved significantly lower PQ and RQTh. Overall, we attribute PanosterK performance to the high quality instance segmentation resulting from its built-in learning-based separation approach. By optimizing the cluster assignments directly for purity and non-fragmentation, PanosterK could achieve good instance segmentation, without affecting semantic accuracy. Furthermore, applying our optional post-processing techniques *post_** (Section III-D) increased PQ by 7.1 reaching 52.7, with RQTh at 58.5, improving all *thing* classes. More details on this are presented in Section IV-C.

We report validation set results in Table III, including PanopticTrackNet [9], which has no test entry to date. We add validation results of the combined approaches [11], as outlined in [9]. Although our projection-based PanosterS, outperformed previous methods, our point-based PanosterK delivered even better results. We attribute this to the direct consumption of 3D points, which leads to a better use of geometric information, fundamental when separating instances in 3D space.

C. Ablation Study

In Table IV, we summarize an ablation study to assess the impact of different components of PanosterK on PS.

Instance improvements. Although using a single decoder achieved higher PQ and RQTh, dedicated decoders led to a significantly better RQ on *car* and *person*, while improving semantic performance on mIoU, motivating our choice for the latter configuration. As described in Section III-C, compared

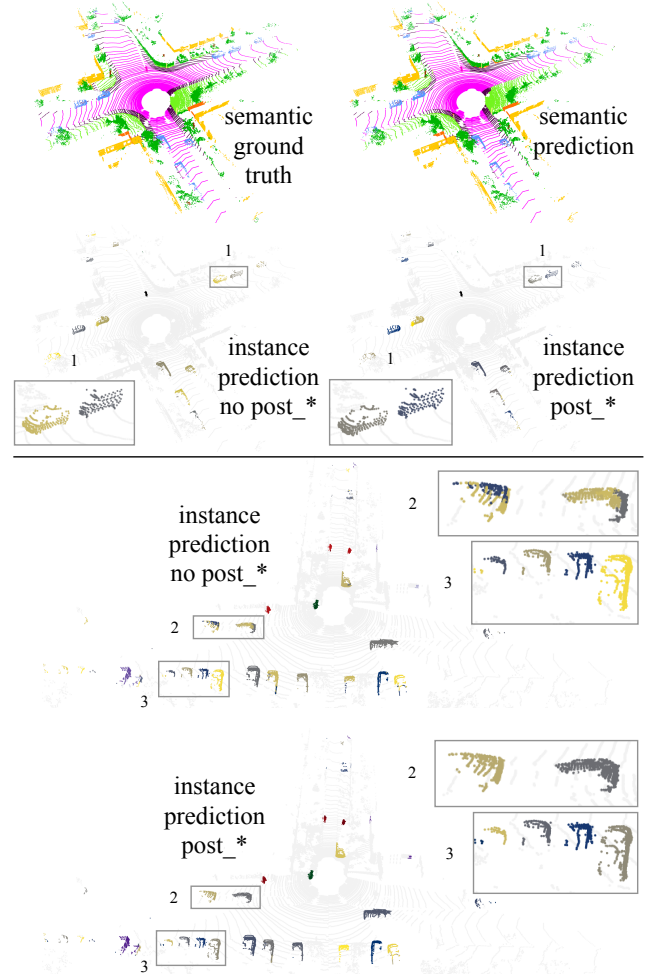


Fig. 5. Qualitative results of PanosterK on two challenging samples from the validation set of SemanticKITTI. Enlarged numbered boxes show representative regions.

to the instance clustering approach adapted from [12], we increased the robustness against unbalanced cluster sizes. This change, shown in Table IV as “% conf. matrix”, largely improved the predictions, preventing closer and larger objects, in terms of point counts, from dominating smaller and further ones in the clustering process. Furthermore, we added a skip connection from semantic to instance head, which led to a higher RQTh, with better detection for the classes *car* and *bicycle*.

Semantic improvements. Since Panoster is a bottom-up method, to avoid errors propagating through the *things* mask, precise semantic segmentation is key for high quality panoptic outputs. Towards this end, we applied the Lovász-softmax loss and we limited false negatives by increasing the weight of small instances (denoted by “small inst. 3x” in Table IV). The former had a positive impact on mIoU, PQ and RQTh. The latter, despite decreasing mIoU, increased precision on smaller *thing* classes, such as *person*.

Number of predictable clusters N. For all experiments we set N equal to 60, higher than the amount of objects found across the dataset in a single scan. This number can be

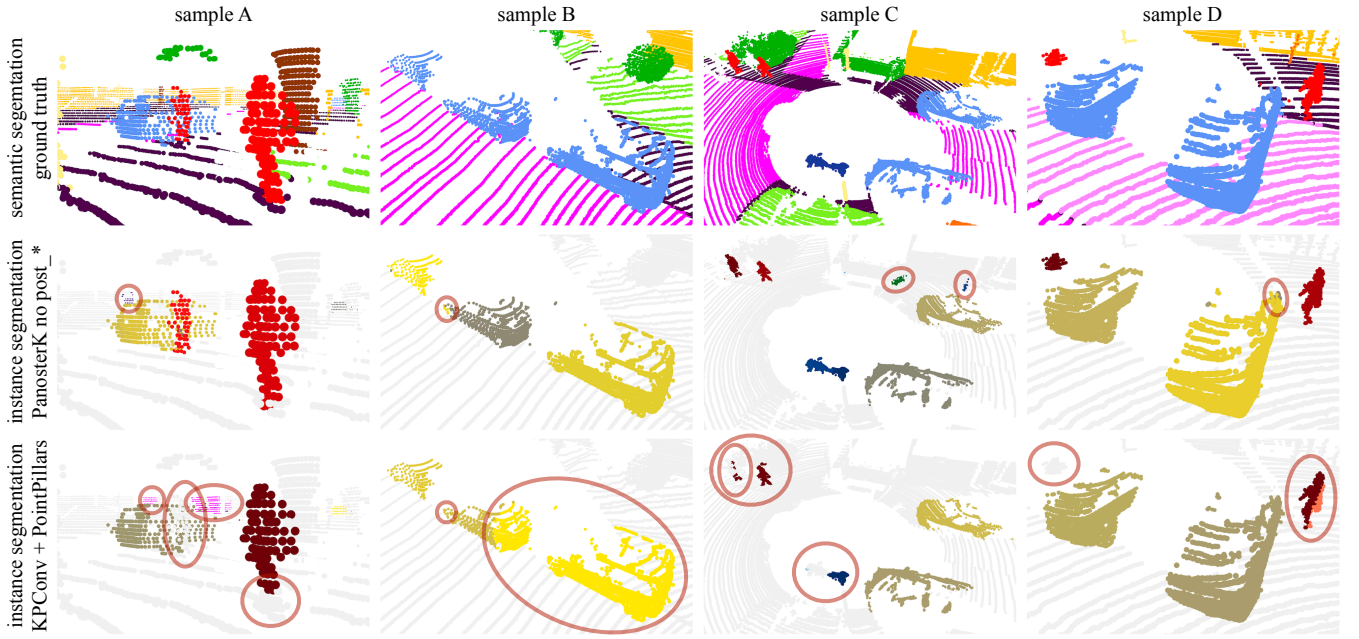


Fig. 6. Qualitative comparison on four challenging portions of the validation set of SemanticKITTI, with instance segmentation predicted by our PanosterK without post-processing, and KPConv + PointPillars. In the second and third rows, each color shade represents an ID, while red circles indicate errors.

seen equivalent to the amount of bounding boxes processed in an object detector [25]. Two architectures using $N = \{60, 90\}$ resulted in a PQ difference of 0.1, showing Panoster insensitivity against N .

Post-processing. As it can be seen in Tables I, II, III and IV, our optional *post_** strategies (Section III-D) bring significant improvements to Panoster predictions. They complement our model, fixing its clustering errors through plausibility checks by exploiting 3D data and the task duality of PS. The largest improvement is brought by *post_merger*, which joins close instances likely to be part of the same object, divided by the model or by *post_splitter*.

D. Runtime Comparison

Our PanosterK model runs on 80K points at 58 FPS at inference time for a single forward-pass on a NVIDIA GTX 1080 8GB GPU, which is 3 FPS (i.e. 5%) slower than its semantic counterpart. Therefore, Panoster brings panoptic capabilities with a relatively small overhead with respect to semantic segmentation networks. This high speed is possible thanks to the simplifications we adopted at inference time (Section IV-A). In comparison, LPSAD, the previously fastest existing method for PS on LiDAR point clouds, runs at 11.8 FPS [10]. Panoster’s built-in learning-based clustering avoids cumbersome grouping strategies to form instances, allowing it to be fast. Moreover, Panoster is more efficient than the combined methods, which require three networks, i.e., one for semantic segmentation, and two for detecting smaller and larger objects [11].

E. Qualitative Results

In Fig. 5 we showcase PanosterK predictions on two complex validation scenes, with road intersections and several

objects. Each shows the whole 50m radius scan. Despite point sparsity, predictions do not degenerate by increasing the distance from the sensor. Additionally, Panoster is able to distinguish neighboring objects, assigning them different IDs (i.e. colors), as in detail 3 of Fig. 5. Although in challenging scenes, such as the bottom one in the figure, it reuses the same ID for multiple instances, these can be fixed via post-processing as matching IDs are found only in distant objects. We attribute this behavior to the relatively low amount of instances and complex scenes in most training samples.

Furthermore, in Fig. 6 we compare PanosterK, without any post-processing, against the approach proposed in [11] combining KPConv and PointPillars. Qualitative results confirm the findings of our experiments, with our PanosterK delivering superior instance segmentation, with minor errors. Despite relying on a strong object detector, the combined approach completely ignored the challenging pedestrian in sample A next to the car and the partially occluded one in D, wrongly divided a person in D, partially ignored the left person and the rare motorcycle in C, assigned the same ID to multiple people in C, and wrongly detected parked cars in B. As both methods are based on the same semantic architecture, i.e. KPConv, we attribute this difference to the benefit of the integrated panoptic solution offered by our Panoster, as well as to the issues rising from dealing with multiple and possibly contrasting predictions when combining the outputs of various networks.

V. CONCLUSION

In this paper we proposed Panoster: a fast, flexible, and proposal-free panoptic segmentation method, which we evaluated on the challenging LiDAR-based SemanticKITTI dataset,

across two different input representations. Panoster outperformed published state-of-the-art approaches, while being simpler and faster. Our method directly delivers instance IDs with a learning-based clustering solution, embedded in the model and optimized for pure and non-fragmented clusters. With its small overhead, Panoster constitutes a valid and efficient approach to extend existing and upcoming semantic methods to perform panoptic segmentation.

REFERENCES

- [1] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár, “Panoptic segmentation,” in *CVPR*, 2019.
- [2] Jonathan Long, Evan Shelhamer, and Trevor Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.
- [3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *NeurIPS*, 2015.
- [4] Rohit Mohan and Abhinav Valada, “EfficientPS: Efficient panoptic segmentation,” in *arXiv*, 2020.
- [5] Lorenzo Porzi, Samuel Rota Buló, Aleksander Colovic, and Peter Kotschieder, “Seamless scene segmentation,” in *CVPR*, 2019.
- [6] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen, “Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation,” in *CVPR*, 2020.
- [7] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia, “Associatively segmenting instances and semantics in point clouds,” in *CVPR*, 2019.
- [8] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung, “JSIS3D: joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields,” in *CVPR*, 2019.
- [9] Juana Valeria Hurtado, Rohit Mohan, and Abhinav Valada, “MOPT: Multi-object panoptic tracking,” in *CVPR Workshop*, 2020.
- [10] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss, “LiDAR panoptic segmentation for autonomous driving,” in *IROS*, 2020.
- [11] Jens Behley, Andres Milioto, and Cyrill Stachniss, “A benchmark for LiDAR-based panoptic segmentation based on KITTI,” in *arXiv*, 2020.
- [12] Stefano Gasperini, Magdalini Paschali, Carsten Hoppe, David Wittmann, and Nassir Navab, “Signal clustering with class-independent segmentation,” in *ICASSP*, 2020.
- [13] Daniel Maturana and Sebastian Scherer, “VoxNet: A 3D convolutional neural network for real-time object recognition,” in *IROS*, 2015.
- [14] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger, “OctNet: Learning deep 3D representations at high resolutions,” in *CVPR*, 2017.
- [15] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss, “RangeNet++: Fast and accurate LiDAR semantic segmentation,” in *IROS*, 2019.
- [16] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy, “SalsaNext: Fast semantic segmentation of LiDAR point clouds for autonomous driving,” in *arXiv*, 2020.
- [17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *CVPR*, 2017.
- [18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *NeurIPS*, 2017.
- [19] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham, “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” in *CVPR*, 2020.
- [20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in *CVPR*, 2019.
- [21] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotequi, François Goulette, and Leonidas J Guibas, “KPConv: Flexible and deformable convolution for point clouds,” in *ICCV*, 2019.
- [22] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao, “SpiderCNN: Deep learning on point sets with parameterized convolutional filters,” in *ECCV*, 2018.
- [23] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij, “KPRNet: Improving projection-based LiDAR semantic segmentation,” in *ECCV Workshop*, 2020.
- [24] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han, “Searching efficient 3D architectures with sparse point-voxel convolution,” in *ECCV*, 2020.
- [25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask R-CNN,” in *ICCV*, 2017.
- [26] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun, “UPSNet: A unified panoptic segmentation network,” in *CVPR*, 2019.
- [27] Bert De Brabandere, Davy Neven, and Luc Van Gool, “Semantic instance segmentation with a discriminative loss function,” in *arXiv*, 2017.
- [28] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool, “Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth,” in *CVPR*, 2019.
- [29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *KDD*, 1996.
- [30] Ji Hou, Angela Dai, and Matthias Nießner, “3D-SIS: 3D semantic instance segmentation of RGB-D scans,” in *CVPR*, 2019.
- [31] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas, “GSPN: Generative shape proposal network for 3D instance segmentation in point cloud,” in *CVPR*, 2019.
- [32] Feihu Zhang, Chenye Guan, Jin Fang, Song Bai, Ruigang Yang, Philip HS Torr, and Victor Prisacariu, “Instance segmentation of LiDAR point clouds,” in *ICRA*, 2020.
- [33] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin, “AdaptIS: Adaptive instance selection network,” in *ICCV*, 2019.
- [34] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen, “DeeperLab: Single-shot image parser,” in *arXiv*, 2019.
- [35] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang, “SSAP: Single-shot instance segmentation with affinity pyramid,” in *ICCV*, 2019.
- [36] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko, “The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks,” in *CVPR*, 2018.
- [37] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in *ICCV*, 2019.
- [38] Andreas Geiger, Philip Lenz, and Raquel Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *CVPR*, 2012.

5.2 Generalizing to Out-of-Distribution Data

Section 5.1 looked into reliability through consistent outputs in setups with standard conditions and generalization settings and fully specified classes. Therefore, training and test data used belong to the same distribution and were captured in ideal settings (i.e., daytime with good visibility). In contrast, in this section, we look into reliability from the perspective of generalization to corner cases, challenging conditions, and out-of-distribution data. So, we consider test samples captured under different illumination and weather settings, countries, and sensors. We also look into strategies to handle completely unseen, unknown objects.

As discussed in Section 4.3, the ability of a model to generalize to out-of-distribution data is crucial for its safe applicability in real-world scenarios. This is because the real world is highly diverse, dynamic, and evolving, constituting a major challenge for learning-based approaches that are typically trained on frozen training data.

In this section, we propose three ways to tackle different domain gaps. Since we aim to avoid using information about the target domain or target objects to be recognized, all methods focus on domain generalization rather than domain adaptation (Section 4.8). In Sections 5.2.1 and 5.2.2, we explore techniques to improve the generalization to corner cases and different countries, sensors, weather, and illumination conditions. In Section 5.2.3, we bring generalization a step further with a new open-set setting to identify and separate unseen, unknown objects without prior knowledge about them.

5.2.1 Domain Generalization with Uncertainty Estimation (RA-L 2021)

In this work, we take a step towards safer perception systems by proposing uncertainty estimation to assess domain gaps and reduce them by improving generalization to out-of-distribution data [82]. We do so with CertainNet, a real-time method providing uncertainty estimates of all output signals of an object detector (i.e., objectness, location, size, and class).

Essentially, we build on top of recent advances in 2D object detection and uncertainty estimation, extending the anchor-less detector CenterNet [367] with the ability to estimate the uncertainty alongside its predictions. Depending on the output signal, we use different techniques to estimate the uncertainty, some of which act as post-processing and do not need modifications at training time. The only one altering the training process is the method to compute the uncertainty at a given location of the objectness, i.e., the object's existence in the scene. To estimate this, we extend DUQ [310] from image classification to a dense regression task and improve its training stability and convergence.

By building on top of DUQ, ours is a single deterministic method, hence delivering bounding boxes and uncertainty estimates from the same model. DUQ works by measuring the distances in the feature space of an input image from averages of the training data [310]. Similarly, our CertainNet compares the pixel-level features with those of prototypical objects representing entire classes (e.g., *car*).

Compared to the straightforward adaptation of DUQ to the object detection task, we propose a set of techniques aiming to build a more structured feature space and better class representatives (i.e.,

centroids). These include a more balanced and controlled update of the class centroids, their protection against outliers, and the regularization of the feature space to better align the centroids with the training samples. We developed these modifications after noticing the post-training mismatch between the centroids and the samples they should be surrounded by.

Instead, for the other output signals of the object detector (i.e., size, location, and class), we exploit the high amount of predictions discarded by the non-maximum suppression stage. Unlike other object detectors that rely on proposals, the selected CenterNet [367] gives as intermediate output a bounding box for every pixel in input. This means that extra boxes are systematically available, enabling the computation of our uncertainties. While we compute them differently from one another, the basic principle is the same: estimating the uncertainty through a measure of the variation between neighboring bounding boxes and discarded outputs.

Unlike other works estimating the uncertainty for object detection [107, 203], ours is the first to quantify the uncertainty for each individual aspect of the detection output (i.e., objectness, location, size, and class) and use it to improve the generalization to out-of-distribution data. Remarkably, the proposed CertainNet achieves around 20% better AP on out-of-distribution data while remaining real-time thanks to its sampling-free nature.

In this work, we also propose an evaluation setup with a mix of in-domain and out-of-domain data, as well as a new metric termed uncertainty boundary quality (UBQ) to evaluate the qualities of the uncertainties on the object location and size. The out-of-distribution data used in the evaluation sets our method as a domain generalization approach, as the data was recorded in different countries, showcasing different vehicles, traffic scenarios, mounting positions, and especially illumination and weather conditions.

The following includes the letter, while the manuscript’s license can be found in Appendix C.

CertainNet: Sampling-Free Uncertainty Estimation for Object Detection

Stefano Gasperini^{*,1,2} Jan Haug^{*,1,2} Mohammad-Ali Nikouei Mahani²
Alvaro Marcos-Ramiro² Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google
* equal contribution

IEEE Robotics and Automation Letters (RA-L), November 2021

Summary. Uncertainty is key reliability information that better aligns with the model’s knowledge and performance capabilities than the standard confidence metric. In this work, we propose a novel sampling-free uncertainty estimation method for object detection and show its benefits in the context of domain generalization. We call it CertainNet, and it is the first to provide separate uncertainties for each output signal: objectness, class, location, and size. To achieve this, we propose an uncertainty-aware heatmap and exploit the neighboring bounding boxes provided by the detector at inference time. Additionally, we propose a new metric to evaluate location and size uncertainties. When transferring to unseen datasets and conditions (e.g., with rain), CertainNet generalizes substantially better than previous methods.

Contributions. Under my supervision, Jan carried out most of the implementation of CertainNet and the baselines as part of his Master’s thesis, following up on a preliminary implementation started by Mohammad. I proposed to evaluate across different domains to investigate the uncertainty under distribution shifts and to estimate the uncertainty for all output signals, turning the project into the proposed work. I designed the experiments to validate our claims, and Jan executed them. Jan and I came up with the techniques to improve DUQ, estimate the various uncertainties, and the proposed metric. I drove the analysis and discussions, wrote the manuscript, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: <https://doi.org/10.1109/LRA.2021.3130976>.

Copyright Statement. ©2021 IEEE. Reprinted, with permission, from Stefano Gasperini, Jan Haug, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, Benjamin Busam, and Federico Tombari, ‘CertainNet: Sampling-Free Uncertainty Estimation for Object Detection’, IEEE Robotics and Automation Letters (RA-L), November 2021.

The license can be found in Appendix C.

CertainNet: Sampling-Free Uncertainty Estimation for Object Detection

Stefano Gasperini* Jan Haug* Mohammad-Ali Nikouei Mahani Alvaro Marcos-Ramiro
Nassir Navab Benjamin Busam Federico Tombari

Abstract—Estimating the uncertainty of a neural network plays a fundamental role in safety-critical settings. In perception for autonomous driving, measuring the uncertainty means providing additional calibrated information to downstream tasks, such as path planning, that can use it towards safe navigation. In this work, we propose a novel sampling-free uncertainty estimation method for object detection. We call it CertainNet, and it is the first to provide separate uncertainties for each output signal: objectness, class, location and size. To achieve this, we propose an uncertainty-aware heatmap, and exploit the neighboring bounding boxes provided by the detector at inference time. We evaluate the detection performance and the quality of the different uncertainty estimates separately, also with challenging out-of-domain samples: BDD100K and nuImages with models trained on KITTI. Additionally, we propose a new metric to evaluate location and size uncertainties. When transferring to unseen datasets, CertainNet generalizes substantially better than previous methods and an ensemble, while being real-time and providing high quality and comprehensive uncertainty estimates.

Index Terms—Computer Vision for Transportation, Deep Learning for Visual Perception, Uncertainty Estimation, Object Detection.

I. INTRODUCTION

WHILE neural networks have been widely employed in research and mobile applications, their usage is still limited in safety-critical real-world settings, such as



Fig. 1. Example predictions of the proposed CertainNet from the KITTI validation set [6]. Uncertainties for objectness, location and size are shown as the complement of the value at the top left of each box, the plus around the center, and dashed boxes respectively. The object class is color-coded.

autonomous driving [1]. The ability of a model to estimate the uncertainty of its predictions is a key enabler for its safe and reliable use in these contexts and unknown scenarios [2]. This makes uncertainty estimation a fundamental companion to object detection [3], semantic segmentation [4], visual odometry [5], and other relevant visual tasks, especially to handle out-of-domain data.

As one of the main perception tasks, object detection comprises a plurality of outputs (e.g., object location, size, and class), requiring both regression and classification, and rendering it particularly challenging for uncertainty estimation. Existing real-time approaches have focused on providing an uncertainty estimate only on a subset of these detection outputs (e.g., location and size [3, 7]). In traffic, there can be situations where an object's location and size are relatively certain, while its existence in the scene (i.e., objectness) is uncertain, e.g., a person in a dark area or entering a car with an unusual pose, as shown at the bottom of Figure 1. In these cases, it is particularly important to assess the objectness uncertainty, to avoid ignoring the detection solely based on a low confidence score. A method confidence is a score which does not correspond to the actual probability of being correct, while uncertainty reduces this gap [1].

However, prior works cover this only partially, as they leave behind the uncertainty over important parameters, such as the objectness, posing a potential safety issue. By extracting a dedicated uncertainty measure for each of the output signals, a system would provide deeper and better insights on the

Manuscript received September 9, 2021; accepted November 7, 2021.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments.

This work was partly sponsored by the German Federal Ministry for Economic Affairs and Energy (grant number: 19A19005B) through the VDA KI-Absicherung project.

* The authors contributed equally.

Stefano Gasperini and Jan Haug are with the Faculty of Computer Science, Technische Universität München (TUM), 85748 Garching bei München, Germany, and also with the BMW Group, 80788 München, Germany (stefano.gasperini@tum.de, jan.haug@tum.de). Corresponding author: Stefano Gasperini.

Mohammad-Ali Nikouei Mahani was with the BMW Group when this work was conducted, 80788 München, Germany (man.mahani@gmail.com).

Alvaro Marcos-Ramiro is with the BMW Group, 80788 München, Germany (alvaro.marcos-ramiro@bmw.de).

Nassir Navab is with the Faculty of Computer Science, Technische Universität München (TUM), 85748 Garching bei München, Germany, and also with the Department of Computer Science, Johns Hopkins University, Baltimore, 21218 MD USA (nassir.navab@tum.de).

Benjamin Busam is with the Faculty of Computer Science, Technische Universität München (TUM), 85748 Garching bei München, Germany (b.busam@tum.de).

Federico Tombari is with the Faculty of Computer Science, Technische Universität München (TUM), 85748 Garching bei München, Germany, and also with Google, 8002 Zürich, Switzerland (tombari@in.tum.de).

Digital Object Identifier (DOI): 10.1109/LRA.2021.3130976.

scene, which could be exploited by downstream tasks (e.g., path planning). Furthermore, incorporating uncertainty can reduce the accuracy of the models [8], or significantly increase the latency [9, 10], hence leading to a challenging trade-off between accuracy, safety, and runtime.

In this work, we address these issues with a novel method that provides separate and dedicated uncertainties for all detection outputs, while preserving speed and accuracy. We name our approach CertainNet, and our contributions can be summarized as follows:

- We introduce a novel sampling-free method to estimate the model uncertainty of an object detector.
- We propose the first real-time approach to estimate the uncertainty of all signals of the detection output (i.e., objectness, location, size and class).
- We present new simple metrics to assess the quality of size and location uncertainties for object detection.

Additionally, we provide extensive evaluations and comparisons on three challenging public datasets, including out-of-domain data, reaching competitive accuracy, while improving significantly on uncertainty-aware metrics.

II. RELATED WORK

A. Uncertainty Estimation for Neural Networks

A learning-based system can express two kinds of uncertainty: the epistemic uncertainty caused by the model, and the aleatoric uncertainty due to the input data [1]. The former can arise from errors in the training procedure (e.g., ignoring data imbalance), a sub-optimal architecture or the knowledge gap with out-of-domain data, while the latter originates from the data itself, e.g., due to sensor noise, coarse approximations, ambiguous samples, or also inaccurate ground truth labels [1].

There are several types of uncertainty estimation approaches. In this Section we provide a brief overview following the categorization of Gawlikowski et al. [1]. **Single deterministic methods** predict with a deterministic model and provide an uncertainty estimate directly through that model [11], or via external computation [12, 13]. There are various ways for external methods to estimate the uncertainty, e.g., by computing the distance to the training data. DUQ, which is the work of Van Amersfoort et al. [12], belongs to this subcategory. The authors used Radial Basis Functions (RBF) to learn a transformation of features into a hyperspace. At inference time, they quantify the uncertainty as the distance in the hyperspace between the predicted features and a set of learned class centroids.

Another group of works is that of **Bayesian methods**, which infer the probability distribution of the model parameters [1]. In this category, Monte-Carlo Dropout (MC Dropout) [9] approximates this distribution by considering the set of models originating from keeping dropout active at test time. Thanks to the randomness of dropout, each forward pass produces different outputs, from which the uncertainty can be estimated according to their disagreement.

Then, **ensemble methods** also combine multiple outputs for both predictions and uncertainty estimates [14]. In this case the outputs are inferred by a set of different deterministic

models, trained independently. Subcategories of ensembles can be found depending on how these different models are obtained [1].

Lastly, **test-time augmentation methods** use a single deterministic model and augment its input during inference, collecting a variety of predictions, from which a final output and uncertainty estimate are computed [15].

A different categorization distinguishes sampling-based and sampling-free approaches, depending on how the uncertainty is estimated. The former group requires inferring multiple times on the same input with different models (for Bayesian and ensemble methods), or different inputs with the same model (with test-time augmentations), and then aggregating the results. This significantly increases the runtime [9, 10], which makes it impractical for most real-time scenarios, such as autonomous driving. Instead, the latter category does not require sampling, thanks to built-in quantification techniques, such as those typically used in single deterministic methods [12].

Our work builds on top of the findings of Van Amersfoort et al. [12]. In particular, we considerably extend and adapt their DUQ from image classification to a dense regression task for object detection, estimating the model objectness uncertainty at a given location, while significantly improving training stability and convergence. Additionally, we estimate the uncertainty on the variety of regressed outputs of object detection (e.g., object size), by aggregating the predictions provided by the detector.

B. Uncertainty Estimation in Object Detection

While many approaches quantify the uncertainty on basic tasks, posing the foundations of this domain [9, 12], fewer estimated it for safety-critical settings, such as autonomous driving [4, 16]. As we focus on estimating the uncertainty for object detection, in this Section we provide a brief summary of existing works in this area.

Choi et al. proposed Gaussian YOLOv3 (GYOLO) [3], extending the popular YOLOv3 detector [17] to estimate the object location and size uncertainties. They achieved this by predicting Gaussian parameters (i.e., mean and variance) for the position and dimensions of the boxes. Miller et al. [10] used MC Dropout [9] and thoroughly evaluated different clustering techniques, extracting variance and entropy of spatial and class predictions respectively. Harakeh et al. [16] also used MC Dropout [9] and replaced the usual non-maximum suppression (NMS) with clustering and Bayesian inference, obtaining an uncertainty quantification for the object size and class. Instead, Lee et al. [7] introduced with their Gaussian-FCOS a dedicated head on top of an anchor-less detector to estimate the localization uncertainty for each of the four box boundaries independently.

Likewise, our proposed CertainNet estimates the uncertainty for object detection, but in a different way. Unlike the works of Miller et al. [10] and Harakeh et al. [16], ours is sampling-free and real-time capable. Compared to GYOLO [3] and Gaussian-FCOS [7], ours does not regress the uncertainty as an explicit additional output, thereby reducing the potential

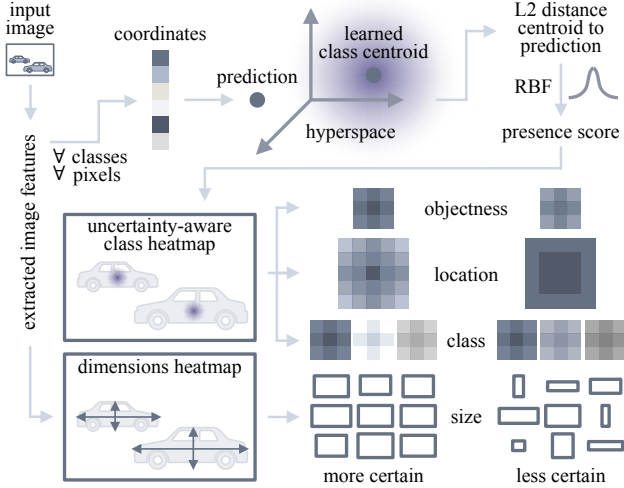


Fig. 2. The proposed CertainNet. The top shows how each uncertainty-aware objectness score is computed. At the bottom right, examples of more and less certain predictions for each output signal are shown.

impact of biases in the training data. Moreover, we are the first to quantify the uncertainty for each individual aspect of the detection output (i.e., objectness, location, size and class), increasing the explainability.

III. METHOD

In this work, we estimate the uncertainty for each aspect of object detection, improving the model explainability for safety-critical applications. Towards this end, we build on top of an anchor-less detection framework (Section III-A), which we render uncertainty-aware (Section III-B). We then compute the uncertainty for objectness, location, dimensions and class (Section III-C). Figure 2 shows an overview of the proposed method. This fragmentation of the uncertainty along the individual detection components introduces valuable information to improve the safety of downstream tasks.

A. Base Detection Framework

CenterNet [18] is a fast and accurate anchor-less detector. It predicts objects centers via a class heatmap, and regresses the bounding box size in a separate dimensions heatmap. CenterNet gives as intermediate output a bounding box for every pixel, hence providing a distribution of outputs to estimate our uncertainties. For these reasons, we use it as base detector. While we focus on 2D bounding boxes, 3D boxes can be predicted following [18], and the additional uncertainties (e.g., for object distance and length) can be computed analogously to the ones presented in this work.

B. Uncertainty-awareness

We make the class heatmap of the detector uncertainty-aware, by exploiting DUQ [12], extending it from recognizing out-of-domain samples in image classification tasks, for which it was proposed, to a dense regression task within our object detector. During training, we learn a set of class representatives (i.e., centroids), which are then compared with each prediction

at inference time. By doing so, similarly to DUQ [12], we assess the deviation from learned object prototypes, which is linked to the model uncertainty. We compute this for all regressed heatmap values, which represent the objectness scores. In particular, it is the comparison with the learned centroids, which makes every value of this heatmap uncertainty-aware.

Proposed for image classification, DUQ [12] learns to map each image to N_{cls} high dimensional spaces, one for each of N_{cls} classes. At training time, class centroids are updated as moving averages of positive samples. As Berger et al. [19] also pointed out, DUQ suffers from instability issues and deployment difficulties. Therefore, after performing a thorough analysis of DUQ and its feature space, we incorporate several modifications to better suit the object detection task, as well as to improve the overall convergence and stability of the model, as visualized in Figure 3.

Adaptation of DUQ to dense regression: Compared to the original implementation [12], due to the high dimensionality of the centroid space (e.g., 512 in [12]), and also the two additional dimensions required for our class heatmap output (i.e., width and height), speed and memory efficiency issues arise. We circumvent these by exploiting the shared operation across the pixels when transforming to the hyperspace. Towards this end, we use 1×1 convolutions, instead of individual multiplications as in [12].

Balanced centroid update: The centroids need to be a good representation of the data, as at inference time they determine both predictions and uncertainties, making the centroid update a critical step. In particular, the centroids are updated as trailing averages, of which the current estimates are computed to get closer to the current sets of predictions. In a regular setup [12], scaling by the amount of samples of a given class c is reasonable, as each minibatch t is representative of the whole dataset. However, with unbalanced classes and a varying amount of objects, that would lead to inconsistent update magnitudes, hindering convergence. We circumvent this and account for the minibatch variability by computing the centroid $\mathbf{e}_{c,t}$ as:

$$\mathbf{e}_{c,t} = \gamma \cdot \mathbf{e}_{c,t-1} + \frac{(1-\gamma)}{\sum_i \mathbf{y}_{c,t,i}^\lambda} \sum_i (\mathbf{y}_{c,t,i}^\lambda \circ \mathbf{W}_c f_\theta(\mathbf{x}_{t,i})) \quad (1)$$

This allows to set a weight on the center pixel higher than its surroundings, depending on λ . The right term is an average weighted and scaled by the ground truth class heatmap values $\mathbf{y}_{c,t}$, having a Gaussian around each object center. $\mathbf{W}_c f_\theta(\mathbf{x}_{t,i})$ are the predicted hyperspace coordinates, with f_θ being the feature extractor, and $\mathbf{W}_c$ the hyperspace transformation for class c . γ is a hyperparameter, defined as the centroid momentum, as in [12].

Hyperspace regularization: To properly represent the training data, the centroids should lie among positive samples distributed along a Gaussian hypersphere. However, by visualizing the hyperspace via PCA and t-SNE, we noticed that this is not the case, with the centroid often falling around the boundary of an irregularly shaped training distribution (top of Figure 3). We circumvent this and aim at the ideal case, by

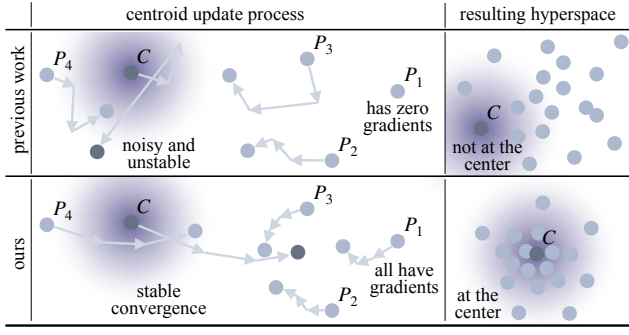


Fig. 3. Centroid update scheme, with the learned hyperspace. The bottom shows the impact of our modifications, compared to the unstable procedure of DUQ [12] at the top when applied on an unbalanced detection task.

regularizing this hyperspace with:

$$\mathcal{L}_{reg} = \frac{1}{\sum_{c,i} \mathbf{y}_{c,i}^\lambda} \sum_{c,i} (\mathbf{y}_{c,i}^\lambda \circ \|\mathbf{W}_c f_\theta(\mathbf{x}_i) - \mathbf{e}_c\|_2^2) \quad (2)$$

which acts on the euclidean distance between the centroid and prediction for each pixel, and is weighted by the ground truth class heatmap values $\mathbf{y}_{c,i}$. Furthermore, we do not use the gradient penalty $\mathcal{L}_{grad}$ of DUQ [12], as we find $\mathcal{L}_{reg}$ to deliver more stable and consistent results.

Centroid momentum scheduling: The centroids are moved using steps with a magnitude proportional to their distance to the predictions, and to the centroid momentum γ . A large γ slows down training, while improving its stability, but reduces the robustness against the initialization. Instead, a low γ has the inverse effect. While a relatively low γ worked well for balanced classes and large batch sizes [12], it did not in our task. We apply a scheduling of γ similar to that of learning rates, to improve the stability in later training stages, and preserve robustness and speed at earlier ones. This effect is represented in Figure 3. Specifically, we reduce γ by a factor of 10 at fixed epochs.

Outliers protection: Additionally, we prevent outliers from impacting the centroids. In particular, we define an outlier as a prediction being further from its centroid than the triple of the length scale σ . This further improves the training stability and convergence, as instead of having the predictions following a highly moving target (i.e., centroid), they go towards a more stable target.

Length scale annealing: At early stages of training, the centroids position and the hyperspace transformation are not yet properly tuned, due to random initialization. This, together with the high dimensionality of the space, causes instability, as even the correct inputs can be mapped far from their centroid. We alleviate this problem and improve stability with an approach similar to that of simulated annealing [20]: we increase the length scale σ at the initial stages, to then slowly reduce it at each step towards the original value. This increases the gradients at larger distances, nearly zero otherwise, thereby allowing to improve convergence and predictions, as depicted in Figure 3.

C. Uncertainties Estimation

Objectness: Thanks to the class heatmap uncertainty-awareness (Section III-B), the objectness uncertainty $\mathcal{U}_{obj}$ can be directly inferred as the complement of the score:

$$\mathcal{U}_{obj} = 1 - \hat{p} \quad \text{with: } \hat{p} = \max_c p(\mathbf{W}_c f_\theta(\mathbf{x}) \mid \mathcal{N}(\mathbf{e}_c, \sigma)) \quad (3)$$

with $\mathcal{N}(\mathbf{e}_c, \sigma)$ being a Gaussian distribution centered at the centroid $\mathbf{e}_c$ with variance σ , which is the length scale.

Location: An object location is computed according to the class heatmap. In our CertainNet, this output is uncertainty-aware (Section III-B). We make use of this by extracting the location uncertainty considering the distribution of scores around an object center. As can be seen in Figure 2, a location is more certain if the center peak is more defined and sharper, while it is less certain with a smoother and flatter peak. The uncertainty $\mathcal{U}_x$ for the location in x direction depends on the horizontal variance and is normalized by the predicted width $\hat{w}$. We compute $\mathcal{U}_x$ as follows:

$$\mathcal{U}_x = \frac{\sqrt{\text{var}_x}}{\hat{w}} \quad \text{with: } \text{var}_x = \frac{\sum_i o_x^2(i) \hat{p}_i \cos^\eta \alpha_i}{\sum_i \hat{p}_i \cos^\eta \alpha_i} \quad (4)$$

where i iterates over the pixels around the center, $o_x(i)$ is the x -offset from pixel i to the center, α_i is the angle between the x -axis and the pixel, and $\hat{p}_i$ is the predicted objectness (Equation 3). η weights the influence of the scores that are not axis-aligned. For the vertical location $\mathcal{U}_y$ is computed analogously, replacing $\hat{w}$ with the predicted height $\hat{h}$ and using var_y , obtained through the y -offset $o_y(i)$ and $\sin \alpha_i$.

Dimensions: As described in Section III-A, the model provides a bounding box for each pixel in the output heatmaps. We exploit this by computing the uncertainty on the object size $\mathcal{U}_{w,h}$, by comparing the boxes around an object center. Intuitively, the more these boxes are similar to one another, the more certain the model is on the object dimensions, and vice versa, as shown in Figure 2. Towards this end, we compute $\mathcal{U}_w$ as an RMSE over the surrounding predicted dimensions for each object center, and weight this by the pixel objectness score:

$$\mathcal{U}_w = \frac{1}{\hat{w}} \sqrt{\frac{1}{\sum_i \hat{p}_i} \sum_i \hat{p}_i \cdot (\hat{w} - \hat{w}_i)^2} \quad (5)$$

where $\hat{w}$ is the predicted object width and $\hat{w}_i$ is the width at the heatmap pixel i around the center. The height uncertainty $\mathcal{U}_h$ is computed analogously using the predicted heights $\hat{h}$.

Class: The object class is predicted via the class heatmap, which is uncertainty-aware in our model (Section III-B). As in Figure 2, a class prediction is more certain, if there is a peak only at such class with the other class probabilities being low, while it is less certain, the higher those other probabilities are. We formalize this, by comparing the objectness scores at each predicted center across the classes:

$$\mathcal{U}_{cls}(i) = \mathcal{U}_{cls}(i-1) + (1 - \mathcal{U}_{cls}(i-1)) \cdot \left(\frac{\hat{p}_{cls}(i)}{\hat{p}_{cls}(1)} \right)^i \quad (6)$$

with $\hat{p}_{cls}(i)$ being the i^{th} -highest class probability, so $i = 1$ for the predicted one. We compute $\mathcal{U}_{cls}$ recursively, considering $\mathcal{U}_{cls}(0) = 0$ and reaching the final class uncertainty $\mathcal{U}_{cls}$ as $\mathcal{U}_{cls}(N_{cls})$.

TABLE I
DETECTION PERFORMANCE COMPARISON WITH RELATED WORKS ON THE KITTI [6] VALIDATION SET ACROSS CLASSES AND DIFFICULTIES.

Method	mAP	Easy	<i>Car</i> Mod.	Hard	<i>Pedestrian</i>			Easy	<i>Cyclist</i> Mod.	Hard	all		fps
GaussianYOLO [3]	56.74	89.91	84.36	75.65	58.43	50.95	43.69	45.97	31.17	30.53	65.39	96.42	18.06
CenterNet [18]	71.49	92.30	89.15	82.17	<u>76.53</u>	<u>67.53</u>	59.37	73.48	52.63	50.25	68.33	<u>99.82</u>	20.21
5-Ensemble [10]	73.70	97.84	89.75	80.87	77.08	67.66	59.17	79.45	57.30	54.17	72.13	97.32	5.10
CertainNet [ours]	<u>73.00</u>	<u>93.81</u>	<u>89.36</u>	<u>82.11</u>	76.33	66.13	58.54	<u>78.02</u>	57.49	55.20	<u>69.85</u>	99.85	16.46

TABLE II

UNCERTAINTY QUALITY COMPARISON WITH RELATED WORKS ON THE VALIDATION SETS OF KITTI [6], BDD100K [21], AND nuIMAGES (nuIm.) [22]. ALL MODELS WERE TRAINED ON KITTI, AND TRANSFERRED (→) TO THE OTHER DATASETS. ALL RESULTS ARE FOR *Car* (MODERATE FOR KITTI). ON CenterNet[‡] WE APPLIED OUR LOCATION AND DIMENSIONS UNCERTAINTY ESTIMATIONS AS POST-PROCESSING.

	Method	AP	AUPR-In	AUPR-Out	objectness AUROC	ECE ↓	UE ↓	CE ↓	location UBQ	BR	CE ↓	dimensions UBQ	BR
KITTI	GaussianYOLO [3]	84.36	<u>74.93</u>	91.24	86.23	23.37	20.02	11.58	60.01	60.56	6.18	75.34	77.47
	CenterNet [‡] [18]	89.15	72.76	<u>99.56</u>	96.65	10.49	6.00	7.17	68.27	69.80	4.23	85.84	89.58
	5-Ensemble [10]	89.75	73.79	93.61	87.13	16.93	17.72	<u>4.59</u>	86.71	90.43	5.01	84.28	87.54
	CertainNet [ours]	<u>89.36</u>	78.16	99.81	98.00	4.60	4.98	4.54	<u>75.92</u>	<u>77.81</u>	<u>4.47</u>	86.76	91.05
→ BDD	GaussianYOLO [3]	31.03	<u>88.73</u>	79.72	85.84	22.80	20.68	9.48	57.32	58.49	7.67	57.94	59.19
	CenterNet [‡] [18]	<u>34.51</u>	81.35	98.06	<u>91.74</u>	<u>7.78</u>	14.03	25.72	35.90	36.76	10.53	<u>73.43</u>	<u>79.55</u>
	5-Ensemble [10]	26.75	96.42	81.84	91.99	25.64	15.26	3.31	84.83	93.54	9.40	84.81	93.50
	CertainNet [ours]	40.93	78.77	<u>97.82</u>	91.17	4.89	<u>14.85</u>	14.79	44.71	46.03	9.68	72.91	78.89
→ nuIm.	GaussianYOLO [3]	30.38	86.24	87.95	87.78	17.05	17.74	<u>10.42</u>	<u>56.20</u>	<u>57.62</u>	8.74	58.38	60.14
	CenterNet [‡] [18]	<u>43.93</u>	75.94	<u>98.93</u>	91.58	5.51	15.70	27.07	35.69	36.21	8.39	74.79	79.21
	5-Ensemble [10]	35.31	93.93	90.90	<u>93.04</u>	17.74	<u>13.98</u>	3.37	89.48	98.51	8.36	89.31	98.21
	CertainNet [ours]	53.14	79.97	99.28	94.24	<u>7.80</u>	11.90	16.21	43.57	44.46	7.73	<u>75.36</u>	<u>80.25</u>

IV. EXPERIMENTS AND RESULTS

A. Experimental setup

Datasets: We conducted our experiments on three public autonomous driving datasets, namely KITTI [6], the recent nuImages extension of nuScenes [22], and the dashcam-based BDD100K [21]. **KITTI** is a popular benchmark for object detection, and it has been recorded in Germany. We followed the standard 3DOP split [18], which comprises 3712 training and 3769 validation images. Moreover, we used the standard *car*, *pedestrian* and *cyclist* classes. **NuImages** is a highly diverse large-scale dataset, collected in Boston and Singapore. It includes rain, snow and night conditions. **BDD100K** is another diverse large-scale dataset. It was crowd-sourced from a variety of dashboard cameras and different vehicles, around cities of the USA. It also includes various challenging weather and lighting conditions, such as snow, fog and night. In particular, we used nuImages and BDD100K to assess the generalization capability to out-of-domain data, which is critical for autonomous driving and uncertainty estimation. Towards this end, we evaluated KITTI models (without any fine-tuning) on the validation set of nuImages, which includes 3249 images, sized 1600x900, and the validation set of BDD100K, which contains 10K samples, sized 1280x720. Due to the sub-optimal class overlap between KITTI and the two transfer datasets, for the evaluation we focused on the *car* class.

Evaluation metrics: We evaluated the object detection performance on the standard AP, with the standard IoU threshold of 0.7 for KITTI *cars* and 0.5 for the other classes and datasets. We split the evaluation according to the specific uncertainty

estimates. For the objectness uncertainty we computed various metrics, such as the area under the precision-recall (AUPR) curve, as AUPR-In and AUPR-Out [10], which should be considered jointly. The former assesses the ability to accept all correct detections and minimise wrong acceptances. Conversely, the latter looks at the rejection of negative samples and the rate of wrong rejections. Additionally, the area under the receiver operating characteristic curve (AUROC) compares true and false positives rates, as used by Miller et al. [10]. Following [23], the expected calibration error (ECE) is the average difference between confidence and detection accuracy, weighted by the occurrence. Moreover, the minimum uncertainty error (UE) checks the ability to accept correct detections and reject incorrect ones based on the uncertainty estimate [10]. In the top of Figure 4, we show how location and size uncertainties have a similar effect on the box uncertainty boundaries, despite leading to different box distributions. This allows to evaluate them in the same way. We adapted the expected calibration error (ECE) [23] for our detection case, as the calibration error (CE): average deviation between predicted uncertainty and measured error for matched detections. We independently computed the CE on *x* and *y* for the location, and on *width* and *height* for the size, reporting the average of each pair. All metrics in this work are percentage-based.

Uncertainty boundary quality: We introduce a new metric to evaluate the location and size uncertainties. As shown at the bottom of Figure 4, the goal is assessing how much of an object is in the uncertainty boundaries. We name this metric uncertainty boundary quality (UBQ), and compute it as the

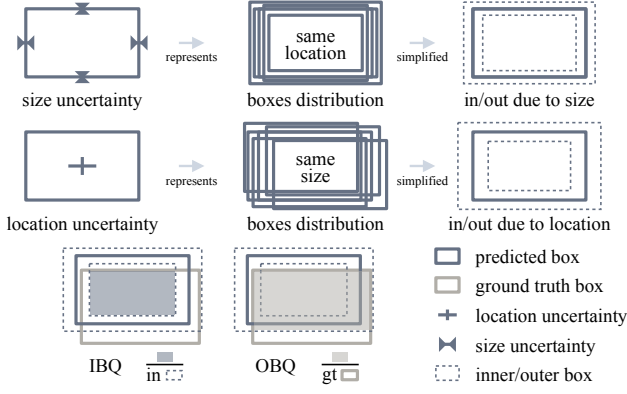


Fig. 4. The top shows the effect of size and location uncertainties on the box uncertainty boundaries, allowing to evaluate them in a similar fashion. The bottom shows the IBQ and OBQ terms of our novel UBQ metric.

average across matched boxes of:

$$UBQ_i = \frac{1}{2} (IBQ_i + OBQ_i) BR_i \quad (7)$$

with BR being the boundary ratio between inner and outer:

$$BR_i = \frac{1}{2} \left(\frac{\text{width}(in_i)}{\text{width}(out_i)} + \frac{\text{height}(in_i)}{\text{height}(out_i)} \right) \quad (8)$$

UBQ is composed of an inner (IBQ) and an outer (OBQ) term, which are averaged and multiplied by the ratio (BR), to penalize extremely broad and trivial confidence intervals. Figure 4 shows how IBQ_i and OBQ_i are computed:

$$IBQ_i = \frac{A(gt_i \cap in_i)}{A(in_i)} \quad OBQ_i = \frac{A(gt_i \cap out_i)}{A(gt_i)} \quad (9)$$

where A is the area of a bounding box i , gt is the ground truth box, in and out are the inner and outer predictions. As for the CE, our UBQ metric can be used for both location and size. We report UBQ as a summary, and BR as the ratio.

Network architecture: All models were based on a single DLA architecture [24] for all classes. Compared to others it offers a good speed-accuracy trade-off [18]. Thus, we followed the DLA-34 configuration of CenterNet, with deformable convolutions [18]. We then made the class heatmap head uncertainty-aware, with the hyperspace transformation and the RBF kernel, as described in Section III-B.

Implementation details: We trained our models with an Adam optimizer for 70 epochs, with 10 additional epochs for the objectness head. We used an initial learning rate of $1.25e-4$, reduced by a factor of 10 after epochs 45 and 60. We chose a batch size of 16 and a resolution of 1280x384 for KITTI, resizing BDD and nuImages samples to 896x512. We selected 512 as centroid dimensionality, and initialized their momentum γ to 0.9, increasing it to (0.99, 0.999, 0.9999) at epochs (5, 20, 60) respectively. The length scale σ was initialized at $2.5e-1$, with a decay rate of 0.999 per step for the sigma annealing, until reaching $5e-2$. We weighted the hyperspace regularization loss $\mathcal{L}_{reg}$ by $1e-2$, and set λ to 20 for balancing the score. To compensate the added loss on the objectness, the weight on the dimensions loss was increased to 0.2. We set $\eta = 4$, and kept the remaining hyperparameters the

TABLE III
ABLATION STUDY ON THE KITTI [6] VALIDATION SET ACCORDING TO
Car MODERATE DETECTION AND UNCERTAINTY QUALITY.

ID	Description	AP	AUPR-In	AUPR-Out
A0	adapted DUQ [12]	76.41	81.64	99.62
A1	A0 - $\mathcal{L}_{grad} + \mathcal{L}_{reg}$	87.33	76.26	99.15
A2	A1 + balanced update	87.98	74.12	99.71
A3	A2 + outlier protection	88.14	77.34	99.56
A4	A3 + momentum schedul.	88.76	75.27	99.72
A5	A4 + length scale anneal.	88.85	81.08	99.80
A6	A5 + freeze last 10 epochs	89.36	78.16	99.81

same as proposed for CenterNet [18]. We trained our models using PyTorch on a single NVIDIA Tesla V100 32GB GPU and evaluated the runtime (fps) on a single NVIDIA Quadro RTX 4000 8GB GPU.

Details for prior works: For a fair comparison, we re-trained all methods on the same standard split [18], using official implementations and hyperparameters, all pretrained on [25], until convergence. For GYOLO [3] we used the best settings with input size 704 on Darknet. For location and size uncertainties, we extended CenterNet [18] with our corresponding uncertainty estimations, on top of its outputs. For the ensemble, we trained 5 DLA-34-based CenterNets (randomly initialized, with data shuffle and augmentations), and combined their predictions following Miller et al. [10], via DBSCAN. For CenterNet, GYOLO and the ensemble, the objectness uncertainty is the confidence score provided.

B. Quantitative Results

Detection: Table I compares the detection performance on KITTI [6]. Our uncertainty-aware CertainNet outperformed the CenterNet [18] baseline on the overall mAP, as well as most classes and difficulties, while sharing the same architecture. Our added safety-related outputs reduced the runtime by 18% to 16.46 fps, thereby remaining real-time. Moreover, the sampling-based 5-Ensemble [10] models achieved the highest mAP, and the best score in most classes and difficulties, with ours as close second. However, the ensemble superiority came at the cost of a significantly higher runtime. GYOLO [3] was slightly faster than ours, but produced the worst detections among tested methods. As shown in Table I, its APs are significantly lower than those reported by Choi et al. [3], due to the different KITTI splits used. In particular, Choi et al. split training and validation randomly across single frames. We used instead a standard split [18], reducing the similarity between training and validation.

Uncertainty estimation: In Table II we report a variety of uncertainty metrics for different datasets, for *Car*. On KITTI, ours achieved higher scores and lower errors across most metrics. Remarkably, our approach was able to achieve the best objectness results across the board, often with a substantial margin (e.g., on the calibration ECE), thanks to its effective uncertainty-aware heatmap. Interestingly, in terms of uncertainties, the ensemble could not achieve satisfactory results, except for the location. This can be due to the disagreement of the 5 models, and to the difficulty of merging

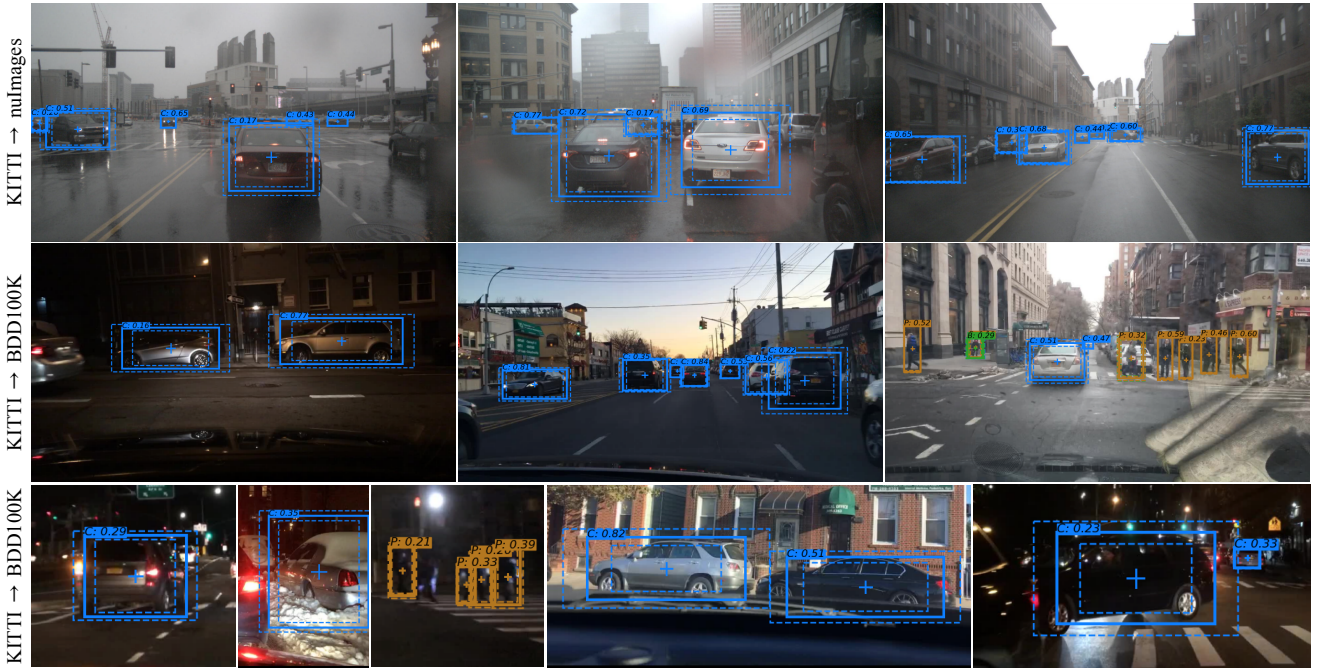


Fig. 5. Example predictions of the proposed CertainNet on out-of-domain data. The model was trained only on KITTI [6] and applied on nuImages [22] and BDD100K [21] without any fine-tuning. The bottom row shows crops. The boxes follow the format introduced in Figure 1, with KITTI $\rightarrow$ KITTI.

contrasting bounding boxes. Table II shows also the suboptimal performance of GYOLO [3], with a tendency to overestimate the uncertainty boundaries, especially for the location (BR around 60%), despite its training time modifications for location and size, and having been trained on the same dataset. Furthermore, the flexibility of our uncertainty quantification techniques allowed to produce estimates also for the location and size of the uncertainty-unaware CenterNet [18]. However, this configuration could not match the uncertainty quality of our CertainNet, especially for the location, computed on the class heatmap. This shows how our training time modifications contributed to the uncertainty-awareness of the whole model.

Out-of-domain data: As all models were trained on KITTI, the results for BDD100K and nuImages in Table II show the ability of each method to generalize to unseen scenarios in out-of-domain data. As also pointed out in [26, 27], multiple differences define a substantial domain gap between these two datasets and KITTI: weather and lighting conditions, image resolution and aspect ratio, camera mounting position and type (e.g., dashcam often with reflections on BDD), different continents and street layouts, amount of dynamic scenes, as well as different vehicles on sale in those regions. All these render the transfer task particularly difficult. Most notably, our CertainNet reached the highest AP on both datasets, by a significant margin on both BDD and nuImages. This shows the robustness of our approach, as well as the strong generalization provided by our uncertainty estimations, compared to CenterNet [18]. Interestingly, the ensemble, which performed well on KITTI (i.e., the training dataset), underperformed on both out-of-domain datasets. We attribute this to the rule-based clustering step required to merge the predictions [10], which was tuned on KITTI. This reiterates that the ensemble’s

aggregation step is critical on object detection, and is prone to deliver suboptimal results on out-of-domain data, which is fundamental in autonomous driving. GYOLO [3] followed a similar trend, but suffered the transfer to out-of-distribution less than the ensemble, thanks to its learning-based techniques that generalized better. Nevertheless, the uncertainty estimations of the ensemble seemed high quality for the most part, especially on the object location and size. However, this is due to the rather low number of detections that all related works provided, as discussed below. Overall, our CertainNet achieved the highest AP, while maintaining high quality uncertainty estimates, which correctly increased on out-of-domain data.

Amount of detections: The uncertainty evaluation metrics used in this work are highly influenced by the number of predictions or that of matched detections. Therefore, high scores and low error rates can be achieved by a model that outputs a low amount of high confidence boxes. We found this to be the case for the methods in Table II: in nuImages [22] our method detected 4905 *cars*, CenterNet 3714, 5-Ensemble 3229, and GYOLO 2248; similarly in BDD [21], our CertainNet 26802, CenterNet 20210, ensemble 16440, and GYOLO 15395. These values also show the limitations of the ensemble, confirming that contrasting predictions within its models, especially in highly uncertain scenarios, such as out-of-domain data (e.g., BDD100K and nuImages in Table II), can happen and thereby significantly degrade the overall detection and uncertainty performance of the system. In general, each of the uncertainty metrics evaluates only a specific aspect and should be considered together with others, and combined with the detection performance (i.e., AP).

Ablation study: The importance of our modifications over DUQ [12] is testified by the results shown in Table III, where

A6 represents our full approach. In particular, the baseline A0, which is the adaptation of DUQ [12] from image classification to object detection, could not converge properly, due to the severe limitations described in Section III-B, worsened by the high imbalance of KITTI [6]. The biggest improvement was brought by A1 replacing the gradient penalty $\mathcal{L}_{grad}$ of DUQ with our hyperspace regularization $\mathcal{L}_{reg}$, which also improved the consistency of training convergence. Moreover, the AP increased monotonously with the introduction of each of our modifications (Section III-B), while maintaining good uncertainty estimates throughout, thereby showing their significant impact over DUQ (i.e., A0).

C. Qualitative Results

In Figures 1 and 5 we show predictions of our method on challenging scenes of KITTI and out-of-domain samples respectively. The images confirm the large domain gap between KITTI and the two transfer datasets, as described in Section IV-B. The night and rainy settings in Figure 5 from nuImages [22] and BDD100K [21] are completely different from the sunny and overcast scenarios of KITTI [6] on which the model was trained, making these very difficult samples. Nevertheless, the proposed CertainNet correctly detected most objects. Compared to the KITTI predictions in Figure 1, the objectness certainties (values at the top left of each box) are lower for the unusual and difficult to see out-of-domain vehicles in Figure 5. This shows that the model recognized them as out-of-distribution samples. Interestingly, the uncertainty estimates for location and size improved the overlap between sub-optimal predictions and the object, e.g., bottom left of Figure 5, as tested by our UBQ metric.

V. CONCLUSION

In this work we introduced CertainNet, a novel method to estimate the uncertainty of every aspect of 2D object detection. Extensive evaluations showed the benefit of quantifying the uncertainty, especially when transferring to out-of-distribution data for safety-critical applications. Our method substantially improved the generalization ability over previous works, with more accurate predictions and high quality uncertainty estimates, while running in real-time. Therefore, the proposed CertainNet constitutes a valuable contribution towards robust object detection for autonomous driving.

REFERENCES

- [1] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, *et al.*, “A survey of uncertainty in deep neural networks,” *arXiv*, 2107.03342, 2021.
- [2] D. Feng, L. Rosenbaum, and K. Dietmayer, “Towards safe autonomous driving: Capture uncertainty in the deep neural network for LiDAR 3D vehicle detection,” in *IEEE ITSC*, 2018, pp. 3266–3273.
- [3] J. Choi, D. Chun, H. Kim, and H.-J. Lee, “Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving,” in *IEEE/CVF ICCV*, 2019, pp. 502–511.
- [4] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, “Sampling-free epistemic uncertainty estimation using approximated variance propagation,” in *IEEE/CVF ICCV*, 2019, pp. 2931–2940.
- [5] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, “D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry,” in *IEEE/CVF CVPR*, 2020, pp. 1281–1292.
- [6] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *IEEE/CVF CVPR*, 2012, pp. 3354–3361.
- [7] Y. Lee, J.-w. Hwang, H.-I. Kim, K. Yun, and Y. Kwon, “Localization uncertainty estimation for anchor-free object detection,” *arXiv*, 2006.15607, 2020.
- [8] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, “SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds,” in *ISVC*, 2020, pp. 207–222.
- [9] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016, pp. 1050–1059.
- [10] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf, “Evaluating merging strategies for sampling-based uncertainty techniques in object detection,” in *IEEE ICRA*, 2019, pp. 2348–2354.
- [11] M. Sensoy, L. Kaplan, and M. Kandemir, “Evidential deep learning to quantify classification uncertainty,” in *NeurIPS*, 2018, pp. 3183–3193.
- [12] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *ICML*, 2020, pp. 9690–9700.
- [13] M. Raghu, K. Blumer, R. Sayres, Z. Obermeyer, B. Kleinberg, S. Mullainathan, and J. Kleinberg, “Direct uncertainty prediction for medical second opinions,” in *ICML*, 2019, pp. 5281–5290.
- [14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *NeurIPS*, 2017, pp. 6405–6416.
- [15] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks,” *Neurocomputing*, vol. 338, pp. 34–45, 2019.
- [16] A. Harakeh, M. Smart, and S. L. Waslander, “BayesOD: A Bayesian approach for uncertainty estimation in deep object detectors,” in *IEEE ICRA*, 2020, pp. 87–93.
- [17] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv*, 1804.02767, 2018.
- [18] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” *arXiv*, 1904.07850, 2019.
- [19] C. Berger, M. Paschali, B. Glocker, and K. Kamnitsas, “Confidence-based out-of-distribution detection: A comparative study and analysis,” in *UNSURE MICCAI Workshop*, pp. 122–132, 2021.
- [20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [21] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “BDD100k: A diverse driving dataset for heterogeneous multitask learning,” in *IEEE/CVF CVPR*, 2020, pp. 2636–2645.
- [22] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, *et al.*, “nusenes: A multimodal dataset for autonomous driving,” in *IEEE/CVF CVPR*, 2020, pp. 11 621–11 631.
- [23] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *ICML*, 2017, pp. 1321–1330.
- [24] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, “Deep layer aggregation,” in *IEEE/CVF CVPR*, 2018, pp. 2403–2412.
- [25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
- [26] S. Gasperini, P. Koch, V. Dalletta, N. Navab, B. Busam, and F. Tombari, “R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes,” in *IEEE 3DV*, 2021, pp. 751–760.
- [27] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, “3D packing for self-supervised monocular depth estimation,” in *IEEE/CVF CVPR*, 2020, pp. 2485–2494.

5.2.2 Domain Generalization with Adversarial Augmentations (CVPR 2022)

With CertainNet [82] (Section 5.2.1), we addressed domain generalization from the perspective of the method, enhancing predictions via uncertainty estimation. Here, we tackle domain generalization from the perspective of data, synthesizing corner cases via adversarial methods.

In this work, we propose an adversarial method to improve the generalization to out-of-distribution point cloud data. We achieve this with a two-stage approach to first generate plausible adversarial examples and then use them as data augmentation to expand the training data.

Inspired by vector-based adversarial attacks from the imaging domain, we propose a new attack for point clouds based on vector fields. The vectors are learned adversarially, minimizing the model's confidence by shifting the 3D points. Unlike previous works that are powerful attacks generating highly perturbed samples [177, 343], ours produces plausible point clouds thanks to a series of constraints applied to the perturbations. Most notably, the points can only move along the sensor view ray. Then, we enforce the preservation of smoothness along the surfaces by limiting the shift of each point depending on that of its neighbors.

Another significant difference to prior adversarial attacks for point clouds is that our attack is sample-independent, meaning that it does not need to be optimized for a specific input, but it is learned on a large set of inputs, and it maintains its efficacy on unseen point clouds. This generalization largely improves the applicability of our method.

Adversarial training [336] is known to improve robustness against adversarial attacks. It works by introducing the generated examples into the training data, training a new model to generate new adversarial examples that are put into the training data, and iterating to achieve a robust model. However, when the adversarial examples are too far from the training data distribution, which occurs when they are highly perturbed [177, 343], this technique likely deteriorates the performance and generalization capabilities of the model.

Our adversarial examples circumvent this critical issue by positioning themselves within the training data distribution and slightly outside of it with new and diverse yet plausible samples. This is illustrated in Figure 5.2. This changes what can be done with such inputs. In this work, we show their benefits by using them as data augmentation for domain generalization. Other works use GANs to generate data for augmentation [7]. Instead, we keep the augmented samples more plausible by driving the generation starting from the available data and enforcing constraints on the perturbation.

Therefore, we propose 3D-VField, a two-stage method to improve generalization and robustness for scene understanding tasks on point clouds. First, we generate plausible adversarial examples; then, we train using them to augment the training data. We apply this to LiDAR and RGB-D point clouds for the task of 3D object detection. In our journal extension [163], we extend the method to 3D semantic segmentation and also perturb the LiDAR intensity signal.

The effectiveness of this method lies in its ability to introduce new samples in the training data that are otherwise difficult for the model (adversarial examples) yet plausible (constrained). We assess

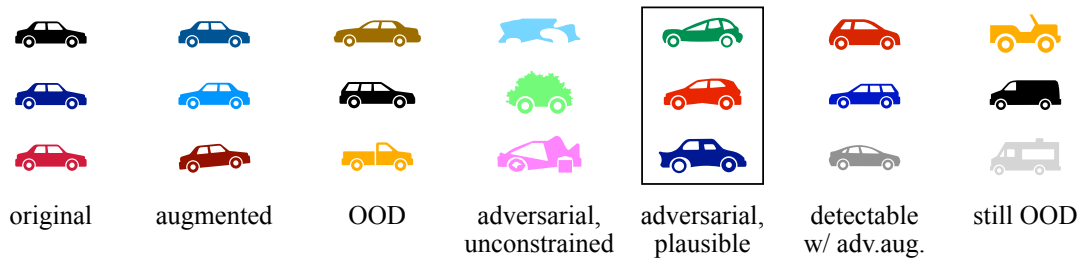


Figure 5.2. Representation of data distributions. In this simplified example, training data is particularly limited, including only one car model and a few colors (original). Augmented samples add more colors, flip and rotate the inputs (augmented), but rare colors and unseen models are out-of-distribution (OOD). Unconstrained adversarial examples (e.g., those of prior works in the point cloud domain [177, 343]) introduce highly unrealistic samples that do not represent existing vehicles. The proposed technique (highlighted with a rectangle) generates plausible adversarial examples that, when used as augmentation, expand the training data, thus increasing its diversity. By constraining the perturbations with domain knowledge, the generated samples may resemble those that can be found in the wild. Thanks to adversarial augmentation, the system can detect more instances and models, including those that could not be previously detected (detectable w/ adv.aug.). Nevertheless, this technique does not allow the system to detect all unseen models, so some may remain out-of-distribution (still OOD).

the boost in generalization by testing the models with out-of-distribution data collected by a different sensor in a different country, introducing a significant domain gap. Furthermore, in this work, we propose a new synthetic dataset containing a variety of corner cases and out-of-distribution samples, including classic, sports, and damaged cars. We call this dataset CrashD.

Across the experiments, we report substantial generalization improvements to out-of-distribution data (up to 59%) and on-par performance in-domain. The plausible expansion of the training data is particularly beneficial to detect long tail samples, such as damaged vehicles, sports cars, and classic cars from the proposed CrashD dataset. 3D-VField also improves the performance with out-of-distribution data collected with another sensor in a different country, showing its effectiveness at generating plausible and diverse samples.

Our journal extension [163] includes various enhancements to 3D-VField, such as adversarial vectors learned to resemble specific classes of choice (i.e., targeted attack), the application to semantic segmentation, and the introduction of the LiDAR intensity as an extra adversarial signal. Furthermore, through extensive new experiments, we introduce insights into the mechanisms of the proposed method, including the analysis of how it alters the decision boundaries of the model, its impact on the robustness against input transformations, and the comparison between adversarial vectors learned for different tasks as well as their impact on the models.

The following includes the main paper, while the supplementary material can be found in Section B.1 and the paper’s license in Appendix C.

3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection

Alexander Lehner^{*,1,2} Stefano Gasperini^{*,1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
Mohammad-Ali Nikouei Mahani² Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

* equal contribution

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022

Summary. In this work, we substantially improve the reliability and generalization of 3D object detectors to out-of-domain data by deforming point clouds during training. We achieve this with 3D-VField: a novel data augmentation method that plausibly deforms objects via vector fields learned in an adversarial fashion. Our approach constrains 3D points to slide along their sensor view rays while neither adding nor removing any of them. The obtained vectors are transferable, sample-independent and preserve shape and occlusions. Despite training only on a standard dataset, such as KITTI, augmenting with our vector fields significantly improves the generalization to differently shaped objects and scenes. Towards this end, we propose and share CrashD: a synthetic dataset of realistic damaged and rare cars, with a variety of crash scenarios.

Contributions. Alexander proposed and implemented the vector field-based adversarial attack. Given my experience with CertainNet [82], I proposed to turn the method towards domain generalization through adversarial augmentation, introducing out-of-distribution data and ultimately showing the effectiveness of the adversarial examples. So, I implemented most of the CrashD dataset. I proposed and guided most of the experiments, including those that Alexander executed to prove the claims. I drove the analysis and discussions, wrote the paper, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: <https://doi.org/10.1109/CVPR52688.2022.01678>.

Copyright Statement. ©2022 IEEE. Reprinted, with permission, from Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Mohammad-Ali Nikouei Mahani, Michael Schmidt, Nassir Navab, Benjamin Busam, and Federico Tombari, '3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection', IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, New Orleans, Louisiana, USA.

The supplementary material can be found in Section B.1 and the license in Appendix C.

3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection

Alexander Lehner^{*,o,1,2} Stefano Gasperini^{*,1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
 Mohammad-Ali Nikouei Mahani² Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

Abstract

As 3D object detection on point clouds relies on the geometrical relationships between the points, non-standard object shapes can hinder a method's detection capability. However, in safety-critical settings, robustness to out-of-domain and long-tail samples is fundamental to circumvent dangerous issues, such as the misdetection of damaged or rare cars. In this work, we substantially improve the generalization of 3D object detectors to out-of-domain data by deforming point clouds during training. We achieve this with 3D-VField: a novel data augmentation method that plausibly deforms objects via vector fields learned in an adversarial fashion. Our approach constrains 3D points to slide along their sensor view rays while neither adding nor removing any of them. The obtained vectors are transferable, sample-independent and preserve shape and occlusions. Despite training only on a standard dataset, such as KITTI, augmenting with our vector fields significantly improves the generalization to differently shaped objects and scenes. Towards this end, we propose and share CrashD: a synthetic dataset of realistic damaged and rare cars, with a variety of crash scenarios. Extensive experiments on KITTI, Waymo, our CrashD and SUN RGB-D show the generalizability of our techniques to out-of-domain data, different models and sensors, namely LiDAR and ToF cameras, for both indoor and outdoor scenes. Our CrashD dataset is available at <https://crashd-cars.github.io>.

1. Introduction

With the established wide-spread progress of learning-based methods tackling a variety of perception tasks (e.g.,

* The authors contributed equally.

^o Contact author: Alexander Lehner (alexander.lehner@tum.de).

Work partly sponsored by the German Federal Ministry for Economic Affairs and Energy (grant 19A19005B), VDA KI-Absicherung project.

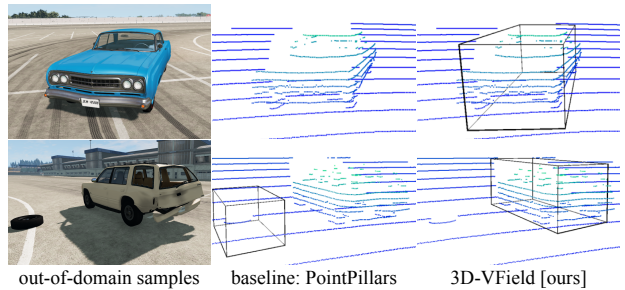


Figure 1. Predictions of PointPillars [18] trained on KITTI [13], without and with our adversarial augmentations on out-of-domain samples from the proposed CrashD dataset. CrashD comprises rare (top) and damaged (bottom) vehicles, resulting in natural adversarial examples [17]. As the models were applied to CrashD without fine-tuning, due to the different object shapes, the standard PointPillars delivered two false negatives and a false positive. Images used with courtesy of BeamNG GmbH.

object detection, semantic and panoptic segmentation), a recent trend denoted a focus shift towards ensuring the safe applicability of these powerful approaches in critical scenarios, such as autonomous driving and robotics [27]. This has led to the pursuit of improving the model robustness and generalization [12, 22, 37], especially against out-of-domain data, which can naturally occur in the real world [17]. Such approaches include domain adaptation [39] and generalization [37], uncertainty estimation [11], simulations [4], and adversarial alterations [35].

Since corner cases are difficult to be captured as they occur in a dynamic real-life scenario, current datasets include only a limited amount of them, if any [5], leaving most of these cases out-of-domain. However, taking care of corner cases is particularly important in safety-critical settings, where long-tail and out-of-distribution samples could lead to dangerous issues if not accounted for during training [5].

While several works have addressed some of these concerns on the imaging domain [4, 11, 16, 26], this is still

mostly unexplored for 3D point clouds [35], also due to the inherent challenges of point clouds, as they are unordered, sparse and irregularly sampled. Nevertheless, as the output of 3D sensors (e.g., LiDAR, ToF cameras), point clouds are especially useful in high automation, where robustness and redundancy are intertwined with safety.

In this context, real non-standard objects, such as damaged and rare cars, or those from different regions, can lead to false negatives, as shown in Figure 1, since the inter-point geometry on which 3D detectors rely is different than usual. While these examples can naturally occur in the real-world [17], they can also be generated artificially with adversarial attacks [14]. This kind of approaches show the vulnerabilities of a model, which can then be addressed to improve robustness. Recent adversarial point cloud alteration methods [35] have tackled this problem to improve the generalization to out-of-distribution data. However, despite being effective attacks, existing adversarial deformation strategies [19, 40] are sample-specific, lack wide-applicability, and by being designed without considering a 3D sensor, are mostly unconstrained in space [19].

In this work, we substantially improve the generalization capability of 3D object detectors to out-of-domain data, bridging this gap by deforming point clouds during training. We propose 3D-VField: a novel adversarial augmentation method that learns to deform point clouds via widely-applicable and sample-independent vector fields (i.e., collections of vectors linked to a set of points in a given space). Our deformations preserve the overall object shape, only slide points along the view ray, and do not add or remove any points. After learning a vector field, we use it to alter objects as data augmentation. The main contributions of this paper can be summarized as follows:

- We raise awareness on natural adversarial examples, such as those represented by damaged and rare cars, around their ability to fool popular 3D object detectors.
- We propose 3D-VField: a sensor-aware adversarial point cloud deformation method based on vector fields able to increase the generalization of 3D object detectors to out-of-domain samples via data augmentation.
- We introduce and publicly release CrashD: a dataset of damaged and rare cars. Extensive experiments on four outdoor and indoor datasets, namely KITTI [13], Waymo [33], our CrashD, and SUN RGB-D [30], show the wide applicability of our approach.

2. Related Work

Our work is about adversarial augmentation to improve the generalization of 3D object detectors for point clouds. In this section we provide a brief overview of existing approaches in these neighboring fields.

2.1. Improving Generalization

Generalization to unseen data is a highly desirable property for any learning-based approach [37]. Unseen data includes any samples on which a model has not been trained on, comprising both out-of-domain and in-domain data (e.g., validation set), depending on the size of the domain shift. In particular, domain generalization deals with improving the performance on a target domain, without any knowledge about it [37], in contrast to domain adaptation which has access to the target data [39]. These works can be grouped in two broad categories: those acting on the model itself, and those operating on the input data.

Among the former category, model regularization strategies are commonly used to reduce overfitting [31] or address domain generalization [3]. Estimating the model uncertainty was also found beneficial for out-of-domain data [11]. Moreover, specific architectures can be found via search algorithms to improve robustness [22].

A different category of works targets generalization by manipulating the input data. Towards this end, it is possible to leverage pretraining and multi-task learning to improve on out-of-distribution samples [2]. Additionally, synthetic data can be included to increase the accuracy on rare classes [4]. Data augmentation methods [16, 32, 45] also belong to this category. Among these, there are adversarial approaches, which extended the training data with altered inputs learned in an adversarial fashion as a way to improve generalization [26, 35, 36].

The method we propose in this work addresses domain generalization (i.e., does not use any target information) and belongs to the data category, specifically to the adversarial approaches, which are detailed in Section 2.2.

2.1.1 Generalization for 3D Object Detection

In the context of generalization, some works addressed the task of 3D object detection, which is also the focus of this work. Simonelli et al. [29] created virtual views normalizing the objects with respect to their distance, to better generalize to samples at different depths in the image domain. Tu et al. [35] improved the generalization towards cars with roof-mounted objects, via adversarial examples on LiDAR point clouds. Wang et al. [39] used domain adaptation to fill the gap between vehicles from multiple countries and different LiDAR sensors.

2.2. Adversarial Examples

Adversarial examples are input alterations designed to lead a model to false predictions [14, 34]. A variety of works explored adversarial examples in the image domain [9, 23, 24, 41, 44], where pixel perturbations imperceptible to humans are able to fool the target model. Alaifari et al. [1] deformed images using a different adversarial vector

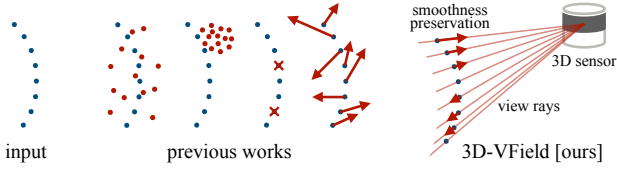


Figure 2. Adversarial deformations introduced by previous works, compared to ours. Other methods add, drop or move points with minor constraints. Ours only slides points along the view ray, while preserving shapes and occlusions.

field learned for each sample. Wang et al. [38] proposed adversarial morphing fields to alter image pixels spatially and fool classifiers. However, this topic is still mostly unexplored on point clouds, especially those captured by 3D sensors (e.g., LiDAR, ToF camera).

2.2.1 Adversarial point clouds

Adversarial methods for 3D point clouds can be grouped in three categories: generation if they add points, removal if they remove points, and perturbation if points are only shifted. Then we present the methods from the perspective of generalization to out-of-domain samples.

Generation and removal Xiang et al. [40] pioneered adversarial point clouds proposing a series of methods, some of which added points to fool the shape recognition. Cao et al. [8] showed the vulnerability of LiDAR-based methods against adversarial objects added to the scene. Similarly, Tu et al. [35] added adversarial meshes on top of cars. A different line of works explored sensor attacks, adding points by means of a spoofing device [7]. Conversely, removal methods adversarially learn to discard a few critical points [43].

Perturbation Xiang et al. [40] also proposed the first two adversarial perturbation approaches. One is the iterative gradient L2 attack, which is an adaptation of PGD from the image domain [20], optimizing for a minimal deformation constrained by the L2 norm. Another approach is the Chamfer attack, which uses the Chamfer distance (CD) between the original and the deformed object to decrease the perceptibility of the attack [19]. The CD is measured by averaging the sum of the distances of the nearest neighbor from each point of the original point cloud to the deformed one. Using this distance function encourages point shifts across the surface of the object. Our method is closely related to the iterative gradient L2 attack, but we do not learn a vector for each point of each sample. Instead, we learn a sample-independent vector field and introduce further constraints to improve our deformations. Liu et al. [19] investigated perturbations more noticeable than the ones of Xiang et al., while producing continuous shapes by altering neighboring points accordingly. Cao et al. [6] 3D printed

adversarial objects to fool multi-modal (LiDAR and camera) detectors.

Generalization Several works on adversarial point clouds were proposed targeting the ModelNet dataset [15, 19, 40], which comprises a set of synthetic 3D point clouds resembling various object shapes. Since ModelNet was not created with a 3D sensor, these foundation works often produce unrealistic outputs [19, 40], that were not intended to improve the generalization of the models, but rather set the basis for adversarial attacks on point clouds [40]. Additionally, these mechanisms are sample-specific, making their applicability limited [15, 19, 40]. Instead, Tu et al. [35] explored the impact on LiDAR object detection of meshed objects, such as canoes and couches, synthesized on top of a car roof. Moreover, they attacked these meshes in an adversarial fashion, and used them to defend the detector, thereby improving its robustness and generalization capability to unseen samples with roof-mounted objects.

Our work sets itself apart from all sample-specific methods [1, 19, 40, 43], as we construct a single highly transferable and generic set of perturbations. Similar to the work of Tu et al. [35], we aim to improve the generalization to out-of-domain samples. However, compared to theirs, as can be seen in Figure 2, we do not add any points, making ours a perturbation method. Additionally, unlike Tu et al., by not making any assumptions on the object nor the kind of sensor, our method has a wider applicability, from indoor to outdoor settings. Plus, we improve realism by taking into account occlusion constraints, which were ignored so far, and making our deformations sensor-aware, as we only shift points along the sensor ray. Additionally, our method differs from all the ones above also because it generates adversarial point clouds via transferable learned vector fields, which has not been explored yet.

3. Method

We now illustrate our method, based on deforming point clouds to account for natural object variations, thereby improving the generalization of 3D object detectors to out-of-domain data via adversarial augmentation. As shown in Figure 3, we achieve this by adversarially learning a vector field (Section 3.1). Once trained, this vector field can be frozen and then applied to any previously seen or unseen objects, after scaling it to match the target size and constraining the points movement to preserve shapes and occlusions (Section 3.2). We apply it to deform all objects of its class, which we use as data augmentation (Section 3.3).

3.1. Adversarially learned vector field

We create a lattice of uniformly spaced 3D vectors within a 3D bounding box. Since the aim is to perturb the point cloud without adding or removing points, vectors are an immediate representation of this set of point shifts. This

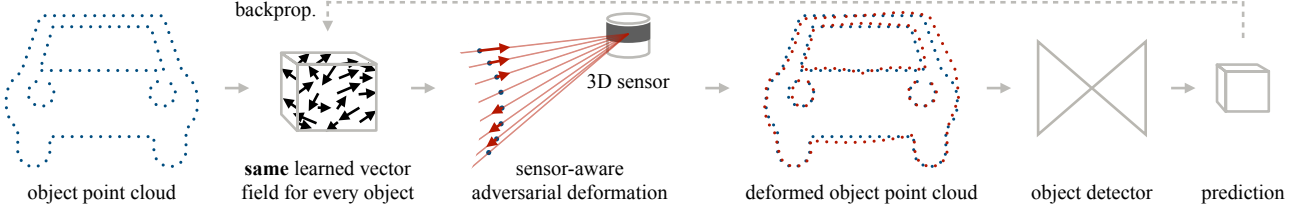


Figure 3. Overview of the proposed 3D-VField. We first learn a vector field adversarially to plausibly deform objects, taking constraints into account. The modified scenes are later used as augmentations to improve the generalization to unseen object shapes.

allows for both compactness and transferability, since the same learned vector field can be applied to any target object. To construct such a vector field, we discretize the space of a default bounding box B_o with a step size t to obtain root coordinates $\mathbf{f}$ in 3D space and assign an empty vector $\mathbf{v} = (x, y, z)$ to each root. B_o is defined by width w , height h , length l , orientation angle α and its center $\mathbf{c} = (x, y, z)$.

Adversarial loss We use a binary cross entropy loss to suppress all relevant bounding box proposals, following [35]. We consider a proposal as relevant if the prediction confidence score $s > 0.1$. $\mathcal{Q}$ is the set of relevant proposal q , where each q has a confidence score s . We minimize s , weighed by the 3D IoU with the the ground truth q^* :

$$\mathcal{L}_{\text{adv}} = \sum_{q, s \in \mathcal{Q}} -\text{IoU}(q^*, q) \log(1 - s). \quad (1)$$

By repeatedly reducing the confidence score while training the vector field, the detector misses the object or predicts a misaligned box. During training, we apply the same vector field to each target object in every scene, minimizing the loss on the whole dataset. At each optimization step, the vectors are updated, resulting in differently deformed point clouds of target objects, which eventually lead to different predictions. As $\mathcal{L}_{\text{adv}}$ smoothly converges, the performance of the detector, against which the vector field is optimized, decreases. Once trained, the vectors can be used for data augmentation.

3.2. Objects Deformation

Before applying a vector field, we scale it to match the target object size. Manipulating the points through these vectors, we constrain their movement as described below.

Optical ray consistency To help generalization and preserve the sensor’s physical constraints when generating deformations, we employ a simple sensor model in which the 3D points can only be moved across the optical ray. We first compute the ray $\mathbf{u}_i$ between the 3D sensor and each point $\mathbf{p}_i$, which determines the deformation direction for each point. Then we calculate the deformation vectors $\mathbf{r}_i$, for each $\mathbf{p}_i$ by projecting its nearest vector $\mathbf{v}_i$ onto the ray $\mathbf{u}_i$. Points are therefore only moved by $\mathbf{r}_i$.

Regularizing the deformations We limit the perturbation of the points by restricting the vectors with $\|\mathbf{v}\|_\infty < \epsilon$ following the standard PGD L_∞ attack [20]. We then ensure shape smoothness along the object surface by sampling multiple k neighboring vectors to move a given 3D point. For each j -th nearest neighbor we calculate the euclidean distance d_{ij} between each point $\mathbf{p}_i$ of the object and its nearest vector $\mathbf{v}_{ij}$ from the vector field. The final shift $\mathbf{m}_i$ of each point is calculated by weighting the deformation vectors $\mathbf{r}_{ij}$ with their corresponding distance d_{ij} :

$$\mathbf{m}_i = \frac{\sum_{j=1}^k d_{ij} \mathbf{r}_{ij}}{k} \quad (2)$$

This allows for a more gradual depth difference between neighboring points, as neighboring vectors with opposite directions would lead to almost no movement of the affected point. Thus, shape smoothness is preserved and less irregular deformations are produced.

Relative rotation We found that using a single vector field for all objects present in the dataset leads to very low amounts of deformation. Due to the various object poses, its vectors would be pointing in all directions, decreasing its efficacy. We circumvent this and allow for a larger degree of alignment between neighboring vectors, by first clustering all the objects in the dataset w.r.t. the relative orientation between object and sensor, and then learning G different fields, one for each cluster.

3.3. Adversarial Data Augmentation

During training of the object detector, we perturb the input point clouds by using the adversarially learned vector fields as data augmentation. This increases the robustness, given that the learned deformations are structurally-consistent, and are therefore more capable than standard augmentations (e.g., scaling, flip, rotation) of resembling out-of-domain car shapes, such as vehicles from a different country [39]. We increase the variability by learning N different vector fields for each of the G rotations (Section 3.2). During training, we randomly select only one object in the scene, and we deform it with a randomly chosen vector field out of the N possible ones for its relative rotation. This high

variability ensures that the model learns both normal and deformed objects, and that each sample can be deformed differently across training, thereby preventing overfitting to specific deformations.

4. Experiments and Results

4.1. Experimental Setup

Datasets We conducted our experiments on four different datasets. Three of them are autonomous driving LiDAR-based: KITTI [13], the Waymo Open Dataset [33], and the proposed synthetic CrashD, which we introduce below. Additionally, we apply our method also on the indoor SUN RGB-D dataset [30], showing its wide applicability. **KITTI** is a popular 3D object detection benchmark recorded in Germany. We adopted a standard split [18], which comprises 3712 training and 3769 validation LiDAR point clouds, where we used the *car* class, reporting on the standard *easy*, *moderate* and *hard*. We evaluated models trained on KITTI (without any fine-tuning) on Waymo and our CrashD to assess the generalization capability of the models to out-of-domain data, particularly critical for autonomous driving. The **Waymo** dataset is a challenging large-scale collection of real scenes recorded in various locations of the USA. It is highly diverse with different weather and illumination conditions, such as rain and night. Furthermore, in the Supplementary Material we show the wide-applicability of our techniques on time-of-flight (ToF) cameras with the **SUN RGB-D** dataset.

CrashD dataset To quantify the generalizability on out-of-domain samples, we produced a synthetic dataset named CrashD. As this includes various types of cars, such as normal, old, sports and damaged, it comprises a variety of plausible vehicle shapes, thereby serving as a valuable out-of-domain test. Specifically, the crashes are individually generated with a realistic simulator [21] and distinguished depending on the intensity, namely *light*, *moderate*, *hard*, as well as the kind of damage: *clean* (i.e., undamaged), *linear* (i.e., frontal or rear), and *t-bone* (i.e., lateral). The randomly and automatically generated 15340 scenes were captured by a 64-beam LiDAR configured to mimic the KITTI one. Each scene presents between 1 and 5 vehicles, with visible damages, before being repaired and placed at the same locations to collect the *clean* set, resulting in a total of 46936 cars. We are releasing this data publicly, as an out-of-domain evaluation benchmark for models trained on KITTI [13], Waymo [33] or similar datasets. Further details can be found in the Supplementary Material.

Evaluation metrics We evaluated the object detection performance on the standard **AP**, with a 3D IoU threshold of 0.7 for KITTI and CrashD, 0.5 for Waymo, and the standard 0.25 for SUN RGB-D. To measure the quality of the adversarial perturbations we followed Tu et al. [35] using

the **attack success rate** (ASR) metric. It measures the percentage of objects that become false negatives after undergoing an adversarial alteration. For the ASR, we considered an object detected if its 3D IoU was larger than 0.7.

Network architectures We used four different 3D object detectors. PointPillars [18] voxelizes the scene in vertical columns (i.e., pillars) from the bird’s eye view, using PointNet for feature extraction. Second [42] voxelizes the point cloud and uses a learned voxel feature encoding. Part-A² Net [28] is an extension of PointRCNN that predicts intra-object part locations for improved accuracy. VoteNet [25] (Supplementary Material) is based on PointNet++ and Hough voting. While the first three are mostly used for autonomous driving, VoteNet is used indoor.

Implementation details We constructed each vector field within B_o with $w = 1.8\text{m}$, $h = 1.6\text{m}$, $l = 4.6\text{m}$ and a step size of $t = 20\text{cm}$ resulting in 1656 vectors per vector field. If not stated otherwise, we grouped objects by relative rotations with $G = 12$ groups, and set $N = 6$. During the perturbation stage, we moved points according to their $k = 2$ nearest vectors and deform only along the sensor ray. For the PGD optimization, we used Adam with a learning rate of 0.05. The distance threshold was set to $\epsilon = 30\text{cm}$. Each vector was randomly initialized from a uniform distribution with values between -1cm and 1cm. We trained all models using PyTorch and MMDetection3D [10] on a single NVIDIA Tesla V100 32GB GPU.

Prior works and baseline We focused on object detection and compared with other adversarial methods. All models were applied on PointPillars [18], unless otherwise noted. As point perturbation methods we used the iterative gradient L2 [40] and the Chamfer attack [19]. For generation we used [40] adding 10% and [43] removing 10% of the objects points. For a fair comparison, we trained all on the same KITTI dataset split [18], with $\epsilon = 30\text{cm}$, then we altered the point clouds as data augmentation with the same settings as ours (i.e., random selection of one object per scene to augment). Moreover, we combined ours with the domain adaptation statistical normalization (SN) strategy of [39]. Following [39], after computing the average box dimensions in the target datasets (i.e., Waymo and CrashD), we scaled the source (i.e., KITTI) point clouds within the ground truth boxes accordingly and fine-tuned the trained models with this altered target-aware source data.

4.2. Quantitative Results

Adversarial methods and generalization Table 1 shows the comparison between our 3D-VField and related adversarial approaches when applied on PointPillars [18] in the context of generalization. In particular, we report other adversarial perturbation methods, such as the iterative gradient L2 [40] and the Chamfer attack [19], adversarial generation [40], as well as adversarial removal [43]. Augment-

Architecture	Method	KITTI				→ Waymo	→ CrashD			
		easy	AP mod.	hard	ASR	AP	AP normal		AP rare	
							clean	crash	clean	crash
PointPill. [18]	no augm. [18]	70.00	61.88	56.23	-	30.68	1.79	0.93	3.92	2.33
	no obj. sampl. [18]	83.83	74.14	68.30	-	37.85	50.36	36.44	28.70	20.02
	PointPillars [18]	88.24	77.11	74.55	-	40.86	65.20	43.67	34.14	22.48
	iter. grad. L2 [40]	86.24	76.92	73.84	*95.9	39.86	58.65	41.86	35.92	23.69
	Chamfer att. [19]	87.15	77.05	74.07	*99.8	40.54	56.84	39.56	36.29	24.73
	advers. gener. [40]	86.12	76.39	73.18	*91.6	40.55	57.75	38.03	35.73	24.18
	advers. remov. [43]	86.51	76.85	74.04	*86.1	40.32	66.52	48.88	41.42	28.10
	3D-VField [ours]	87.05	77.13	75.55	63.4	44.61	67.95	52.87	43.40	30.37
	SN dom. adapt. [39]	-	-	-	-	49.27	79.42	72.59	60.53	48.23
	[ours] + SN [39]	-	-	-	-	51.32	92.14	87.28	86.26	76.42
Second [42]	Second [42]	88.93	78.68	76.87	-	42.45	72.73	56.74	41.85	32.84
	3D-VField [ours]	88.87	78.56	76.81	54.9	43.51	76.54	60.51	47.47	36.14
Part-A ² [28]	Part-A ² [28]	89.60	79.16	78.52	-	49.76	83.05	63.25	74.03	52.33
	3D-VField [ours]	89.65	79.26	78.62	50.5	56.08	88.80	73.80	81.10	61.34

Table 1. Comparison of models trained on KITTI [13] towards out-of-domain data (without any fine-tuning), namely Waymo validation set [33] and our CrashD datasets, as well as on the KITTI validation set. Each method applies a data augmentation (for adversarial ones ASR is measured on their adversarial examples), or performs domain adaptation (only SN [39] in this work), resulting in the reported APs. →: transfer from KITTI. *: being sample-specific, the adversarial method had to be trained on the validation set of KITTI.

ing with the adversarial examples of our 3D-VField did not reduce the overall in-domain AP compared to PointPillars, but brought numerous benefits in terms of out-of-domain generalization. As demonstrated by Wang et al. [39], the transfer from KITTI to **Waymo** is particularly challenging due to the different shapes and sizes of the vehicles found in Germany and the USA, as well as the 50% higher point density and the narrower field of view [33]. This test assesses the quality of the generated deformations with respect to real vehicle shapes found in a different country. On Waymo our 3D-VField delivered more than 9% relative improvement over PointPillars and the other adversarial methods, and 13% over Part-A² [28], proving the benefit of our added sensor-awareness on real and challenging out-of-domain data. On the right of Table 1 we report the results on the proposed **CrashD**. It can be seen that despite the transfer from KITTI, the AP on *clean normal* cars is relatively high for all approaches, likely because those samples are not particularly difficult. However, when damaging those exact same vehicles and placing them at the same locations (*crash*), the detection performance dropped. This shows the effort required for the methods to relate these to the cars learned on KITTI, and proves them as natural adversarial examples. Similarly, with *rare* cars (i.e., old and sports cars), the AP dropped even more, quantifying the domain shift from *normal* vehicles. *Rare crash* cars, by combining the two out-of-domain aspects (i.e., rarity and damage),

were the hardest for all methods, reducing the AP from *normal clean* by up to two thirds (PointPillars). Nevertheless, our method improved significantly over the detectors and the other adversarial approaches for all transfers and categories. This can be attributed to our adversarial augmentations introducing diversity in the training data, while being sensor-aware. In particular, the sensor-awareness ensures that the deformed point clouds are still plausible, thereby better resembling possible out-of-domain samples, such as those of Waymo and CrashD. Among the other adversarial approaches, only removing points [43] improved generalization to CrashD, probably because it preserved the overall point clouds. Nevertheless, [43] was not beneficial on Waymo, which features denser point clouds and more challenging real scenes.

Combination with data augmentations As adversarial data augmentation, our 3D-VField is not alternative to different augmentation strategies, but can be applied in combinations with others. In Table 1 we show how common data augmentation techniques impact the detections for PointPillars [18]. Using no augmentations (no augm.) critically reduced the APs, especially on CrashD at IoU 0.7 (Table 1). At IoU 0.5, this resulted in an AP on *normal clean* of 65.59, while the baseline [18] delivered 98.91. Introducing standard augmentations (no obj. sampl., e.g., flip and rotation) improved, but adding the popular object sampling [18] (PointPillars) increased the APs further. On top, our aug-

# G	K. ASR $\uparrow$	K. <i>mod.</i>	$\rightarrow$ Waymo	# vectors
1	55.08	77.32	40.43	10K
12	63.37	77.13	44.61	120K
360	44.84	77.06	40.30	3.6M

Table 2. Our 3D-VField trained on KITTI (K.) with varying amounts of relative rotations G . $\rightarrow$: transfer no fine-tuning.

mentations substantially improved all transfers, without decreasing the in-domain performance.

Combination with domain adaptation By addressing domain generalization, our approach does not use any target information. Therefore, ours is not alternative to domain adaptation methods [39], which make use of target data. However, similarly to other data augmentation strategies, our 3D-VField can be combined with domain adaptation techniques. As shown in Table 1, such combination further boosts the performance on challenging out-of-domain data. By altering the objects size via the statistical normalization (SN) of [39], the AP on Waymo increased. Constrained by the high amount of false positives and negatives, when combined with SN, ours retained a margin of over 2% compared to PointPillars with SN. Moreover, the AP on CrashD improved dramatically across all categories, especially for the hardest *rare crash* group. The results show how, despite a substantial increase in AP from PointPillars [18], SN alone did not reach the full potential of the detector. Only when combined with ours, the AP doubled (*normal crash*) and more than tripled (*rare crash*) over PointPillars, without using any extra target information. This shows the benefit of this combination, and reiterates the added value of incorporating adversarially deformed objects via data augmentation to improve generalization to out-of-domain samples.

Adversarial methods as attacks In terms of ASR (Table 1), our approach is not as strong as the other adversarial methods, namely the iterative gradient L2 [40], the Chamfer attack [19], adversarial generation [40] and removal [43]. However, this is expected as our vector fields are sample-independent, compared to their point-to-point deformations being sample-specific. Due to this reason, their alterations had to be learned directly on the KITTI validation set, on which the ASR was measured. Nevertheless, a very high ASR means the altered objects are unrecognizable, which does not aid generalization. The goal of our method is not having a detector fully miss the attacked objects (high ASR), but rather deforming them to improve the performance on out-of-domain data. Towards this end, the perturbed objects need to be at the same time altered enough to add diversity to the training data, and not be too far apart from the training distribution to avoid confusing the detector. We found this balance by learning our vector fields ad-

Method	KITTI		$\rightarrow$ W.	$\rightarrow$ CrashD	
	<i>mod.</i>	ASR		<i>n.,clean</i>	<i>r.,crash</i>
P.P. [18]	77.11	-	40.86	65.20	22.48
no learn	76.36	10.1	41.62	62.94	21.75
unleash	76.82	97.7	40.95	60.43	27.55
ray con.	76.35	59.5	41.03	59.82	29.16
full	77.13	63.4	44.61	67.95	30.37

Table 3. Ablation on the deformation constraints imposed by our method, compared to PointPillars (P.P.) [18]. Trained on KITTI. $\rightarrow$: transfer no fine-tun.; W.: Waymo.

versarially, while preserving the objects shape and the sensor realism with our added constraints.

Different 3D detectors In Table 1, we also compare the performance of our 3D-VField when paired with different 3D object detectors, namely PointPillars [18], Second [42], and Part-A² [28]. Remarkably, using the proposed adversarial augmentation improved the AP of Part-A² on Waymo by a large margin. The superiority of Part-A² over the other detectors can be attributed to its part-awareness [28], which might have set its focus on the most relevant object parts (e.g., wheels) and their relationships to identify cars also in out-of-domain settings. For Second [42], the performance on KITTI turned out lower than the one reported in [10], despite using the same settings and framework. This reduced AP affected both the baseline [42] and our approach. Nevertheless, adding our adversarial deformations significantly improved the generalization of all three detectors to out-of-domain data, despite training our vector fields solely against PointPillars. This shows the wide applicability and transferability of our techniques.

Specificity-generalization trade-off Table 2 shows that by varying the amount of relative rotations G , a trade-off arises between generalization, attack specificity (i.e., strength on individual samples by overfitting to the training data), and storage (i.e., amount of vectors). $G = 12$ offers a good balance. With the extreme $G = \#$ of objects, ours would become sample-specific, inheriting the weaker generalization capabilities of [19, 43]. While these methods needed to be trained on the validation set, allowing for high ASRs (Table 1), our vectors were learned on the training set. So with high G , ours overfitted on the training data, which is visible evaluating on the validation set. Our augmentation strategy learns only 1656 3D vectors to perturb objects. However, by training with $G = 12$ and $N = 6$, the amount of vectors increased to 120K. Conversely, the sample-specific iterative gradient L2 [40] and the Chamfer [19] attacks required 10.9M and 12.6M vectors for training and validation sets respectively. This shows the easy applicability of our 3D-VField.

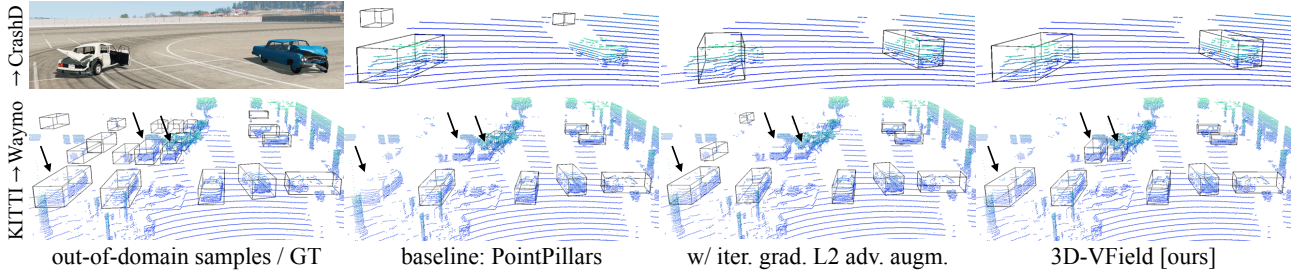


Figure 4. Predictions on challenging out-of-domain samples from the proposed CrashD (top) and Waymo [33] (bottom). Models based on PointPillars [18] trained on KITTI (without fine-tuning). Iterative gradient L2 [40] and ours trained with adversarial augmentation.

Ablation study on deformation constraints As we introduced the sensor-awareness and the surface smoothness constraints to our deformations, we investigate their impact in terms of generalization to out-of-domain data. In Table 3, we report this comparison when limiting the deformations to $\epsilon = 30$ cm. It can be seen that not learning the perturbations, but applying all our constraints (no learn) could already be a beneficial augmentation technique, as it improved the transfer to Waymo. Instead, removing all constraints, but learning the vector fields (unleash) delivered a strong ASR of 97.7%. This significantly increased the AP on the CrashD *rare* cars. When deforming with sensor-awareness (ray con.), ASR reduced, but the AP on the most difficult transfer settings (i.e., *rare crash*) improved. Our full model 3D-VField, adds the distance smoothing (Section 3.2) delivering superior transfer capabilities. Furthermore, increasing the maximum deformation ϵ to 40 or 60 cm, improved the ASR to 73.3% and 87.1%, but as augmentation decreased the AP on KITTI by 1% and 1.7%, respectively. This means that higher deformations do not generalize well, as their plausibility decreases, while 30 cm offers a good trade-off.

4.3. Qualitative Results

In Figure 4 we compare the transfer predictions from KITTI to CrashD and Waymo [33] of the standard PointPillars [18], augmented with ours and the iterative gradient L2 adversarial approach [40], which is the closest to ours in terms of adversarial deformation (Section 2). For CrashD, as seen in the quantitative results (Section 4.2), the iterative gradient L2 method delivered better detections compared to not using any adversarial augmentations [18], but our 3D-VField outperformed it, with a more aligned box for the left damaged car. The figure also shows the severity of the *hard* damages present in CrashD, and how adversarial augmentation helps to detect such challenging samples. For the difficult transfer KITTI $\rightarrow$ Waymo (Section 4.2), it can be seen that all methods had troubles detecting the cars with few points in the parking lot on the left. Furthermore, PointPillars [18] ignored 3 recognizable cars with a high amount

of points, while augmenting with the iterative gradient L2 caused missing 2 of them and detecting 2 further ones, albeit with misaligned boxes. Instead, despite missing further ones, our method was able to recognize these visible cars.

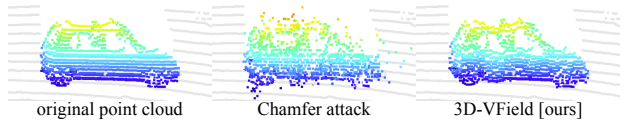


Figure 5. Example deformations by our method and the Chamfer attack [19] on a car of the KITTI validation set [13].

Figure 5 confirms that the strong ASR of the Chamfer attack [19] seen in Table 1 corresponds to unrecognizable objects. It also provides an example of the minor deformations introduced by our adversarial vector fields. By preserving the overall shape of the car and its surfaces, ours allowed for superior generalization to unseen data.

We refer to the **Supplementary Material** for more results on indoor settings, transferability, robustness against noise, detailed evaluations on CrashD, and various ablation studies on grouping and aggregation strategies, as well as the amount of deformed objects during training.

5. Conclusion

In this paper we presented 3D-VField: an adversarial augmentation method for point clouds to improve the object detection performance on natural adversarial examples and out-of-domain data, such as rare, damaged cars, or vehicles from different regions. Towards this end, 3D-VField produces plausible shapes used as data augmentation. Extensive experiments showed the high generalization and transferability of the proposed approach, from indoor to outdoor settings, on both real and synthetic data. Furthermore, we proposed and released CrashD: a new benchmark to challenge 3D object detectors on out-of-domain data, including various kinds of damaged and rare cars.

References

- [1] Rima Alaifari, Giovanni S. Alberti, and Tandri Gauksson. ADef: An iterative algorithm to construct adversarial deformations. In *Proceedings of the International Conference on Learning Representations*, 2019. 2, 3
- [2] Isabela Albuquerque, Nikhil Naik, Junnan Li, Nitish Keskar, and Richard Socher. Improving out-of-distribution generalization via multi-task self-supervised pretraining. *arXiv preprint arXiv:2003.13525*, 2020. 2
- [3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018. 2
- [4] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 863–873, 2020. 1, 2
- [5] Daniel Bogdoll, Jasmin Breitenstein, Florian Heidecker, Maarten Bieshaar, Bernhard Sick, Tim Fingscheidt, and Marius Zollner. Description of corner cases in automated driving: Goals and challenges. In *IEEE/CVF International Conference on Computer Vision Workshop*, pages 1023–1028, 2021. 1
- [6] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and LiDAR: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 176–194, 2021. 3
- [7] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on LiDAR-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2267–2281, 2019. 3
- [8] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019. 3
- [9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 2
- [10] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>, 2020. 5, 7
- [11] Stefano Gasperini, Jan Haug, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, Benjamin Busam, and Federico Tombari. CertainNet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, 7(2):698–705, 2021. 1, 2
- [12] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 751–760, 2021. 1
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 2, 5, 6, 8
- [14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, 2015. 2
- [15] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. AdvPC: Transferable adversarial perturbations on 3D point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 241–257. Springer, 2020. 3
- [16] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadam, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 1, 2
- [17] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 1, 2
- [18] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 5, 6, 7, 8
- [19] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3D point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 88–104. Springer, 2020. 2, 3, 5, 6, 7, 8
- [20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*, 2018. 3, 4
- [21] Pascale Maul, Marc Mueller, Fabian Enkler, Eva Pigova, Thomas Fischer, and Lefteris Stamatogiannakis. BeamNG.tech technical paper, 2021. 5
- [22] Jisoo Mok, Byunggook Na, Hyeokjun Choe, and Sungroh Yoon. AdvRush: Searching for adversarially robust neural architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12322–12332, 2021. 1, 2
- [23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582. IEEE, 2016. 2
- [24] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Proceedings of the IEEE European Symposium on Security and Privacy*, pages 372–387, 2016. 2

- [25] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. [5](#)
- [26] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. [1](#), [2](#)
- [27] Martin Rabe, Stefan Milz, and Patrick Mader. Development methodologies for safety critical machine learning applications in the automotive domain: A survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 129–141, 2021. [1](#)
- [28] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. [5](#), [6](#), [7](#)
- [29] Andrea Simonelli, Samuel Rota Buló, Lorenzo Porzi, Elisa Ricci, and Peter Kotschieder. Towards generalization across depth for monocular 3D object detection. In *Proceedings of the European Conference on Computer Vision*, pages 767–782. Springer, 2020. [2](#)
- [30] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. [2](#), [5](#)
- [31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. [2](#)
- [32] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 1262–1270, 2019. [2](#)
- [33] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. [2](#), [5](#), [6](#), [8](#)
- [34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014. [2](#)
- [35] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for LiDAR object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13713–13722, 2020. [1](#), [2](#), [3](#), [4](#), [5](#)
- [36] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 5339–5349, 2018. [2](#)
- [37] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4627–4635, 2021. [1](#), [2](#)
- [38] Run Wang, Felix Juefei-Xu, Qing Guo, Yihao Huang, Xiaofei Xie, Lei Ma, and Yang Liu. Amora: Black-box adversarial morphing attack. In *Proceedings of the ACM International Conference on Multimedia*, pages 1376–1385, 2020. [3](#)
- [39] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in Germany, test in the USA: Making 3D object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
- [40] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3D adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9128–9136, 2019. [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [41] Chaowei Xiao, Bo Li, Jun-yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating Adversarial Examples with Adversarial Networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3905–3911, July 2018. [2](#)
- [42] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, Oct. 2018. [5](#), [6](#), [7](#)
- [43] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019. [3](#), [5](#), [6](#), [7](#)
- [44] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019. [2](#)
- [45] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *Proceedings of the International Conference on Learning Representations*, 2021. [2](#)

5.2.3 Segmentation of Known and Unknown Objects and Regions (ICCV 2023)

In the previous sections, 3D-VField [162] (Section 5.2.2) and CertainNet [82] (Section 5.2.1) consider corner cases and out-of-distribution data but only look at objects associated with a predefined set of classes (e.g., *cars*). In this section, we address completely unknown objects of any category unrestrictedly, exploring both autonomous driving and more general indoor and outdoor scenarios.

In this work, we propose a new segmentation setting dealing with known areas and unseen, unknown objects without prior knowledge about unknowns. We name this holistic segmentation. Then, we introduce a method called U3HS, which tackles this challenging problem by exploiting uncertainty estimation, instance-aware embeddings, and clustering.

This work developed from the need for reliability with completely unseen object categories. As shown in Figure 5.3, unknown classes comprise anything that is not part of the training classes definition. Since real-world scenarios are unconstrained, the variety of objects that can be encountered is enormous and cannot be fully specified in the training data. Alarming, prior works and tasks either ignore unknown objects [150], cannot distinguish individual objects [179], can only re-identify already seen unlabeled objects [127], require external knowledge about unknowns [32] (such as that of a large vision-language model [346]), or need highly vast amounts of training data also covering unknowns or information about them [151, 243].

Instead, we are interested in understanding what can be achieved without extra information, relying only on the training distribution of the known categories. This is motivated by the necessity to reduce assumptions and improve applicability and reliability. Thus, we want to extend segmentation to any unknown object without needing prior knowledge beyond that of a small set of known classes. The ability to recognize individual unknown objects without having ever seen them or accessed information about them is a critical skill for an autonomous system to operate safely in unseen scenarios. This enables tracking them and estimating their trajectories.

While training data may be available for many objects that are not part of the few standard semantic classes found in common datasets, it cannot realistically cover all possible objects and scenarios to the extent needed by learning-based models (Section 4.4). Additionally, not all objects can be easily described or interpreted by a vision-language model, such as CLIP [243]. This includes broken and unidentified pieces following an accident. As shown by Radford et al. [243], CLIP is not immune from limitations with long tail and out-of-distribution samples. Compared to a standard segmentation model, however, an open-vocabulary approach has a larger pool of objects it can recognize, meaning that its out-of-distribution boundary is further than that experienced by standard methods. Nevertheless, such a boundary still exists, and the objects outside of it cannot be recognized. Furthermore, CLIP associates text and image embeddings in the same space [243], so to retrieve the object masks, the category names must also be available for unknown objects.

Most notably, the proposed holistic segmentation setting removes the limitations on the pool of recognizable objects, now unrestricted to any object, and reduces the assumptions on the training data, which now does not need to contain any unknown or unlabeled object. In turn, this simplifies the requirements for the data collection, as no unknowns and out-of-distribution object categories need to

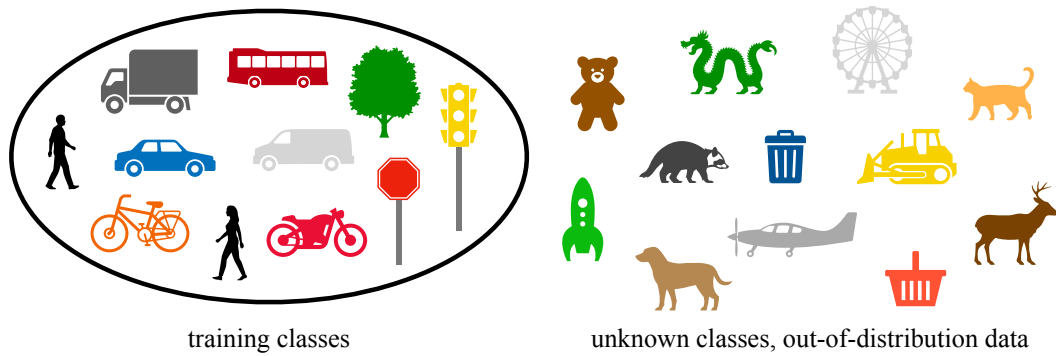


Figure 5.3. Examples of unknown objects (right) given a standard set of known categories learned during training (left). Depending on the training data and the evaluation settings (e.g., the real world), out-of-distribution data can comprise a wide range of objects and classes. With segmentation tasks, when not properly taken into account, unknown objects are systematically and wrongly associated with a class from the pool of known ones.

be part of the training set. It also improves the applicability by leaving the setting unconstrained as in real-life scenarios and not requiring any information about unknowns at inference time.

We address this new, challenging problem with a novel method called U3HS. Essentially, U3HS builds upon existing works in the areas of panoptic segmentation, uncertainty estimation, contrastive learning, and clustering. For known regions, U3HS learns to associate pixel-level embeddings with objects and class prototypes. These embeddings are learned such that those belonging to the same object are close in the latent space, far from other objects, regardless of their semantic class, making them instance-aware. For unknown objects, U3HS estimates the uncertainty at the pixel level over the semantic classes and flags as unknown all the pixels having a high uncertainty (i.e., larger than a threshold). The instance-aware embeddings of the highly uncertain regions are then clustered into unknown instances.

Despite the difficulty of addressing the introduced holistic segmentation problem, the proposed U3HS can find unknown objects thanks to a relatively simple yet fundamental aspect: the distribution of the known categories in the training data is different from that of the unknown objects. By leveraging this straightforward point and learning the training distribution, there is no need for extra information about unknowns, as they can be found as those objects not belonging to the known distribution. This principle is crucial for out-of-distribution detection [179]. Then, the added difficulty of the proposed holistic segmentation setting is segmenting individual instances of unknown objects. With U3HS, we address this by learning the concept of generic objects through class-agnostic and instance-aware embeddings.

While foundation models, such as CLIP [243] and SAM [151], provide promising segments on a variety of inputs thanks to their comprehensive training data and large models, U3HS has the unique advantage of being trained on a relatively small, standard dataset containing only known objects (e.g., Cityscapes [50] or COCO [175]), yet deliver segments of unknown instances, too, without any restriction in terms of the objects in input. Nevertheless, the generalization strength of foundation models could be exploited by equipping them with uncertainty estimation and the techniques presented in this work, thereby improving their performance with long tail samples and making them operate beyond their training data distribution.

The following includes the main paper, while the supplementary material can be found in Section B.2 and the paper's license in Appendix C.

Segmenting Known Objects and Unseen Unknowns without Prior Knowledge

Stefano Gasperini^{1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
Nassir Navab¹ Benjamin Busam¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² BMW Group ³ Google

IEEE/CVF International Conference on Computer Vision (ICCV), 2023

Summary. Panoptic and semantic segmentation approaches assign a known class to each pixel in the input. This inevitably enforces decisions that systematically lead to unreliable predictions for objects outside the training categories. In this work, we propose to extend segmentation with a new setting which we term holistic segmentation. Holistic segmentation aims to identify and separate objects of unseen, unknown categories into instances without any prior knowledge about them while performing panoptic segmentation of known classes. We tackle this new problem with U3HS, which finds unknowns as highly uncertain regions and clusters their corresponding instance-aware embeddings into individual objects. By doing so, for the first time in panoptic segmentation with unknown objects, our U3HS is trained without unknown categories, reducing assumptions and leaving the settings as unconstrained as in real-life scenarios.

Contributions. Given my prior experiences with Panoster [84] and CertainNet [82], I initiated this project in mid-2021 to explore the benefits of uncertainty estimation for unknown objects in the context of panoptic segmentation. The project was partly done in its early stages in the context of a Master's thesis, which I supervised. I executed a large part of the implementation, proposing and guiding the rest. I proposed the new setting and ran the experiments to validate the claims. I drove the analysis and discussions, wrote the paper, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: https://openaccess.thecvf.com/content/ICCV2023/html/Gasperini_Segmenting_Known_Objects_and_Unseen_Unknowns_without_Prior_Knowledge_ICCV_2023_paper.html.

Copyright Statement. ©2023 IEEE. Reprinted, with permission, from Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Nassir Navab, Benjamin Busam, and Federico Tombari, 'Segmenting Known Objects and Unseen Unknowns without Prior Knowledge', IEEE/CVF International Conference on Computer Vision (ICCV), October 2023, Paris, France.

The supplementary material can be found in Section B.2 and the license in Appendix C.

Segmenting Known Objects and Unseen Unknowns without Prior Knowledge

Stefano Gasperini^{1,2,◦} Alvaro Marcos-Ramiro² Michael Schmidt²
Nassir Navab¹ Benjamin Busam¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² BMW Group ³ Google

Abstract

*Panoptic segmentation methods assign a known class to each pixel given in input. Even for state-of-the-art approaches, this inevitably enforces decisions that systematically lead to wrong predictions for objects outside the training categories. However, robustness against out-of-distribution samples and corner cases is crucial in safety-critical settings to avoid dangerous consequences. Since real-world datasets cannot contain enough data points to adequately sample the long tail of the underlying distribution, models must be able to deal with unseen and unknown scenarios as well. Previous methods targeted this by re-identifying already-seen unlabeled objects. In this work, we propose the necessary step to extend segmentation with a new setting which we term *holistic segmentation*. Holistic segmentation aims to identify and separate objects of unseen, unknown categories into instances without any prior knowledge about them while performing panoptic segmentation of known classes. We tackle this new problem with U3HS, which finds unknowns as highly uncertain regions and clusters their corresponding instance-aware embeddings into individual objects. By doing so, for the first time in panoptic segmentation with unknown objects, our U3HS is trained without unknown categories, reducing assumptions and leaving the settings as unconstrained as in real-life scenarios. Extensive experiments on public data from MS COCO, Cityscapes, and Lost&Found demonstrate the effectiveness of U3HS for this new, challenging, and assumptions-free setting called *holistic segmentation*. Project page: <https://holisticseg.github.io>.*

1. Introduction

Since neural networks have achieved unprecedented performance in perception tasks (e.g., object detection and semantic segmentation), there has been a growing interest

[◦] This work was conducted while working at BMW Group.
Contact author: Stefano Gasperini (stefano.gasperini@tum.de).

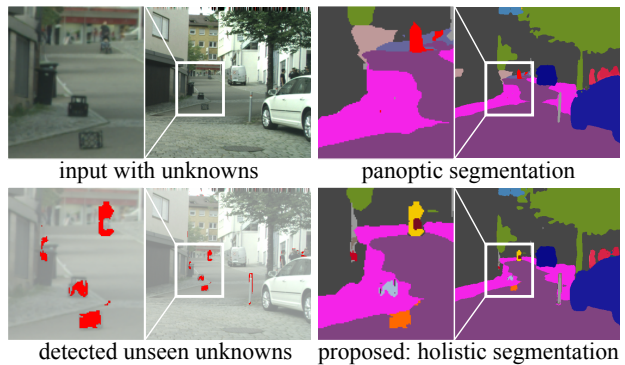


Figure 1. State-of-the-art panoptic segmentation methods [13] cannot deal with unseen classes from [52] (top right). Instead, our U3HS addresses the proposed holistic segmentation setting. U3HS finds unseen unknowns (bottom left) and separates them into instances (bottom right) without prior knowledge about unknowns.

in ensuring their safe deployment, especially important for safety-critical scenarios, such as autonomous driving and robotics [25]. Recently, several works have been proposed to improve robustness and generalization by addressing corner cases and out-of-distribution data [6, 65, 22], via domain adaptation [70], adversarial augmentations [44], simulations [1], and uncertainty estimation [54].

Due to the difficulty of collecting corner cases from the long tail of the underlying data distribution, current datasets cannot fully represent the diversity of the world, leaving its vast majority as difficult out-of-distribution (OOD) samples [32, 8]. In safety-critical applications, considering them during development and deployment is of utmost importance [3, 44], or they could cause severe damage.

Furthermore, since the powerful and popular *softmax* highly promotes the probability of the highest *logit*, state-of-the-art methods tend to be overly confident even on wrong predictions [58, 25]. In safety-critical settings, reliable confidence together with interpretability techniques [20, 77] increases trust for downstream tasks [25], e.g., trajectory prediction and path planning. Towards this end, estimating the uncertainty of a model’s output is commonly considered a key enabler for its safe applicability [39, 25].

While several works addressed some of these problems for image classification [58, 46, 54, 61] and object detection [48, 21], they remain primarily unexplored for dense tasks such as semantic and panoptic segmentation [5, 66]. Compared to object detection, the problem is more severe for dense tasks, where a model needs to provide a prediction for every input unit (e.g., each pixel). So unseen objects (i.e., of new, unseen categories) are systematically and wrongly assigned to one of the limited number of known classes (closed-set), as shown in Figure 1. This has led researchers towards designing new methods that work not only with the available data distribution but also with OOD samples that are not available (open-set), thereby improving robustness against unseen scenarios [38, 46, 5, 44].

Open-set panoptic segmentation [66] segments instances of unlabeled objects in addition to panoptic segmentation of known areas, i.e., the combination of semantic and instance segmentation [41]. Unlike OOD segmentation [38], segmenting unknown instances enables tracking and trajectory prediction. Prior works [66, 36, 72] tackled this problem by relying on seeing unlabeled categories during training. They learned these categories through the *void* class (i.e., unlabeled) and assumed unknowns to be within ground truth *void* regions at training time and inside *void* predictions at test time. By doing so, unknowns are transformed into learned unlabeled instances (i.e., essentially known objects) [36], constraining the open-set task. Mainly intended to segment already-seen unlabeled objects [36, 72], current works cannot deal with the wide variability of unknowns and corner cases outside the training data.

In this paper, we propose the necessary next step for panoptic segmentation to include object categories outside the training data (i.e., unseen unknowns). We term the new setting holistic segmentation. The aim is to identify and segment unseen unknowns into instances while segmenting known classes in a panoptic fashion without any external nor prior knowledge about unknowns. Unseen categories pose new challenges compared to already-seen unlabeled ones [36], requiring new solutions. Estimating the uncertainty is a key step towards finding the knowledge boundaries of a model, leaving the problem unconstrained and reducing assumptions on the training data. Therefore, we propose U3HS: Unseen Unknowns via Uncertainty estimation for Holistic Segmentation. The main contributions of this paper can be summarized as follows:

- We introduce the setting of holistic segmentation, which highlights the importance of not using prior knowledge about unknown objects (e.g., text), and leaves the setup unconstrained as in real scenarios.
- We tackle this new setting with U3HS: the first panoptic framework to deal with unseen, unknown object categories, able to segment and separate them.

- We provide uncertainty measures for the output of U3HS to further improve its safe applicability.

2. Related Work

Closed-set panoptic segmentation Combining semantic and instance segmentation, panoptic segmentation [41] distinguishes *things* (countable classes) from *stuff* (amorphous). The vast majority of methods are top-down [71, 60, 53, 40, 34, 50]: two-stage exploiting box proposals and *thing* masks from Mask R-CNN [30], and filling up *stuff* areas with a semantic branch. Bottom-up are proposal-free, e.g., Panoptic-DeepLab [13]: they segment semantically and cluster instances within *thing* regions [13, 64]. A different line of work proposed end-to-end solutions [23, 63] where instance and semantic segments are delivered directly by treating instance segmentation as a class-agnostic classification task. Others explored self-attention [34], videos [67, 11, 47], scene graphs [69, 68], multi-task learning [16, 28], neural fields [43], or text descriptions [18]. Our U3HS framework deals with unseen unknowns and extends [13] via instance-aware embeddings.

Zero-shot learning aims to predict unseen classes outside the training set [4, 78, 75] with the help of external knowledge [10, 26], e.g., a language model [76], used to build semantic spaces common between seen and unseen classes [74]. While zero-shot methods detect only unseen classes at inference time, generalized zero-shot approaches also detect seen ones [9, 56], similarly to the proposed holistic segmentation. Also **open-vocabulary** methods are zero-shot [35, 27, 73, 18]. They exploit language models such as CLIP [57] to describe unknowns. CLIP has been trained on unknown classes and is treated as an oracle, as it is assumed to be able to describe every unknown, allowing open-vocabulary and zero-shot approaches to identify them. However, this implies that unknown classes are known, e.g., to CLIP. Moreover, CLIP is not immune to corner cases and long tail samples [57]. This limits the pool of objects that these methods can recognize. As shown in Figure 2, holistic segmentation segments unseen objects too, but unlike zero-shot and open-vocabulary, it does not use any external support (e.g., text descriptions of unknowns), such that objects of unseen categories are segmented solely by learning on known ones. More recent than this work, SAM [42] is a strong foundation model. Unlike SAM, ours does not use any prompts and outputs semantic classes.

Uncertainty estimation Epistemic uncertainty is caused by the model itself, while aleatoric is due to the input [39, 25]. OOD data typically results in high epistemic due to a knowledge gap. Single deterministic approaches are sampling-free and provide predictions and uncertainty estimates with the same model [25]. Among these, DUQ [61] learns class representatives and compares them with input features. SNGP [46] improves the awareness to domain

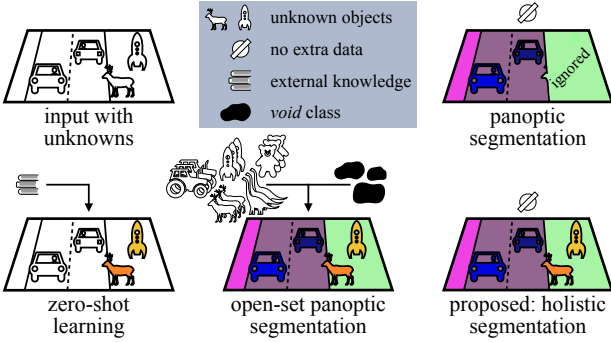


Figure 2. Comparison between closed-set (top right) [41] and open-set [36] panoptic segmentation, zero-shot learning [78], and the proposed holistic segmentation setting. While zero-shot and open-set panoptic methods commonly leverage knowledge about unknown objects, holistic segmentation does not use any priors.

shifts via weight normalization and a Gaussian process. DPN [58] predicts the parameters of a Dirichlet distribution, and uses a Dirichlet density function for each probability assignment and its uncertainty. Various works estimated uncertainty for object detection [49, 48, 21], and segmentation [55, 5, 59], improving robustness and generalization. While most compute uncertainty only to provide it as extra output [25, 59], our U3HS can be paired with any of the above techniques to find unknown objects, which it then separates into instances for holistic segmentation.

Open-set perception Open-set tasks are similar to generalized zero-shot learning [9], with the fundamental difference that here no external knowledge on the unseen classes is used [75]: inference is based only on what was learned from the training data. Uncertainty estimation helps to identify knowledge boundaries [49]. Bayesian SSD [49] uses Dropout sampling for open-set object detection. MLUC [6] tackles this for LiDAR point clouds via metric learning and unsupervised clustering. Open world recognition [2, 7] labels detected unknowns and adds them to the training set. Pham et al. [51] grouped regions perceptually to known and unknown instances, exploiting edges, boxes, and masks. Recently, several works tackled open-set semantic segmentation, telling apart unknown areas from known classes [38, 33, 29]. Among those not learning from OOD data, DML [5] uses metric learning and SML [38] acts as post-processing, standardizing the max *logits*, improving the class distributions. Instead, our work separates unseen, unknown objects into instances and segments known areas, addressing the proposed holistic segmentation.

Open-set panoptic segmentation The pioneering OSIS [66] was the first in this direction. Applied to LiDAR data, it exploits 3D locations to cluster unlabeled points into instances. Later, EOPSN [36] extended Panoptic FPN [40] and grouped its proposals into clusters. At training time, EOPSN clusters similar unlabeled objects across multiple

inputs. When surrounded by known segments, it labels an unlabeled object and uses it to learn to segment its instances. Instead, DDOSP [72] uses a known-unknown class discriminator and class-agnostic proposals. However, as these approaches were intended to re-identify already-seen unlabeled objects, they all rely on seeing unknown data at training time [66, 36, 72]. They all cluster into instances what falls in the predicted *void* class, learned as a fallback (i.e., must be in the training samples) and assumed to contain all unknowns. Since datasets are limited [44], by requiring to learn from unknowns and the *void* class, existing works are not designed to deal with any completely unseen object, as their pool of identifiable unknowns is also limited [36, 72]. For these reasons, they solve only part of the problem. Instead, by using no OOD data at training time and preventing learning priors for unknowns, the proposed holistic segmentation differs from the way open-set panoptic segmentation has been tackled so far [66, 36, 72]. As shown in Figure 2, our setting allows to segment without constraints and separate even unseen unknown objects. In contrast to all existing approaches, the proposed U3HS neither relies on seeing unknowns at training time nor learns the *void* class. U3HS is the first method to solve this unconstrained, assumptions-free holistic segmentation setting.

3. Proposed Setting: Holistic Segmentation

As shown in Figure 2, the proposed setting of holistic segmentation is a logical extension of open-set panoptic segmentation [66, 36]. To make the setting unconstrained as real-life scenarios, we aim to identify and separate **any unseen, unknown object** into instances while segmenting known classes. Other settings allow to include unknowns in the training data [66, 36, 72] (e.g., within the *void* class) or use information about them [78], and only re-identify already-seen unlabeled objects [36]. Instead, we focus on the case where **no information is available about unknowns**. Therefore, holistic segmentation is more challenging, makes no assumptions about the training data (e.g., the presence of *void*, with unknowns in it), leaves the problem unconstrained to any object, and simplifies data collection as no unknowns need to be in the training data. Formal definition and metrics follow open-set panoptic segmentation [66, 36]. As shown in Figure 2, the outputs are comparable, but the definition of unknowns differs (here unseen), as well as the inability to learn from unknowns. Unknown instances can then be used for downstream tracking, trajectory prediction, path planning, or active learning.

4. Proposed Framework: U3HS

In Figure 3, we show a representation of U3HS, targeting holistic segmentation. U3HS outputs instances of unseen unknowns by clustering instance-aware embeddings corre-

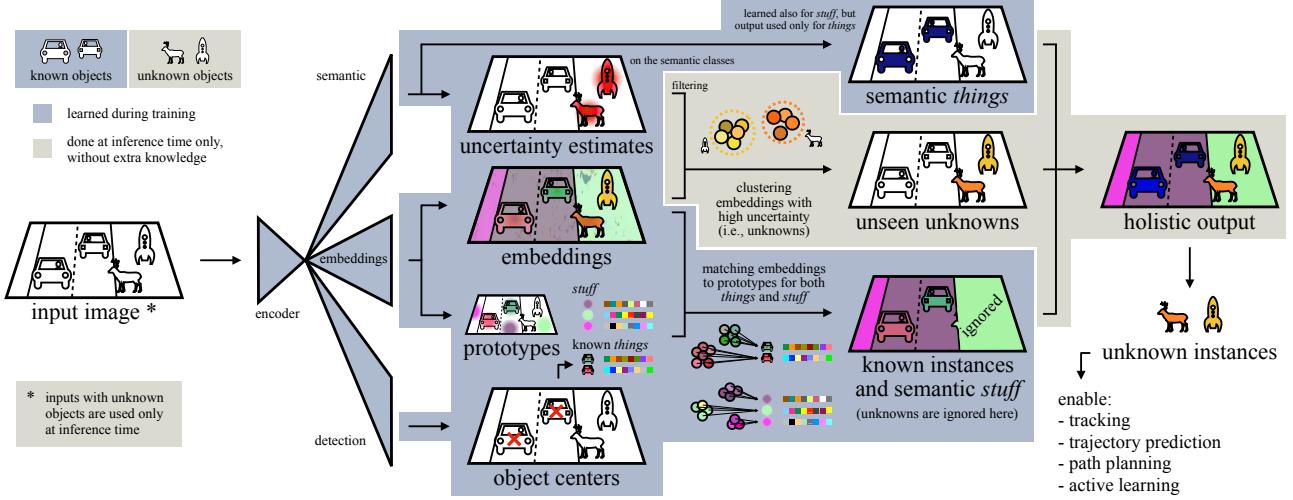


Figure 3. The proposed U3HS framework. Uncertainty is estimated in the semantic branch, and with the instance-aware embeddings, it determines unknown instances. Known instances are found via center regression and formed by grouping embeddings with their prototypes.

sponding to highly uncertain regions (Section 4.2). We use uncertainty estimation to distinguish known classes from unknowns, while embeddings are learned solely on known objects with panoptic segmentation (Section 4.1).

4.1. Panoptic Segmentation for Known Classes

Our approach for closed-set panoptic segmentation builds upon learning instance-aware embeddings. As shown in Figure 3, an encoder extracts features from an input image and propagates them to different decoders: 1) a semantic branch performing semantic segmentation and uncertainty estimation to identify unknown regions (Section 4.2); 2) a detection branch identifying object centers similarly to Panoptic-DeepLab [13]; and 3) an embeddings branch, with two separate heads, for prototypes and embeddings.

We make the embeddings instance-aware via discriminative loss functions (Section 4.3) and by concatenating the detection branch features to prototype and embeddings heads. Embeddings and detections are made also semantic-aware by concatenating the semantic logits to the last layers of the heads. Prototypes Ω are feature vectors the prototype head predicts to represent objects and *stuff* classes. While computed at each pixel, only the features at object centers C are considered *thing* prototypes, plus one for each *stuff* class. This is inspired by the size heatmap in [79]. *Thing* prototypes are $\Omega_{th} = \{(\mu_o, \sigma_o^2) \in \mathbb{R}^F \times \mathbb{R}^+ : o \in I\}$, one for each object o of all detected instances I . F is the embedding size, μ and σ^2 are mean and variance. *Stuff* prototypes are $\Omega_{st} = \{(\mu_k, \sigma_k^2) \in \mathbb{R}^F \times \mathbb{R}^+ : k \in K_{st}\}$, one for each *stuff* class $k \in K_{st}$ independently.

Similarly to [66], the embeddings head predicts embeddings $\phi_{(i,j)} \in \mathbb{R}^F$ for each pixel (i, j) , then matches them with their prototype $\omega \in \Omega$ as follows. We compute association scores $\hat{y}_{(i,j),\omega}$ for each pixel (i, j) and prototype as:

$$\hat{y}_{(i,j),\omega} = -\|\phi_{(i,j)} - \mu_\omega\|^2 / 2\sigma_\omega^2 \quad (1)$$

Compared to [66], we relax the problem by not including the term $-\frac{F}{2} \log \sigma_\omega^2$, and let the embedding variance be indirectly controlled by the final task, which naturally bounds it (shown empirically in Section 5.2). Then, we keep the prototype variance σ_ω^2 strictly positive by using *softplus*.

At inference time, for *things*, the semantic class of each instance is determined by majority voting of its semantic branch predictions, ensuring output consistency. Instead, the ID is computed from the highest score in Eq. 1. For *stuff* regions, we follow [66], determining the semantic classes by associating the pixel embeddings to the prototypes Ω_{st} via the highest scoring class from Eq. 1. This decoupling allows semantic awareness throughout the model.

4.2. Dealing with Unseen Unknown Objects

We find unknown segments by relying on uncertainty estimates, which can help identify the knowledge boundaries of a model [58, 46]. Specifically, instead of predicting the *void* class and searching in it for unknowns as in [66, 36], we estimate the uncertainty related to the semantic segmentation predictions and consider as unknown the areas with a high associated uncertainty. Although our framework can flexibly work with various uncertainty estimators (Section 5.2), here we exemplify it with DPN [58, 37], which we extended from image classification to semantic segmentation, and also improved its convergence in this context. We chose DPNs as they allow for minimal modifications at training time, i.e., replacing the *softmax* with a strictly positive activation function while providing good uncertainty estimates on OOD data without training on such data [58].

Following [58], we consider the evidence $e_k = \alpha_k - 1$ as a measure of the number of hints given by data for a pixel

to be assigned to a class $k \in K$ known classes, with α_k being the parameters of the Dirichlet distribution $Dir(\alpha)$. We compute the uncertainty as $u = K / \sum_{k=1}^K \alpha_k$. Given that the class probabilities $\mathbf{p} = \{p_k : k = [1, \dots, K]\}$ follow a simplex (i.e., are positive and sum to 1), the class assignment corresponds to a Dirichlet distribution parametrized over the evidence, as the probability density function:

$$D(\mathbf{p}|\alpha) = B(\alpha)^{-1} \prod_{k=1}^K p_k^{\alpha_k - 1} \quad (2)$$

with: $B(\alpha) = \prod_{k=1}^K \Gamma(\alpha_k) / \Gamma(\sum_{k=1}^K \alpha_k)$

where Γ is the gamma function and $B(\alpha)$ is the K -dimensional multinomial beta function [58].

We apply this to semantic segmentation by predicting a concentration parameter $\alpha^{(i,j)}$ for each pixel (i, j) , replacing the last layer with the smooth *softplus* activation function, thus converting the *logits* to a strictly positive vector, which we use as evidence $e^{(i,j)}$ in the Dirichlet distribution. We learn this distribution with the semantic loss $\mathcal{L}_s$ minimizing the negative expected log likelihood of the correct class $Y^{(i,j)}$, for the random variable $\mathcal{X}^{(i,j)} \sim \text{Dir}(\alpha^{(i,j)})$:

$$\begin{aligned} \mathcal{L}_s^{(i,j)} &= -E[\ln \mathcal{X}_{Y^{(i,j)}}^{(i,j)}] \\ &= \psi\left(\sum_{k=1}^K \alpha_{(i,j),k}\right) - \psi(\alpha_{Y^{(i,j)}}) \end{aligned} \quad (3)$$

where ψ is the digamma function (i.e., Γ 's logarithmic derivative) and $\alpha_{(i,j),k}$ is the output of the semantic branch. Due to the difficulty of modeling the target distribution in our holistic setting, we omit the KL term used in [58], simplifying the loss design (Section 5.2). After training on the closed-set data, we consider all pixels (i, j) with an estimated uncertainty $u_{(i,j)} \geq \mu + t \cdot \sigma$ as unknown regions with μ and σ^2 being mean and variance of the uncertainties of all training pixels, and t being a hyperparameter.

Separating unknowns After finding the unknown segments, we cluster their instance-aware embeddings trained only on known objects into individual unknowns using DBSCAN [19]. We find the DBSCAN hyperparameters on the training closed-set data (Appendix). Finally, we re-assign the few DBSCAN's outliers to their originally predicted semantic class, thus ignoring their uncertainty estimates.

4.3. Learning to Find Knowns and Unknowns

We train our models with a combination of four losses. The semantic branch is optimized with $\mathcal{L}_s^{(i,j)}$ (Eq. 3) over the whole image sized $W \times H$ as:

$$\begin{aligned} \mathcal{L}_s &= \frac{1}{WH} \sum_{i,j} -E[\ln \mathcal{X}_{Y^{(i,j)}}^{(i,j)}] \\ &= \frac{1}{WH} \sum_{i,j} \psi\left(\sum_{k=1}^K \alpha_{(i,j),k}\right) - \psi(\alpha_{Y^{(i,j)}}) \end{aligned} \quad (4)$$

As in [13], the detection branch is trained with an L2 loss between predicted $\hat{C}$ and ground truth C center heatmaps:

$$\mathcal{L}_o = \frac{1}{WH} \sum_{i,j} \left(\hat{C}^{(i,j)} - C^{(i,j)}\right)^2 \quad (5)$$

For *stuff*, we use the predicted Ω_{st} as a pseudo label to learn the prototypes Ω . For *things*, the same is done with Ω_{th} at the true instance centers. The prototype loss $\mathcal{L}_p$ is the cross-entropy on the *softmax* of the association scores $\hat{\mathbf{y}}_{(i,j),\omega}$, as $\hat{\mathbf{z}}_{(i,j),\omega} = \exp(\hat{\mathbf{y}}_{(i,j),\omega}) / \sum_{\omega' \in \Omega} \exp(\hat{\mathbf{y}}_{(i,j),\omega'})$, with $\omega_{(i,j)}$ being the pseudo label prototype:

$$\mathcal{L}_p = \frac{1}{WH} \sum_{i,j} -\log(\hat{\mathbf{z}}_{(i,j),\omega_{(i,j)}}) \quad (6)$$

We learn embeddings $\phi_{(x,y)}$ with a discriminative loss [15] $\mathcal{L}_d$ (Appendix). The overall training objective is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_o + \lambda_3 \mathcal{L}_p + \lambda_4 \mathcal{L}_d \quad (7)$$

5. Experiments and Results

5.1. Experimental Setup

Datasets We conducted our experiments on three public datasets, namely Cityscapes [14], Lost&Found [52], and MS COCO [45]. **Cityscapes** is a popular outdoor benchmark. Recorded around 50 different cities, mainly in Germany, it contains 19 classes: 8 *things* and 11 *stuff*. We followed the standard split, with 2975 images for training and 500 as validation set, reporting all metrics on the latter. Also recorded in Germany, the **Lost&Found** dataset contains a variety of unusual OOD objects placed in the middle of the road. We selected it because: 1) it was recorded with the same sensor setup as Cityscapes, allowing seamless transfers and removing the need for fine-tuning; 2) it contains only real images; and 3) unlike similar datasets [3, 8], it provides instance annotations for unknowns. Therefore, it is a challenging complement to Cityscapes for holistic segmentation. We did not train on Lost&Found, but used it only to evaluate models trained on Cityscapes. We report all metrics on the *unknown* class of its 1202 test samples. **MS COCO** is a challenging large-scale benchmark for general image understanding, as it includes a variety of scenarios from indoor to outdoor. The 2017 panoptic split contains 80 *thing* categories, and 53 *stuff* classes. We followed EOPSN [36] by treating as unknown the least frequent 20% *thing* classes (e.g., *bear*, *frisbee*). However, instead of turning their segments into *void* and keeping their images in the training set as in [36], we removed their samples completely and regarded them as unseen unknowns. This reduced the training samples to 98112, with 117 classes to learn. We report on the 827 validation samples with unseen classes.

Evaluation metrics We evaluated the panoptic quality (PQ) metric [41] separately for known classes and unknowns, including recognition (RQ) and segmentation (SQ) qualities. We report PQ on the held-out classes of COCO [45], the unknown class of Lost&Found [52], as well as on the 19 known classes of Cityscapes [14] for both **open and closed** settings. Specifically, in open cases, models detect both knowns and unknowns, while in closed set-

Method	Assumptions	Lost&Found (<i>unseen</i>)		
		PQ	RQ	SQ
EOPSN [36]	data, <i>void</i>	0*	0*	0*
OSIS [66]	data, <i>void</i>	1.45	2.23	65.11
U3HS [ours]	none	7.94	12.37	64.24

Table 1. Segmentation of unseen unknown objects (*unknown* class) of **Lost&Found** [52] test set after training on Cityscapes [14] and transferring with no fine-tuning. *: EOPSN diverged (null TP).

tings, the same models predict only knowns, which in practice means ignoring the uncertainty estimates. By analyzing both, we explore the trade-off between detecting unknowns (open) and the in-domain performance (closed).

Network architecture All our models share the structure with Panoptic-DeepLab [13], using a ResNet50 [31] backbone and decoders following Deep-LabV3+ [12]. ResNet50 was chosen to increase reproducibility with limited resources. As described in Section 4.2, the only modification to the semantic decoder is applying the *soft-plus* activation to quantify the uncertainty. The other branches follow Panoptic-DeepLab for detecting centers and DeepLabV3+ for the embeddings, with two heads.

Implementation details For [14, 52], we used input images sized 1024×512 and batch size 16. For [45], we fed 8 images sized 640×480 . We used the Adam optimizer until convergence, with an initial learning rate of 0.001, which was reduced by 2% at each epoch. We set $t = 3$ for the uncertainty threshold (i.e., 3 times the standard deviation) and $F = 8$ for the embedding size to keep the memory low. We adjusted to the different data distribution of COCO with $t = 1$. The backbone was pre-trained on ImageNet [17]. The losses were weighted $\lambda_1 = \lambda_3 = \lambda_4 = 1$ and $\lambda_2 = 200$ [13].

Prior works We compared our U3HS with open-set panoptic works: OSIS [66], which we adapted from LiDAR point clouds to images, EOPSN and its baselines [36], and DDOSP [72]. Instead of training them directly on the unknown categories being evaluated (as in [36]), we followed their setup [66, 36, 72] by training them with the *void* class as fallback, and applied them to unseen unknowns. All methods followed this setup, except that ours ignored *void*. On COCO, we facilitated other works following the K=5% setting of EOPSN [36], thus turning 4 classes into *void*, so they learned 4 classes less than ours. We repurposed and extended a variety of uncertainty estimators [58, 61, 46] from image classification to semantic segmentation (Appendix). We then extended them to holistic segmentation by incorporating them in our U3HS framework.

5.2. Quantitative Results

Unseen unknowns, L&F Table 1 compares our U3HS with prior approaches when segmenting instances of unseen

Method	Assumptions	COCO (<i>unseen</i>)		
		PQ	RQ	SQ
EOPSN [36]	data, <i>void</i>	0.40	0.50	<u>80.30</u>
<i>void</i> -train [36]	data, <i>void</i>	4.40	5.90	74.80
<i>void</i> -supp. [36]	data, <i>void</i>	4.50	6.00	75.90
DDOSP [72]	data, <i>void</i>	<u>9.30</u>	<u>11.20</u>	82.50
U3HS [ours]	none	9.62	13.20	72.84

Table 2. Segmentation of unseen, unknown objects (20% least frequent, held-out classes) on the validation set of **MS COCO** [45]. All others learn *void* and are taken from [72], added trailing zero.

unknowns from Lost&Found [52]. OSIS [66] was the first to address the more limited open-set panoptic segmentation setting, followed by EOPSN [36]. However, OSIS performance fell short on PQ for unseen unknowns, proving the severe limitation of relying on unknowns at training time. By learning *void*, OSIS achieved the highest SQ, which ignores wrong predictions [41]. Instead, despite numerous attempts, EOPSN [36] did not work: it diverged as soon as the exemplars were mined, obtaining 0 true positives (TP). We attribute this to the inconsistent similarities within the *void* class of Cityscapes, compared to those across existing major classes treated as *void* (e.g., *car* in their setup). This prevented EOPSN from forming meaningful clusters from the proposal features during training [36]. Despite the similar setup to EOPSN [36], OSIS [66] could converge since it does not rely on associating unknowns across images. Our U3HS outperformed OSIS by 5.5 times on PQ.

Unseen unknowns, COCO In Table 2, we show results on the 16 held-out categories of MS COCO [45]. In this case, other works were trained following the K=5% setup of EOPSN [36], where 4 classes were learned as *void* (*car*, *cow*, *pizza*, and *toilet*). While this allows them to learn more meaningful representations of unknowns (as unlabeled), it limits the number of classes they can distinguish semantically, e.g., they cannot identify cars. For the other works, the benefit of expanding the *void* distribution by enforcing the inclusion of a variety of recurring objects (e.g., *pizza* and *cars*) is evident as it allowed EOPSN to converge, although to a low PQ on the unseen objects. DDOSP [72] delivered a PQ similar to ours, albeit requiring to turn some knowns into *void*, learning unknowns via *void* and only 113 classes. Without altering the data nor making any assumption on the training samples (e.g., the presence of unknowns within *void* as in [36, 72, 66]), our U3HS performed the best on the unseen categories, especially on RQ (i.e., ability to form instances of unknowns), while learning the whole set of 117 classes, thereby distinguishing even more classes than the other works. Avoiding data assumptions with respect to unknowns made our U3HS effective at segmenting unseen unknowns across both datasets.

Known-unknown Table 3 reports the performances in-

ID	Method	Lost&Found (<i>unseen</i>)			open Cityscapes			closed Cityscapes		
		PQ	RQ	SQ	PQ	RQ	SQ	PQ	RQ	SQ
-	Panoptic-DeepLab [13]	-	-	-	-	-	-	<u>45.82</u>	<u>57.66</u>	<u>79.46</u>
-	OSIS [66]	1.45	2.23	65.11	39.42	50.20	78.53	39.42	50.20	78.53
A1	[ours] baseline: semantic uncertainty	0.49	0.82	60.16	35.02	44.83	78.10	35.97	46.12	78.00
A2	A1 + relaxed embedding association	3.64	5.27	69.09	42.14	53.46	78.83	43.99	55.98	78.59
A3	A2 + prototype head = U3HS	7.94	12.37	64.24	<u>41.21</u>	<u>51.67</u>	<u>79.77</u>	46.53	58.99	78.87
A4	A3 – reassigning outliers	<u>7.85</u>	<u>12.25</u>	64.11	39.84	49.97	79.75	46.53	58.99	78.87
A5	A4 – majority voting	<u>7.85</u>	<u>12.25</u>	64.11	23.94	30.15	79.41	26.77	33.86	79.06
A6	A4 – semantic embeddings	2.33	3.48	<u>67.01</u>	35.16	43.34	81.13	35.92	44.30	81.07
U1	U3HS [ours] + <i>softmax</i> uncertainty	0.10	0.20	51.45	39.70	50.77	78.20	45.12	56.83	79.40
U2	U3HS [ours] + DUQ [61]	0.56	0.89	62.56	41.68	53.14	78.42	45.90	58.17	78.90
U3	U3HS [ours] + DPN [58]	2.09	3.30	63.43	38.90	49.56	78.49	44.91	56.95	78.85
U4	U3HS [ours] + SNGP [46]	<u>4.65</u>	<u>7.57</u>	61.49	41.02	<u>51.98</u>	<u>78.91</u>	<u>46.23</u>	<u>58.56</u>	<u>78.95</u>
A3	U3HS [ours] + improved DPN [ours]	7.94	12.37	<u>64.24</u>	<u>41.21</u>	51.67	79.77	46.53	58.99	78.87

Table 3. Segmentation comparison of models trained on Cityscapes [14] and transferred to the test set of Lost&Found [52] without fine-tuning. All were trained with the same constraints (e.g., ResNet50 [31], small batch, and image sizes). An ablation study (A1-A6) shows the impact of the main components of U3HS, with A3 being our full approach. A3 is paired with various uncertainty estimators (U1-U4).

domain, under open and closed settings (Section 5.1). Ideally, a method would suffer from no decrease in PQ between the two settings, meaning that its estimates are aligned with the distribution shift between knowns and unknowns. OSIS [66] does not use uncertainty estimation, so it does not have these two operating modes, resulting in identical open and closed-set outputs, as if it had only the open setting (via the prediction of *void*). Conversely, all others suffered from a reasonable decrease when extended to open-set. DUQ [61] had the smallest gap, which could be attributed to its underestimation of the uncertainty, as supported by its low scores on Lost&Found.

Closed-set In Table 3, we also compare our U3HS with Panoptic-DeepLab [13]. For a fair comparison, both approaches and all others were trained with the same backbone, image, and batch sizes (Section 5.1). As these were all smaller than those used in [13] due to the limited resources used, they resulted in a lower PQ than that reported in [13]. Nevertheless, our full approach (A3) achieved a slightly higher PQ on Cityscapes under the same setting. We attribute this to the effectiveness of the instance-aware discriminative embeddings learned by our approach, compared to the offset vectors and grouping used by Panoptic-DeepLab. As the focus is unknowns, experiments with improved training resources are out of the scope of this work.

Uncertainty In Table 3, we compare various uncertainty estimations paired to our U3HS framework (U1-U4, A3). While DUQ [61] and *softmax* underperformed compared to OSIS [66], DPN [58] and SNGP [46] achieved a higher PQ. Nevertheless, our improved DPN paired with our framework outperformed prior methods by a substantial margin (A3). For DPN and SNGP, this can be attributed to the su-

periority of our uncertainty estimates. Compared to OSIS and EOPSN, U3HS’s combination of uncertainty estimation with instance-aware embeddings was more effective than learning *void* when encountering wholly new and unseen objects, such as those found in unconstrained settings (e.g., this transfer to Lost&Found).

Ablation study Table 3 reports an ablation of the main components of our U3HS, showing their benefits for holistic segmentation. Compared to the open-set panoptic OSIS [66], with A1, we reduced assumptions not learning the *void* class, and we added a semantic branch with uncertainty for unknowns, which by itself worsened the performance. However, combining this with a relaxed embedding association (Section 4.1) for *things* and *stuff* improved all metrics (A2). A dedicated prototype head (A3, i.e., full approach) increased them even further, more than doubling the PQ on unknowns (i.e., Lost&Found). Specifically, dedicated heads allow both prototypes and embeddings to be more meaningful and expressive without sacrificing the other. A4 shows the impact of reassigning outliers (Section 4.2). While its effect was limited on unknowns, it was more significant on Cityscapes [14]. Transforming unknown predictions in standard in-domain outputs is relevant only in open settings. A5 shows the effect of majority voting to enforce consistency between the outputs (Section 4.1). This did not affect unknowns since classes are not distinguished among them, but it significantly impacted RQ and PQ on Cityscapes. Finally, A6 shows the importance of learning the embeddings according to their semantic classes. In A6, predictions are made by the dedicated semantic branch without the model learning to distinguish the embeddings semantically. Although this increased SQ, it caused a discrepancy within

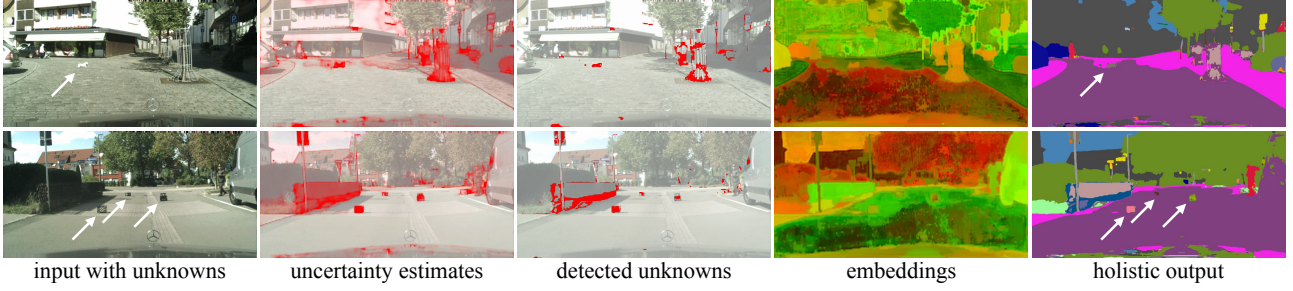


Figure 4. Example predictions of U3HS on OOD data from the Lost&Found [52] test set. The model was trained on Cityscapes [14] and transferred to Lost&Found without fine-tuning. Embeddings are projected to RGB via t-SNE [62]. White arrows mark labeled unknowns.

the model outputs, decreasing RQ and PQ.

5.3. Qualitative Results

Figure 4 shows example predictions of the proposed U3HS. The images illustrate the setting difficulty and provide examples of the OOD objects of **Lost&Found** [52]. These are often small and hard to see, hidden in the shade or far away from the camera. As seen in the quantitative results (Section 5.2), our U3HS could distinguish instances of unknowns (e.g., stroller in Figure 1), albeit leaving room for improvement. While unknowns correctly triggered high uncertainty estimates, their necessary filtering (third col.) sometimes left too few pixels, if any, on unknowns, leading to missed predictions. However, this is to be expected without any access to OOD data. Furthermore, without distinguishing between unknown *things* and unknown *stuff*, also structures (e.g., fence in the lower image) were given an ID. Nevertheless, thanks to our learned instance-aware embeddings, these were not further subdivided but formed a single large instance (e.g., blue in the lower output). Separate unusual *stuff* regions had the same effect, e.g., the structures around the trees in the upper image. This proves that instances are not simply created by separating disjoint OOD segments but are formed using the learned embeddings. As shown in Figure 4, the embeddings are closely coupled with

the uncertainty estimates and the outputs.

Figure 5 reports predictions of U3HS on samples of **MS COCO** [45] containing two held-out classes (i.e., *bear* and *frisbee*). Remarkably, U3HS was able to separate both bears and frisbees into individual instances despite their high inter-class similarity and not having accessed any information about them. This is thanks to the uncertainty estimation and instance-aware embeddings of our U3HS.

Data considerations and limitations Lost&Found [52] introduces a significant domain shift from Cityscapes. By placing real OOD objects on the road, the authors had to choose unusual scenarios (Figure 4), causing the whole scenes to be OOD. This leads to high uncertainty estimates also on a few known areas. As we do not use any OOD data, nothing constrains high uncertainty to unknown segments, decreasing PQ. A similar issue occurs in COCO, albeit less severely, thanks to more training data. However, COCO has no dedicated unknown class, so it had to be extracted from the set of known ones. Nevertheless, results show that uncertainty is highly valuable, allowing to leave the settings unconstrained. U3HS would mainly benefit from improvements in uncertainty estimates, embeddings descriptiveness, and their clustering. So, learning-based clustering [23, 24] could be advantageous.

The **Supplementary Material** includes more details on the proposed holistic segmentation setting, U3HS and the baselines, as well as additional results, including the trade-off between in-domain and OOD performances, failure cases and qualitative comparisons.

6. Conclusion

In this paper we introduced holistic segmentation: a new setting addressing completely unseen unknown objects in unconstrained scenarios. Additionally, we presented U3HS: the first solution for this new problem. Thanks to its uncertainty estimation and instance-aware learned embeddings, U3HS identifies and separates instances of completely unseen unknowns without any information about them, while segmenting known regions. Extensive experiments on multiple datasets showed the effectiveness of U3HS.

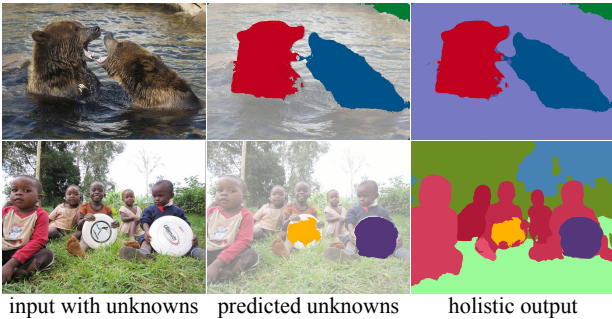


Figure 5. Example predictions of U3HS on OOD data from the COCO [45] validation set. The model had never seen images containing *bear* or *frisbee* (part of the held-out classes), nor had any information about them. Colors represent the predicted instances.

References

- [1] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 863–873, 2020. **1**
- [2] Abhijit Bendale and Terrance Boulton. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902, 2015. **3**
- [3] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes benchmark: Measuring blind spots in semantic segmentation. *Springer International Journal of Computer Vision*, 129(11):3119–3135, 2021. **1, 5**
- [4] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. **2**
- [5] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15333–15342, 2021. **2, 3**
- [6] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-set 3D object detection. In *International Conference on 3D Vision (3DV)*, pages 869–878. IEEE, 2021. **1, 3**
- [7] Jun Cen, Peng Yun, Shiwei Zhang, Junhao Cai, Di Luan, Mingqian Tang, Ming Liu, and Michael Yu Wang. Open-world semantic segmentation for LiDAR point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 318–334. Springer, 2022. **3**
- [8] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMelfYouCan: A benchmark for anomaly segmentation. In *Neural Information Processing Systems - Datasets and Benchmarks Track*, 2021. **1, 5**
- [9] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Proceedings of the European Conference on Computer Vision*, pages 52–68. Springer, 2016. **2, 3**
- [10] Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Ian Horrocks, Jeff Z. Pan, and Huajun Chen. Knowledge-aware zero-shot learning: Survey and perspective. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4366–4373, 2021. **2**
- [11] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 695–714. Springer, 2020. **2**
- [12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818. Springer, 2018. **6**
- [13] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. **1, 2, 4, 5, 6, 7**
- [14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. **5, 6, 7, 8**
- [15] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. **5**
- [16] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5485–5494, 2021. **2**
- [17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. **6**
- [18] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with MaskCLIP. *arXiv preprint arXiv:2208.08984*, 2022. **2**
- [19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. **5**
- [20] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019. **1**
- [21] Stefano Gasperini, Jan Haug, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, Benjamin Busam, and Federico Tombari. CertainNet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, 7(2):698–705, 2021. **2, 3**
- [22] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021. **1**
- [23] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of LiDAR point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021. **2, 8**
- [24] Stefano Gasperini, Magdalini Paschali, Carsten Hopke, David Wittmann, and Nassir Navab. Signal clustering with class-independent segmentation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3982–3986. IEEE, 2020. **8**

- [25] Jakob Gawlikowski, Cedric Rovile Njietcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021. 1, 2, 3
- [26] Yuxia Geng, Jiaoyan Chen, Xiang Zhuang, Zhuo Chen, Jeff Z Pan, Juan Li, Zonggang Yuan, and Huajun Chen. Benchmarking knowledge-driven zero-shot learning. *Elsevir Journal of Web Semantics*, 75:100757, 2023. 2
- [27] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Proceedings of the European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2
- [28] Kratharth Goel, Praveen Srinivasan, Sarah Tariq, and James Philbin. QuadroNet: Multi-task learning for real-time semantic depth aware instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 315–324, 2021. 2
- [29] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. In *Proceedings of the European Conference on Computer Vision*, pages 500–517. Springer, 2022. 3
- [30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 2
- [31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6, 7
- [32] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 1
- [33] Jie Hong, Weihao Li, Junlin Han, Jiyang Zheng, Pengfei Fang, Mehrtash Harandi, and Lars Petersson. GOSS: Towards generalized open-set semantic segmentation. *Springer The Visual Computer*, pages 1–14, 2023. 3
- [34] Rui Hou, Jie Li, Arjun Bhargava, Allan Raventos, Vitor Guizilini, Chao Fang, Jerome Lynch, and Adrien Gaidon. Real-time panoptic segmentation from dense detections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8523–8532, 2020. 2
- [35] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7031, 2022. 2
- [36] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1184, 2021. 2, 3, 4, 5, 6
- [37] Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being Bayesian about categorical probability. In *International Conference on Machine Learning*, pages 4950–4961. PMLR, 2020. 4
- [38] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized Max Logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15425–15434, 2021. 2, 3
- [39] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2
- [40] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 2, 3
- [41] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2, 3, 5, 6
- [42] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2
- [43] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2
- [44] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3D-VField: Adversarial augmentation of point clouds for domain generalization in 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17295–17304, 2022. 1, 2, 3
- [45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 5, 6, 8
- [46] Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020. 2, 4, 6, 7
- [47] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21033–21043, 2022. 2
- [48] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *Proceedings of the International Conference on Robotics and Automation*, pages 2348–2354. IEEE, 2019. 2, 3
- [49] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection

- in open-set conditions. In *Proceedings of the International Conference on Robotics and Automation*, pages 3243–3249. IEEE, 2018. 3
- [50] Rohit Mohan and Abhinav Valada. EfficientPS: Efficient panoptic segmentation. *Springer International Journal of Computer Vision*, 129(5):1551–1579, 2021. 2
- [51] Trung Pham, Thanh-Toan Do, Gustavo Carneiro, Ian Reid, et al. Bayesian semantic instance segmentation in open set world. In *Proceedings of the European Conference on Computer Vision*, pages 3–18. Springer, 2018. 3
- [52] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting small road hazards for self-driving vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1099–1106, 2016. 1, 5, 6, 7, 8
- [53] Lorenzo Porzi, Samuel Rota Buló, Aleksander Colovic, and Peter Kotschieder. Seamless scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 2
- [54] Janis Postels, Hermann Blum, Yannick Strümpfer, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*, 2020. 1, 2
- [55] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2931–2940, 2019. 3
- [56] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2
- [57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2
- [58] Murat Sensoy, Lance M. Kaplan, and Melih Kandemir. Eventual deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing*, pages 3183–3193, 2018. 1, 2, 3, 4, 5, 6, 7
- [59] Kshitij Sirohi, Sajad Marvi, Daniel Büscher, and Wolfram Burgard. Uncertainty-aware panoptic segmentation. *IEEE Robotics and Automation Letters*, 8(5):2629–2636, 2023. 3
- [60] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. AdaptIS: Adaptive instance selection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7355–7363, 2019. 2
- [61] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020. 2, 6, 7
- [62] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008. 8
- [63] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. 2
- [64] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 108–126. Springer, 2020. 2
- [65] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 1
- [66] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *Proceedings of the Conference on Robot Learning*, pages 384–393. PMLR, 2020. 2, 3, 4, 6, 7
- [67] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2705–2714, 2021. 2
- [68] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Incremental 3D semantic scene graph prediction from RGB sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5074, 2023. 2
- [69] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021. 2
- [70] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. DANet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15769–15778, 2021. 1
- [71] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 2
- [72] Hai-Ming Xu, Hao Chen, Lingqiao Liu, and Yufei Yin. Two-stage decision improves open-set panoptic segmentation. *arXiv preprint arXiv:2207.02504*, 2022. 2, 3, 6
- [73] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Proceedings of the European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2
- [74] Kai Yi, Xiaoqian Shen, Yunhao Gou, and Mohamed Elhoseiny. Exploring hierarchical graph representation for large-scale zero-shot image classification. In *Proceedings of the European Conference on Computer Vision*, pages 116–132. Springer, 2022. 2

- [75] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15404–15414, 2021. [2](#), [3](#)
- [76] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. [2](#)
- [77] Yang Zhang, Ashkan Khakzar, Yawei Li, Azade Farshad, Seong Tae Kim, and Nassir Navab. Fine-grained neural network explanation by identifying input features with predictive information. *Advances in Neural Information Processing Systems*, 34:20040–20051, 2021. [1](#)
- [78] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2593–2602, 2021. [2](#), [3](#)
- [79] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. [4](#)

5.3 Improving Robustness for Depth Estimation

While improving generalization is crucial for expanding the capabilities of a model beyond its training data distribution, in this section, we propose new techniques to mitigate severe, inherent limitations that manifest themselves already within the same data distribution. These limitations are due to common training assumptions that lead to problematic safety concerns for all inputs that violate such assumptions. Mitigating them means improving the models' robustness to entire sets of otherwise highly complicated inputs. Specifically, we focus on the task of monocular depth estimation and the issues that self-supervised models exhibit with dynamic objects and challenging weather and illumination settings. Furthermore, we address inherent sensor issues with adverse weather by extending our techniques to supervised models.

5.3.1 Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)

In this work, we focus on mitigating an inherent issue of monocular self-supervised depth estimation learned from image sequences: the wrong estimation of the depth of dynamic objects. We achieve this with a new approach called R4Dyn, which adds weak radar supervision and a camera-radar fusion. Since radars are widely popular in vehicles already on sale (Section 2.3), the proposed method is highly applicable to production cars.

Learning depth estimation from monocular videos in a self-supervised manner relies on the assumption of a moving camera in a static world [307]. However, the world is dynamic, so these assumptions lead to systematic failures with moving objects. In autonomous driving, all traffic participants are dynamic, so estimating their depth wrongly can cause serious issues. A few previous works have focused on addressing the so-called *infinite depth problem* [94, 102], which occurs when a vehicle drives in front at a similar speed. An example is shown in Figure 5.4. Since such vehicles never get closer across the training sequences, the model overestimates their distance, thinking they are infinitely far, e.g., as far as the sky, which similarly never gets closer throughout the sequences.

In this work, as shown in Figure 5.4, we also highlight the opposite problem of underestimating the distance of incoming traffic. Analogous to the infinite depth problem, whenever vehicles drive towards the ego vehicle in the training sequences, they move significantly faster than the surrounding environment, causing the model to think they are much closer than they actually are and associate them with the closest distances. This problem has been ignored so far, possibly due to the popular KITTI benchmark featuring only a few dynamic scenes [89]. Instead, we experiment on the challenging nuScenes dataset [35], which includes a variety of dynamic scenes, including incoming traffic, where the problem becomes evident.

We build the proposed R4Dyn upon the successes of Monodepth2 [94], which learns depth from a sequence of monocular images without ground truth data. One of the contributions of Monodepth2 is the introduction of the automask loss. This loss ignores the pixels whose value does not change across the frames of the training sequences. Since leading vehicles tend to cover the same area throughout the short sequences, this strategy mitigates the *infinite depth problem*. However, as shown in our

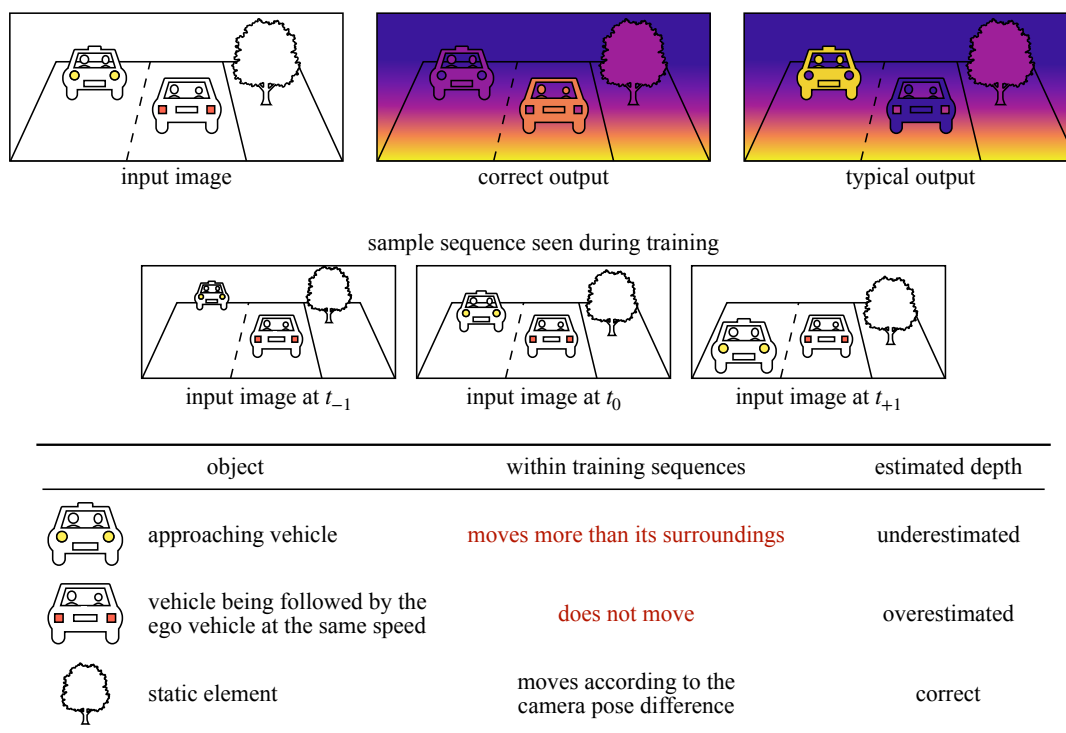


Figure 5.4. Representation of the issues of self-supervised monocular depth models with dynamic objects. The distance of leading traffic tends to be overestimated since it does not change position across training sequences, such as the far background when driving at the same speed. Instead, the depth of incoming traffic is often underestimated because such vehicles move significantly more than their surroundings across training sequences. These problems are due to the assumption of the moving camera in a static world, which is violated systematically by any moving object in the scene, e.g., traffic.

paper [83], the auto-mask loss does not work for incoming traffic since the pixels corresponding to such vehicles are highly dynamic, so they do not get filtered out.

To overcome this issue, we introduce weak supervision from the radar signal and, optionally, radar as an extra input. This radar loss is aimed at correcting the wrong depth estimates on dynamic objects. However, since radar signals are highly sparse, noisy, and often lack elevation information, we must prepare them before using them as supervision. Towards this end, we mitigate these inherent radar issues by expanding its signal over the image and reducing its noise. Furthermore, to successfully use the distance provided by the radar, we need to address another issue of self-supervised monocular depth estimation: the lack of scale. We tackle this problem with the effective weak velocity supervision introduced by Guizilini et al. [101].

Remarkably, our techniques substantially improve the depth estimation of dynamic objects, such as that of cars by 37%, and static elements too. The added robustness introduced by the radar is also beneficial in challenging illumination and weather conditions, such as rain and night, which we explored further in a later work (Section 5.3.2).

The following includes the main paper, while the supplementary material can be found in Section B.3 and the paper’s license in Appendix C.

R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes

Stefano Gasperini^{*,1,2} Patrick Koch^{*,1,2} Vinzenz Dallabetta²
Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google
* equal contribution

IEEE International Conference on 3D Vision (3DV), 2021

Summary. In this paper, we explore the use of radar to address the reliability issues of self-supervised monocular depth estimators with dynamic objects. Toward this end, we present R4Dyn. In particular, we show how radar can be used during training as a weak supervision signal, as well as an extra input to enhance the estimation robustness at inference time. Since automotive radars are readily available, this allows to collect training data from a variety of existing vehicles. Moreover, by filtering and expanding the signal to make it compatible with learning-based approaches, we address radar inherent issues, such as noise and sparsity. With R4Dyn, we can overcome a major limitation of self-supervised depth estimation, i.e., the prediction of traffic participants.

Contributions. I supervised the project and its implementation, carried out by Patrick for his Master's thesis. The team at BMW wanted to use radar to aid supervised depth estimation, similar to the work of Lin et al. [174]. I proposed to turn the project around to address the problem of dynamic objects for self-supervised models, including the core idea of fixing their depth estimates with radar measurements. Patrick picked up on this and proposed to make it based on a heatmap, leading to our weak radar loss. Federico proposed to improve it with bilateral filtering. Patrick and I found ways to mitigate the radar sparsity and noise. I proposed and guided most of the experiments, including those to validate the claims, which Patrick executed. I drove the analysis and discussions, wrote the paper, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: <https://doi.org/10.1109/3DV53792.2021.00084>.

Copyright Statement. ©2021 IEEE. Reprinted, with permission, from Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari, 'R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes', IEEE International Conference on 3D Vision (3DV), December 2021, London, United Kingdom.

The supplementary material can be found in Section B.3 and the license in Appendix C.

R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes

Stefano Gasperini^{*,o,1,2}
Nassir Navab^{1,3}

Patrick Koch^{*,1,2}
Benjamin Busam¹

Vinzenz Dallabetta²
Federico Tombari^{1,4}

¹ Technical University of Munich

² BMW Group

³ Johns Hopkins University

⁴ Google

Abstract

While self-supervised monocular depth estimation in driving scenarios has achieved comparable performance to supervised approaches, violations of the static world assumption can still lead to erroneous depth predictions of traffic participants, posing a potential safety issue. In this paper, we present R4Dyn, a novel set of techniques to use cost-efficient radar data on top of a self-supervised depth estimation framework. In particular, we show how radar can be used during training as weak supervision signal, as well as an extra input to enhance the estimation robustness at inference time. Since automotive radars are readily available, this allows to collect training data from a variety of existing vehicles. Moreover, by filtering and expanding the signal to make it compatible with learning-based approaches, we address radar inherent issues, such as noise and sparsity. With R4Dyn we are able to overcome a major limitation of self-supervised depth estimation, i.e. the prediction of traffic participants. We substantially improve the estimation on dynamic objects, such as cars by 37% on the challenging nuScenes dataset, hence demonstrating that radar is a valuable additional sensor for monocular depth estimation in autonomous vehicles.

1. Introduction

Depth estimation is a fundamental task for scene understanding in autonomous driving and robotics navigation. While learning-based supervised approaches for monocular depth have achieved strong performance in outdoor scenarios [7, 2], the expensive LiDAR sensors required for supervision are not readily available. Additionally, collecting such ground truth data is challenging, and requires further processing, as the raw LiDAR signal may not be sufficient.

* The authors contributed equally.

^o Contact author: Stefano Gasperini (stefano.gasperini@tum.de).

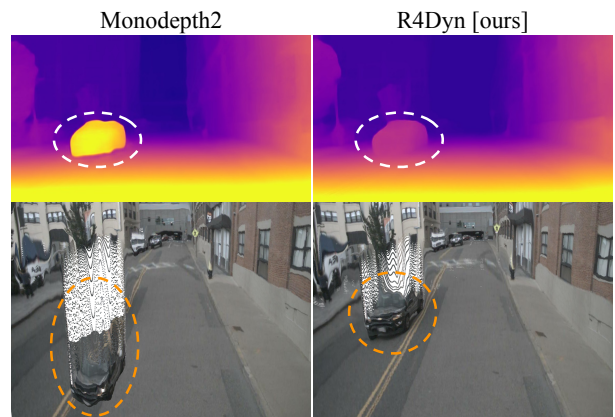


Figure 1. Example of depth prediction of our R4Dyn compared to that of Monodepth2 [11], from the validation set of nuScenes [3]. This dynamic scene violates the static world assumption, posing a challenge for self-supervised approaches. The depth of the safety critical oncoming traffic is severely underestimated by Monodepth2, but correctly predicted by our R4Dyn.

Alternative methods exploit geometrical constraints on stereo pairs or monocular videos to learn depth in a self-supervised fashion. Image sequences offer the most inexpensive source of supervision for this task. However, these approaches require to estimate the camera pose between frames at training time [11], and suffer from inherent issues, such as scale ambiguity and the tendency to incorrectly estimate the depth of dynamic objects.

Furthermore, self-supervised methods rely on the assumption of a moving camera in a static world [30]. As real-world street scenes are typically dynamic, this assumption is often violated, leading to significantly wrong predictions. This raises a variety of issues, such as the infinite depth problem, where leading vehicles driving at the same speed as the camera are predicted to be infinitely far away (such as the horizon), due to their lack of relative motion across frames. Several works addressed this problem either by excluding from the loss computation those regions

where no pixel variation is detected [11], or discarding from the training set those frames with leading vehicles driving at a similar speed [13]. Others increase the complexity by estimating the individual object’s motion [4] or the scene flow [20]. However, existing methods that preserve the model complexity [11, 13] fail to cope with oncoming traffic, as can be seen in Figure 1. In this case, instead of infinitely far, the estimated depth tends to be greatly underestimated, since from the camera perspective its relative motion across frames is larger than that of static objects.

Compared to LiDARs, radars are relatively inexpensive range sensors, already integrated in a large number of mass production vehicles [21, 23], to aid features such as adaptive cruise control. Nevertheless, radar popularity in the learning-based autonomous driving domain is yet limited, and has been explored mainly in the context of object detection [26, 17]. To this date and to the best of our knowledge, only two works [21, 23] investigated the use of radar to improve depth estimation, both proposing a multi-network pipeline to incorporate radar data at inference time, while using LiDAR-supervision during training, transforming the task into sparse depth completion.

In this paper, we aim to bridge this gap and integrate readily available radar sensors to improve self-supervised monocular depth estimation. As radars are already very common, our proposal allows to collect training data from a wide range of existing vehicles, instead of a few prototypes. Despite their inherent noise and sparsity, radars could provide enough information to mitigate the limitations of self-supervised approaches, overcoming the need for LiDARs. Towards this end, we propose a novel loss formulation to complement self-supervised approaches, showing the benefits of radar as additional weak supervision signal to improve on dynamic objects. Moreover, we optionally integrate radar data at inference time, transforming the task into very sparse depth completion. We name our method R4Dyn (Radar for Dynamic scenes), and the contributions of this paper can be summarized as follows:

- We use radar to aid the prediction of dynamic objects in self-supervised monocular depth estimation.
- To the best of our knowledge, this is the first monocular depth estimation work that exploits radar as supervision signal.
- We propose a technique to filter and expand radar detections, and make radar compatible with learning-based solutions.
- We provide extensive evaluations, including errors on safety critical dynamic objects, on the challenging nuScenes dataset [3], training various prior methods under equivalent settings, hence creating a new benchmark, and easing the comparisons for future works.

2. Related Work

2.1. Supervised Monocular Depth Estimation

Estimating depth from a single color image is an ill-posed problem, as there is an infinite number of 3D scenes that can yield the same 2D projection. Nonetheless, tremendous advances have been achieved since Eigen et al. [6] pioneered using CNN-based architectures and Laina et al. [19] leveraged fully-convolutional networks with residual connections [14] to predict dense depth maps from monocular images. While most supervised works regressed directly to the depth measurements of LiDAR sensors (as in KITTI [9]) or RGB-D cameras (as in NYU-Depth v2 [28]), Fu et al. [7] formulated the task in an ordinal fashion.

2.1.1 Depth Completion with Radar

Despite a recently increasing interest in radar for object detection [26, 17], to this date, only two works [21, 23] used it for depth estimation. Both achieved substantial improvements by using it in a LiDAR-supervised setting with a multi-stage architecture, with the first stage filtering the radar signal and improving its quality. In particular, by incorporating the radar as additional input, they transformed the depth estimation task into highly sparse depth completion. Lin et al. [21] were the first to use radar in this supervised context, where they followed a late-fusion approach to account for the heterogeneity of input modalities. Long et al. [23] proposed a sophisticated learning-based association between the projected radar points and the RGB image.

Similarly to these pioneering works [21, 23], our method also incorporates radar for depth estimation, but we follow a novel idea: we focus on using radar to improve the estimation of dynamic objects in a self-supervised setting, for which we propose a specific loss function.

2.2. Self-Supervised Monocular Depth Estimation

Self-supervised methods overcome the need for expensive LiDAR data by leveraging view reconstruction constraints, either via stereo pairs [8, 10] or monocular videos [34, 11, 12]. The latter build on the motion parallax induced by a moving camera in a static world [30]. Furthermore, these methods require to predict simultaneously the depth and the camera pose transformation. Since the pioneering work on video-based training by Zhou et al. [34], vast improvements have been achieved thanks to novel loss terms [11], detail-preserving network architectures [12] and the exploitation of cross-task dependencies [16, 13].

2.2.1 Solutions to Self-supervised Inherent Issues

Scale ambiguity Since infinitely many 3D objects correspond to the same 2D projection, video-based methods

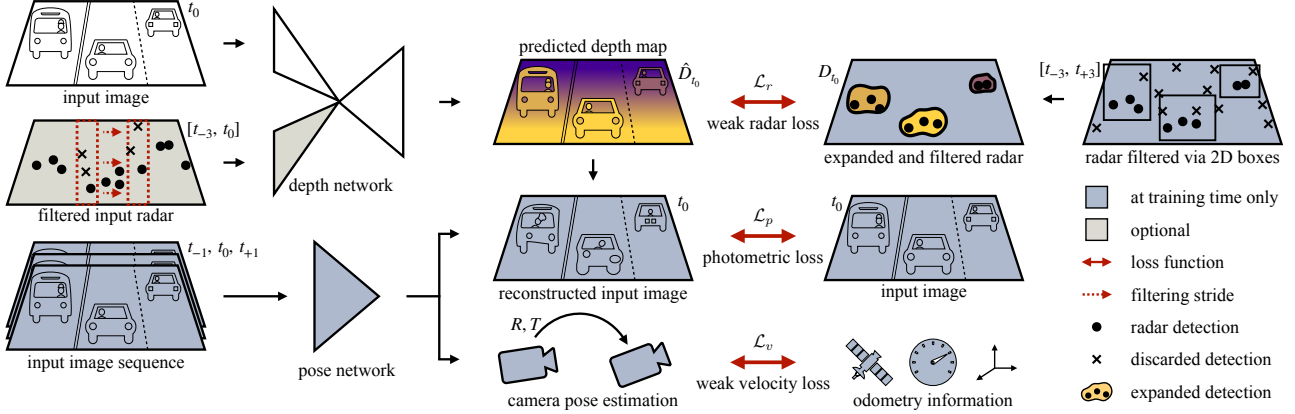


Figure 2. Overview of R4Dyn. The proposed approach incorporates radar as a weak supervision signal, and optionally as additional input. R4Dyn extends a self-supervised framework by incorporating radar to improve on the depth estimation of dynamic objects.

can only predict depth up to an unknown scale factor. Therefore, a plethora of works [34, 11, 4] rely on ground truth (i.e. LiDAR) median-scaling at test time. Guizilini et al. [12] targeted this issue by imposing a weak velocity supervision on the estimated pose transformation, exploiting the available odometry information, achieving scale-awareness.

Dynamic scenes Another major limitation of video-based approaches is due to the inherent static world assumption. This is perpetually violated in driving scenarios, leading to critically incorrect depth predictions of dynamic objects (e.g. traffic participants). A typical failure case is caused by leading vehicles driving at a similar speed as the ego vehicle, thereby lacking relative motion across frames and resulting in a significantly overestimated depth. Godard et al. [11] addressed this problem with an auto-masking loss to ignore pixels without relative motion. In contrast, Guizilini et al. [13] proposed a workaround to detect and discard training samples where this “infinite depth problem” occurs, thereby keeping only uncomplicated frames, albeit reducing the training data.

However, neither of the two [11, 13] accounted for oncoming traffic, which leads to a significantly underestimated depth, since its motion across frames is larger than that of the static elements. This might be linked to the popular KITTI depth benchmark [9] mostly lacking such kind of safety critical dynamic scenes, widely available on nuScenes [3]. Alternative solutions target dynamic scenes, by increasing the model complexity and simultaneously predicting depth, ego-pose, plus 3D motion of dynamic objects [4] or scene flow [33, 24, 15, 20].

Although methods learning scene flow in a self-supervised fashion do not require expensive additional labels, e.g. instance segmentation [4], they might suffer from the same ambiguities of self-supervised depth estimation, and require stereo vision [33, 24, 15].

Our proposed approach is substantially different from previous works targeting dynamic scenes. We address this problem by incorporating radar data, which has not been explored before in this context.

3. Method

In this paper we incorporate radar data to enhance the self-supervised depth prediction of dynamic scenes. An overview of our method can be seen in Figure 2. We build on top of a self-supervised framework which learns from monocular videos, and uses the vehicle odometry (Section 3.1), as in [12], for scale-awareness. Although radar signals are highly sparse and noisy, they could provide enough information to improve self-supervised methods. So, we integrate radar data both during training to improve the prediction of dynamic objects (Section 3.2.3), and inference to increase the overall robustness (Section 3.3).

Radar inherent issues Radar and LiDAR provide significantly different signals: radar signals are noisy, sparse and often lack elevation information. Radar detections projected to the image plane occupy a single pixel, leading to a low density of $\approx 0.03\%$, compared to $\approx 1.63\%$ of a 32-beam LiDAR [3]. Additionally, radar suffers from several sources of noise (clutter), complicating its usage in learning-based approaches. The major causes are multi-path [23] and “see-through” effects, due to the different viewpoints [27], scene geometry and physical sensor properties, and not relatively simple Gaussian noise. Multi-path effects alone affect about 35% of all points [23]. Moreover, due to missing elevation information (e.g. in nuScenes [3]), all detections lie in a plane parallel to the ground. This complicates radar usability, requiring new techniques to make it compatible with learning-based approaches and correctly associate radar detections with image pixels. Therefore, we tackle these issues by expanding the radar influence (Section 3.2.1), and mitigating its noise (Section 3.2.2).

3.1. Self-Supervised Framework

The proposed method is built on top of a video-based monocular depth approach, described in this Section. We aim to simultaneously predict the depth $\hat{D}_t$ of a target frame and the pose transformations $T_{t \rightarrow s}$ between target I_t and source frames $I_s \in \{t-1, t+1\}$. Depth and pose estimates serve to warp the source frames into a reconstructed target view, from which an appearance-based error is computed from the view reconstruction [34] and a Structural Similarity (SSIM) [32] term, as in [11, 12].

Following [11], only the minimum reprojection error $\mathcal{L}_p$ is considered, accounting for partial occlusions. Moreover, pixels without relative motion across frames are masked out [11]. An additional loss $\mathcal{L}_s$ encourages smoothness whilst preserving edges [10]. $\mathcal{L}_p$ and $\mathcal{L}_s$ are computed at each scale of the depth decoder, after upsampling to full resolution [11].

As the radar provides absolute depth values, to use its signal as weak supervision (Section 3.2.3), it is crucial to estimate depth at the right scale. However, as the learning objectives above only allow to predict depth up to an unknown scale factor. We follow [12] to achieve scale-awareness with a weak velocity supervision $\mathcal{L}_v$ on the pose transformation.

3.2. Weak Radar Supervision

In this work, we focus on dynamic objects, which embody safety critical failure cases of self-supervised monocular depth estimation. We aim to mitigate this issue, which is visible in Figure 1, via a weak radar-based supervision.

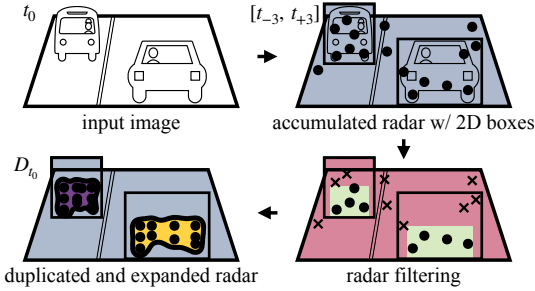


Figure 3. Radar preparation for weak supervision, detail of the top right portion of Figure 2. The signal is accumulated, then filtered via 2D boxes, duplicated and expanded.

3.2.1 Addressing Radar Sparsity

To address radar high sparsity and make it suitable as supervision signal, it is fundamental to expand its influence over a larger portion of the image. In particular, as we focus on dynamic objects, we aim to expand the radar signal across all object pixels, within their boundaries. At training time, we overcome sparsity by incorporating the RGB image context information, similarly to [29, 1], as well as 2D bounding boxes.

Context-aware radar expansion We expand each radar point to cover a larger area of the corresponding object. Expanding it as much as possible, but constraining it within the boundaries, and accounting for 3D shape variations require a precise mapping between the projected radar points and the object pixels. Towards this end, during training, we generate an association map around each radar point. We exploit the idea of bilateral filtering [29], a common edge-preserving image smoothing technique. In particular, we link an image pixel p to a radar point r if p and r are spatially close in the image space, within the same bounding box, and have similar pixel intensities (i.e. $I(p) \approx I(r)$). This bilateral association map can be formalized as follows:

$$w(p, r) = \exp \left(-\frac{(\Delta u)^2 + (\Delta v)^2}{2\sigma_d^2} - \frac{\|\Delta I\|_2^2}{2\sigma_r^2} \right) \quad (1)$$

where $\Delta u = (u - u_r)$ and $\Delta v = (v - v_r)$ are the pixel distances, $\Delta I = I(u, v) - I(u_r, v_r)$ the difference of intensities of a radar point pixel $r = (u_r, v_r)$ and a neighboring pixel $p = (u, v)$, while σ_d and σ_r denote domain and range smoothing parameters, respectively. Hence, the bilateral association $w(p, r)$ represents the probability of a pixel p to be linked to a radar point r .

Pixel-radar association map Although this bilateral confidence map preserves edges by design, it may still leak small, nonzero values beyond object boundaries, leading to undesirably smooth depth predictions. We avoid this by clipping the heatmaps in proximity of the box edges. Moreover, $w(p, r)$ could serve as per-pixel weight for the radar supervision. However, this would put more emphasis on the radar pixel r and result in uneven object depth maps. We address this by transforming the confidence values $w(p, r)$ in a binary map. Considering the set of reliable detections R_{df} (Section 3.2.2) and a threshold $\gamma \in \mathbb{R}$, we compute:

$$W(p, r) = \begin{cases} 1 & \text{if } \max_{r \in R_{df}} w(p, r) > \gamma \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

so all relevant portions get the same amount of supervision.

Duplication To account for the missing elevation, as in Figure 3 we copy each projected radar point r and its corresponding depth along the vertical axis to the lower third and middle of its bounding box. We leave out the upper half of the box to consider depth variations within the boxes (e.g. windshield of the car in the lower part of Figure 4).

Measurement Accumulation We also reduce sparsity by accumulating measurements. To do so, we exploit radar doppler information to compensate for ego- and target-motion across samples, as proposed by [17]. Since the doppler velocity is radial, it provides only a rough estimate of the true velocity of the target, but it is a reasonable approximation considering the frame rate. Via duplication and accumulation, we obtain a dense set of radar points R_d .

3.2.2 Addressing Radar Noise

Noise is another major radar issue to be reduced (Section 3).

Clutter removal We want to extract from R_d reliable measurements R_{df} for supervision. We do so by leveraging 2D bounding boxes during training to filter out noisy radar detections. Within each object in the image space, radar detections closer to the sensor are likely to be reliable, whereas points at higher distances often result from noise. Hence, to obtain R_{df} , we find the minimum depth d_m per bounding box b_i and only keep detections within b_i having $d < d_m + \beta$. The tolerance β allows to keep multiple points in b_i , and accounts for depth variations along 3D objects.

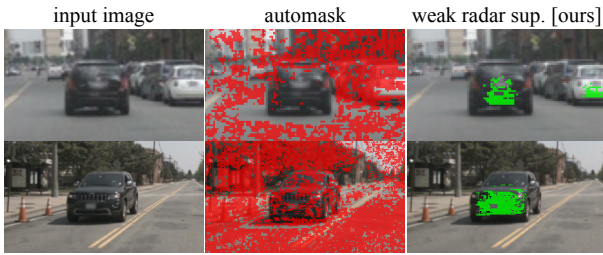


Figure 4. The automask [11] and our radar supervision mask. Red pixels contribute to the loss $\mathcal{L}_p$. The automask correctly masks out the leading car (top), but not the oncoming one (bottom). Our radar supervision $\mathcal{L}_r$ acts (green) successfully on both cars.

Object-focused filtering Radars without elevation information (e.g. nuScenes [4]) provide only detections parallel to the ground plane, thus, in the image, further points appear higher than closer ones, and within the same objects are more likely to be noisy. Analogously, detections around the box edges could be unreliable due to wide boxes or “see-through” effects. For these reasons, as shown in Figure 3, we discard points in the upper 50% and outer 20% of bounding boxes, as well as overlapping areas. Furthermore, it is more intricate to assess radar detections reliability outside object boxes, hence we discard all background radar points to avoid erroneous supervision.

3.2.3 Training Objective for Dynamic Objects

Addressed noise and sparsity, the radar signal is suitable to weakly supervise the depth prediction of dynamic objects. We want to use it to adjust erroneous depth estimations via a loss function. We define this weak radar supervision as:

$$\mathcal{L}_r(\hat{D}, D) = \frac{1}{N} \sum_{p \in I} \left\| \hat{D}_p - D_p \right\|_1 \odot W(p, r) \quad (3)$$

where $\odot$ denotes the Hadamard product. Eq. 3 aims at fixing the predicted depth $\hat{D}$ towards the expanded target radar measurements D , for each of the N image pixels p where the binary association map $W(p, r)$ (Eq. 2) is positive.

Depth gradient preservation The expanded measurements D in Eq. 3 act as ground truth depth. We introduce $\Omega_j = [W(p, r_j) = 1]$ as the area affected by the expanded radar point $r_j \in R_{df}$. If D was constant within each Ω_j , then $\mathcal{L}_r$ would shift the whole area to the same depth D . This would not take into account depth variations within objects, such as the door panels or the windshield of the lower car in Figure 4 being further than its front bumper. Moreover, although the depth of dynamic objects is often under- or over-estimated by video-based self-supervised methods, depth variations within objects are typically well predicted (e.g. in Figure 1). For these reasons, we adapt the pseudo-ground truth D to preserve depth variations. As shown in Figure 5, we do so by computing $\Delta_j = D_{radar}(r_j) - \hat{D}(r_j)$ as the difference between the prediction $\hat{D}(r_j)$ and the measurement $D_{radar}(r_j)$, only at each radar pixel $r_j \in R_{df}$. We then generate the pseudo ground truth $D(\Omega_j)$ by shifting the prediction $\hat{D}(\Omega_j)$ by the same Δ_j :

$$D_j(p) = \hat{D}_j(p) + \Delta_j \quad (4)$$

where Δ_j is computed at the radar pixel r_j such that $\arg \max_{r \in R_{df}} w(p, r_j)$.

The final objective function can thus be formulated as:

$$\mathcal{L} = \mathcal{L}_p + \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_v + \lambda_3 \mathcal{L}_r \quad (5)$$

with λ_1 , λ_2 and λ_3 being balancing coefficients.

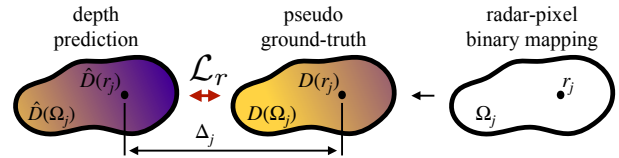


Figure 5. Depth gradient preserving mechanism. We compute Δ_j between prediction and radar detection, then adjust each pixel in the prediction by Δ_j , by enforcing a pseudo-ground truth during training, thereby preserving the predicted gradient.

3.3. Sparse Depth Completion with Radar as Input

As radar sensors are readily available in a wide range of mass-production vehicles [21, 23], a depth estimator could exploit them at inference time, transforming the estimation task in a very sparse depth completion problem. To do so, we first need to mitigate radar inherent sparsity and noise issues. We follow a similar approach as for using the signal as weak supervision (Section 3.2), although we do not make use of 2D bounding boxes at inference time, since extracting them for the input would increase the runtime.

For sparsity, we apply the same measurement accumulation technique described in Section 3.2.1 to obtain a denser point cloud, aggregating the current and past radar frames. For the noise, we adopt the same min-pooling approach as

Method	Sup.	Input	AbsRel	SqRel	RMSE	δ_1	AbsRel _C	AbsRel _V	AbsRel _N	AbsRel _P
Lin et al. [‡] [21]	GT	ImR	0.1086	1.080	5.394	88.21	0.1907	0.2082	0.2088	0.2930
Struct2Depth ^K [4]	Mi*	Im	0.2195	3.799	8.441	73.23	0.3323	0.3516	0.3739	0.2993
PackNet-SfM [12]	Mv	Im	0.1567	2.440	7.230	82.64	0.1814	0.2382	0.2508	0.2473
Monodepth2 [‡] [11]	M*	Im	0.1398	1.911	6.825	84.82	0.1983	0.2110	0.2300	0.2572
baseline [‡] [ours]	Mv	Im	0.1315	1.705	6.520	85.71	0.1862	0.2091	0.2254	0.2351
R4Dyn-L [‡] [ours]	Mvr	Im	0.1296	1.658	6.536	85.76	0.1343	0.1618	0.1686	0.2231
R4Dyn-LI [‡] [ours]	Mvr	ImR	0.1259	1.661	6.434	86.97	0.1250	0.1504	0.1589	0.2146

Table 1. Evaluation on the nuScenes [3] validation *day-clear* set. [‡] were pretrained on ImageNet [5], ^K also on KITTI [9]. Supervisions (Sup.): GT: via LiDAR data, M: via monocular sequences, *: test-time median-scaling via LiDAR, i: instance masks, v: weak velocity, r: weak radar. Inputs (In.): Im: RGB, ImR: RGB and radar. C, V, N and P for *Cars*, *Vehicles*, *Non-parked vehicles* and *Pedestrians*, respectively. R4Dyn-L and R4Dyn-LI: proposed method with radar as L: weak supervision, I: input. Notation reused in other Tables.

in Section 3.2.2, but instead of exploiting 2D boxes, we slide a fixed-size window across all projected radar points.

To account for the fact that radar and camera are heterogeneous sensors, we apply separate encodings and merge the features in a late fusion fashion, as in [21].

4. Experiments and Results

4.1. Experimental Setup

Dataset We conduct all experiments on the challenging nuScenes dataset [3]. We selected it as it is the only available large-scale public dataset with the recording vehicle fitted with a camera and an automotive radar. NuScenes contains around 15h of driving data collected in Boston and Singapore, including diverse traffic scenarios (e.g. both left and right hand drive), with a plethora of dynamic scenes (unlike KITTI [9]), making it difficult for self-supervised depth estimation. As we are interested in sensor setups readily available in production cars, we consider only data from front-facing camera and radar. For training, we use only scenes with good visibility (i.e. *day-clear*). This includes 15129 samples (with synced image, radar and LiDAR data) for the official training set and 6019 for the official validation set (of which 4449 are *day-clear*).

Evaluation metrics We evaluated our models on the standard depth estimation metrics and errors. As ground truth we used single raw LiDAR scans up to a maximum depth of 80m. In particular, as this work focuses on dynamic objects, we are interested in the performance improvements on such objects. Towards this end, we also evaluated according to the semantic class: we exploited LiDAR semantic segmentation annotations from [3] to distinguish between classes, and computed the errors on the depth predictions at the corresponding LiDAR points. The evaluated classes comprise *Cars*, *Vehicles* (e.g. *Cars*, *Trucks*, *Buses*, *Motorcycles*, *Bicycles*), *Non-parked Vehicles* and *Pedestrians*, thereby encompassing all traffic participants.

Network architecture The presented techniques for

processing and incorporating radar data for depth are not tailored to a specific backbone architecture. We used the small and effective ResNet-18-based [14] pose and depth networks from [11]. The pose network comprised 13.0M parameters, while the depth network 14.8M. With the radar as input, an additional ResNet-18 encoder branch was used, increasing the depth parameters by 1.1M.

ID	Weak radar sup.	AbsRel	AbsRel _C
L1	baseline	0.1315	0.1862
L2	L1 + $\mathcal{L}_r$ w/ raw pts	0.1766	0.4841
L3	L2 + filter pts w/ GT box	0.1323	0.1960
L4	L3 + bilateral expansion	0.1306	0.1551
L5	L4 + binary mapping	0.1297	0.1356
L6	L5 + depth gradient pres.	0.1296	0.1343
L7	L6 – GT + predicted box	0.1289	0.1343

Table 2. Weak radar supervision ablation, evaluated on the nuScenes [3] *day-clear* validation set. The baseline L1 is defined in Section 3.1. L3-L6 and L7 make use of 2D boxes from ground truth annotations or an off-the-shelf detector [31], respectively. L6 is equivalent to R4Dyn-L, with image-only input (Im).

Implementation details Each training sample consisted of a 576x320 resolution image triplet (t_{-1}, t_0, t_{+1}), 4 radar measurements at $[t_{-3}, t_0]$ for the input, and 7 radar samples for the supervision at training time at $[t_{-3}, t_{+3}]$. At inference time we used a single image at t_0 , and 4 radar measurements at $[t_{-3}, t_0]$. Using the Adam optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, we trained with a batch size of 16 for a total of 40 epochs with initial learning rate 2×10^{-4} , which was halved every 10 epochs. If enabled, we introduced the radar supervision after 30 epochs, when the correct depth scale had already been learned, and set the learning rate to 1×10^{-5} , halved after 8 epochs. The loss balancing weights were set to $\lambda_1 = 1 \times 10^{-3}$, $\lambda_2 = 0.02$, $\lambda_3 = 0.2$. During training, we applied random horizontal

flipping to the input data as well color jittering with brightness, contrast, saturation ± 0.2 and hue ± 0.05 to the input images. The window for filtering radar at inference time was 320 pixels tall and 8 wide, with stride 3, while the tolerance β was $2m$. Further details can be found in the supplementary material. In the following, we refer to different configurations of our method, namely R4Dyn-L, R4Dyn-I, and R4Dyn-LI, with -L and -I denoting our weak radar supervision and radar as input, respectively. We trained all models using PyTorch on a single NVIDIA Tesla V100 32GB GPU. Inference of our full method (i.e. R4Dyn-LI) takes 27ms on an NVIDIA GTX 1080 8GB GPU.

ID	Input config.	AbsRel	AbsRel _C
I1	RGB only	0.1315	0.1862
I2	I1 + 1 radar input	0.1357	0.1926
I3	I2 + $\mathcal{L}_r$	0.1298	0.1319
I4	I3 + radar accum.	0.1301	0.1279
I5	I4 + doppler accum.	0.1273	0.1264
I6	I5 + filtered	0.1259	0.1250

Table 3. Pre-processing ablation for radar as input, evaluated on the *day-clear* nuScenes [3] validation set. I1 is defined in Section 3.1. I6 is equivalent to R4Dyn-LI, our full approach.

Prior works and baseline For a fair comparison, we re-trained all methods on the same dataset split, using the same image resolution, the official implementations and parameters, until convergence. For Struct2Depth [4] we started from model weights pretrained on ImageNet [5] and then KITTI [9] by the authors, which improved its convergence over ImageNet-pretraining only. All methods except for PackNet-SfM [12] used a ResNet-18 [14] backbone pretrained on ImageNet [5]. Our baseline is the self-supervised image-only method described in Section 3.1, which includes the weak velocity supervision.

Radar %	AbsRel	AbsRel _C	AbsRel _V	AbsRel _P
0 %	0.1315	0.1862	0.2091	0.2351
25 %	0.1309	0.1358	0.1672	0.2254
50 %	0.1261	0.1309	0.1553	0.2167
100 %	0.1259	0.1250	0.1504	0.2146

Table 4. Evaluation on the nuScenes [3] *day-clear* validation set. Different amounts of radar points are shown, both for input and weak supervision. 0% is the baseline, while 100% is R4Dyn-LI. C, V and P stand for *Cars*, *Vehicles* and *Pedestrians* respectively.

4.2. Quantitative Results

Comparison with related methods Table 1 shows the comparison between our R4Dyn and related approaches.

We report alternative solutions for dynamic objects, such as Struct2Depth [4] and Monodepth2 [11], as well as another method using radar for depth, i.e. the LiDAR-supervised work by Lin et al. [21]. Our R4Dyn-L and R4Dyn-LI outperformed the strong baseline across the board, by a significant margin. Remarkably, the error on *Cars* dropped by a substantial 33% with R4Dyn-LI, showing the benefit of radar for estimating the depth of dynamic objects. Our approach largely improved over Monodepth2 [11], on which our baseline builds upon. As can also be seen in Figure 4, the automask from Monodepth2 [11] was not able to cope with oncoming traffic, occurring frequently in the nuScenes dataset. This led to a large error on *Cars* and other traffic participants. Our R4Dyn-L and R4Dyn-LI reduced it by 32% and 37% respectively.

Method	<i>all</i>	<i>clear</i>	<i>rain</i>	<i>night</i>
Lin et al. [21]	0.126	0.109	0.145	0.248
Struct2Depth [4]	0.238	0.220	0.271	0.336
PackNet-SfM [12]	0.168	0.157	0.177	0.262
Monodepth2 [11]	0.161	0.140	0.193	0.287
baseline [ours]	0.146	0.132	0.166	0.242
R4Dyn-L [ours]	0.147	0.130	0.164	0.273
R4Dyn-LI [ours]	0.137	0.126	0.146	0.219

Table 5. Evaluation of AbsRel to generalize on adverse and unseen conditions of the validation set of nuScenes [3]. All methods were trained on *day-clear* (*clear*).

Struct2Depth [4], despite being additionally pretrained on KITTI [9] and its individual motion predictions, produced the worst estimations, with large inconsistencies (AbsRel std. 0.1511). Our R4Dyn-LI outputs were far more consistent (std. 0.079). We attribute this difference to the superiority of radar over instance masks as weak supervision, as well as to Struct2Depth [4] not fully solving the infinite depth problem (e.g. in Figure 6). The sophisticated PackNet architecture [12] was not able to deliver satisfactory results, which could be due to the larger model size (129M instead of 15M for ResNet-18), and the relatively small dataset. This motivated using ResNet [14] as backbone. Furthermore, the LiDAR-supervised work by Lin et al. [21], with radar as input, performed better than ours overall, albeit worse on safety critical traffic participants, such as by 53% on *Cars*. This could be due to the sparsity of the LiDAR from which it learned. Overall, Table 1 demonstrates the benefit of radar for monocular depth estimation, as it can substantially improve the predictions of safety critical dynamic objects, both as weak supervision with R4Dyn-L and as input with R4Dyn-LI.

Weak Radar supervision ablation study Table 2 shows the impact of the various components of our weak radar supervision. Throughout the table, errors do not decrease with

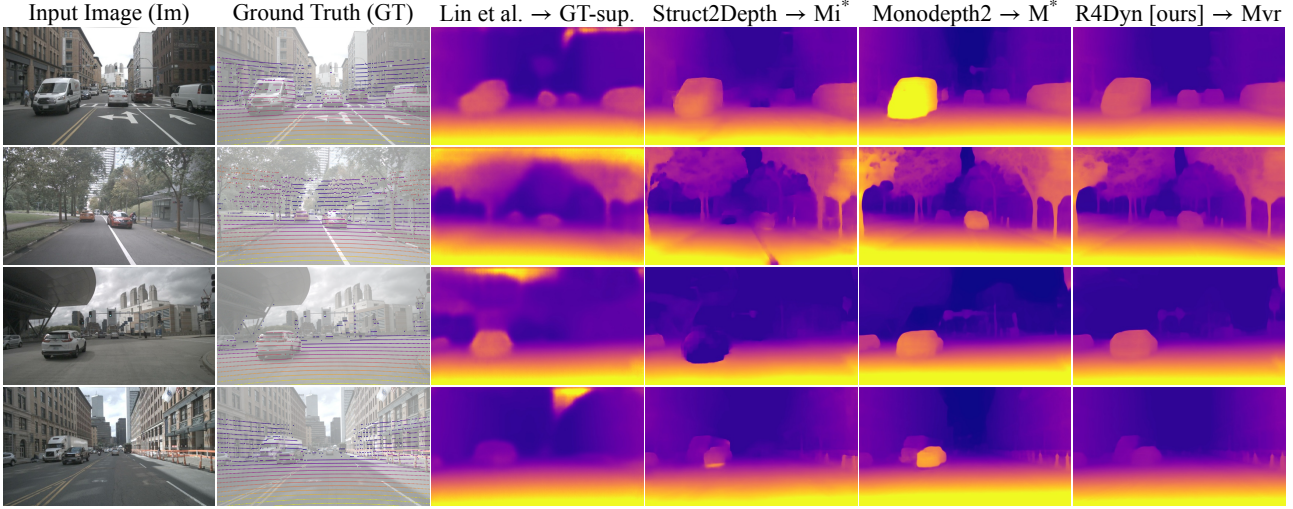


Figure 6. Qualitative results from related works on the nuScenes [3] validation set. Next to each method name we indicate its supervision.

the introduction of each and every feature, thereby confirming that the radar signal necessitates to be filtered (row L3) and expanded (L4) before it can positively contribute over the baseline (L1), while simply integrating the raw points (L2) increased the errors. With L7, we show that 2D bounding boxes can be extracted via an off-the-shelf detector, such as Scaled-YOLOv4 [31] trained on MS COCO [22], removing the need for ground truth annotations. L7 improved over L6 (which used ground truth boxes), presumably due to the nuScenes [3] boxes being oversized (used in L3 to L6). Overall, Table 2 confirms the importance of our modifications to use radar as weak supervision.

Radar as input ablation study Table 3 reports various configurations for using radar as input, again motivating the expansion (I4 and I5) and filtering (I6) of the radar signal. In fact, simply adding a single radar sweep in input (I2) did not outperform the baseline (I1).

Variable amount of radar signal In Table 4 we show how the output quality changed by excluding a variable amount of radar detections from the dataset. Already 25% of the radar detections brought a large improvement over the baseline (0%), overall as well as on safety critical traffic participants. Once again, this shows the benefit of radar for depth estimation, despite its inherent noise and sparsity. In particular, adding more detections (filtered and expanded), systematically reduced all the errors. Moreover, considering the rapid progressing of sensor technology [25], higher resolution (e.g. 200%, 400%) and less noisy automotive radar sensors might be available in the future, which would further increase the gap to the RGB-only baseline.

Generalization to unseen adverse conditions Table 5 reports AbsRel of our R4Dyn and related methods on difficult unseen conditions. This shows the ability of each to generalize to different data distributions. Our R4Dyn-LI

outperformed all self- and weakly-supervised approaches in all conditions. Compared to the LiDAR-supervised work by Lin et al. [21], R4Dyn-LI performs similarly in rain and significantly better in night scenes, reiterating the effectiveness and robustness of our techniques.

4.3. Qualitative Results

Qualitative results in Figure 6 confirm the findings of our experiments, showing the superiority of our R4Dyn in estimating the depth of traffic participants. In particular, Struct2Depth [4] had frequent issues with leading vehicles (first 3 scenes), and added a halo effect to the oncoming car in the fourth scene. Monodepth2 [11] was able to correctly estimate leading vehicles, thanks to its automask, but not oncoming traffic (as seen in Figure 1 and inspected in Figure 4), which resulted in severe underestimations (first, second and fourth scene). The LiDAR-supervised work by Lin et al. [21] correctly estimated the overall depth, but missed most details, delivering blurred outputs. Instead, our R4Dyn could accurately estimate all challenging dynamic scenes, preserving sharp details.

5. Conclusion

In this paper we proposed R4Dyn, a set of techniques to integrate radar into a self-supervised monocular depth estimation framework. Extensive experiments showed the benefit of using radar both during training as weak supervision, and at inference time as added input. Our method substantially improved on the prediction of safety critical traffic participants over all related works. Therefore, R4Dyn constitutes a valuable step towards robust depth estimation. Additionally, the inexpensive and readily available setup required, allows to collect training data from a variety of existing vehicles, removing the need for expensive LiDAR data.

References

- [1] Jonathan T. Barron and Ben Poole. The fast bilateral solver. In *Proceedings of the European Conference on Computer Vision*, pages 617–632. Springer, 2016. [4](#)
- [2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. [1](#)
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. NuScenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#), [11](#), [12](#), [13](#), [14](#), [15](#)
- [4] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019. [2](#), [3](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [14](#)
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. [6](#), [7](#)
- [6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014. [2](#)
- [7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. [1](#), [2](#)
- [8] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference of Computer Vision*, pages 740–756. Springer, 2016. [2](#)
- [9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, Aug. 2013. [2](#), [3](#), [6](#), [7](#), [12](#), [15](#)
- [10] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6602–6611, 2017. [2](#), [4](#), [11](#)
- [11] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3827–3837, 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [11](#), [12](#), [13](#), [14](#), [15](#)
- [12] Vitor Guizilini, Rare Ambru, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2020. [2](#), [3](#), [4](#), [6](#), [7](#), [11](#), [12](#), [13](#), [14](#)
- [13] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *Proceedings of the International Conference on Learning Representations*, 2020. [2](#), [3](#)
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. [2](#), [6](#), [7](#)
- [15] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7394–7403, 2020. [3](#)
- [16] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson W. H. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision*, volume 11219, pages 55–71. Springer, 2018. [2](#)
- [17] Youngseok Kim, Jun Won Choi, and Dongsuk Kum. GRIF Net: Gated region of interest fusion network for robust 3D object detection from radar point cloud and monocular image. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10857–10864, 2020. [2](#), [4](#), [11](#)
- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. [6](#)
- [19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the IEEE International Conference on 3D Vision*, pages 239–248, 2016. [2](#)
- [20] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Proceedings of the Conference on Robot Learning*, 2020. [2](#), [3](#)
- [21] Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10233–10240, 2020. [2](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [14](#), [15](#)
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference of Computer Vision*, pages 740–755. Springer, 2014. [8](#)
- [23] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. [2](#), [3](#), [5](#), [13](#), [15](#)
- [24] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3D holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2624–2641, Oct. 2020. [3](#)

- [25] Enrique D. Martí, Miguel Ángel de Miguel, Fernando García, and Joshué Pérez. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019. [8](#)
- [26] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *Proceedings of the IEEE Sensor Data Fusion: Trends, Solutions, Applications*, pages 1–7, 2019. [2](#)
- [27] Adrian Lopez Rodriguez, Benjamin Busam, and Krystian Mikołajczyk. Project to Adapt: Domain adaptation for depth completion from noisy and sparse sensor data. In *Proceedings of the Asian Conference on Computer Vision*, volume 12622, pages 330–348. Springer, 2020. [3](#)
- [28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference of Computer Vision*, pages 746–760. Springer, 2012. [2](#)
- [29] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 839–846, 1998. [4](#)
- [30] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979 Jan 15. [1](#), [2](#)
- [31] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13029–13038, 2021. [6](#), [8](#)
- [32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. [4](#), [11](#)
- [33] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3D motion understanding. In *European Conference on Computer Vision Workshops*, pages 691–709. Springer, 2019. [3](#)
- [34] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6612–6619, 2017. [2](#), [3](#), [4](#)

5.3.2 Monocular Depth Estimation in Challenging Conditions (ICCV 2023)

Compared to R4Dyn [83], where we looked into the issues depth estimators have with dynamic objects (Section 5.3.1), in this work, we address another inherent issue of monocular depth estimation: that of challenging illumination and weather conditions. We achieve this with md4all, a new method that builds on the effectiveness of state-of-the-art approaches in ideal conditions. We apply the proposed md4all under both fully- and self-supervised settings with substantial improvements across the board.

As seen for R4Dyn (Section 5.3.1), depth estimation can be learned self-supervised via monocular sequences under a series of assumptions. State-of-the-art methods are based on pixel correspondences across different frames of each input sequence [94]. With reduced visibility, such as at night or in the presence of rain, these correspondences become highly unreliable, leading to unsafe depth estimates. As shown in the paper [86], this problem is so severe that it is better to train only in ideal settings (i.e., day-clear) and infer in unseen challenging conditions (e.g., at night) than train in both easy and complex scenarios.

Figure 5.5 shows that the issues at night are due to the high noise levels in the input, the lack of texture in the dark, and the reflections from vehicle lights and streetlights. These factors are detrimental as they can all vary from frame to frame, introducing errors in the pixel correspondences. Learning depth from wrong correspondences causes the depth model to become highly unreliable in such conditions, delivering close to random estimates. Since autonomous vehicles should be able to operate safely at night, this problem has been investigated by various prior works [180, 327]. However, existing methods deliver sub-optimal results, as they introduce significant trade-offs between daytime and nighttime performances, cannot operate across various conditions, or require dedicated branches for each setting.

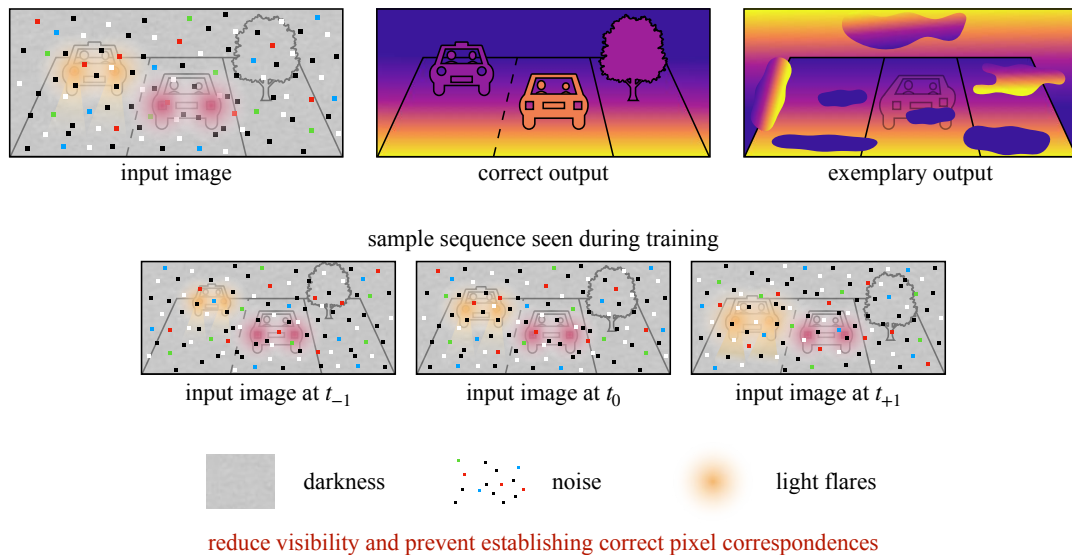


Figure 5.5. Issues of self-supervised monocular depth estimation models at nighttime. General darkness, sensor noise, and light flares caused by vehicle lights and street lamps reduce visibility and prevent establishing correct pixel correspondences across the training sequences. These issues are systematic and detrimental to the learning process, causing severe artifacts in output.

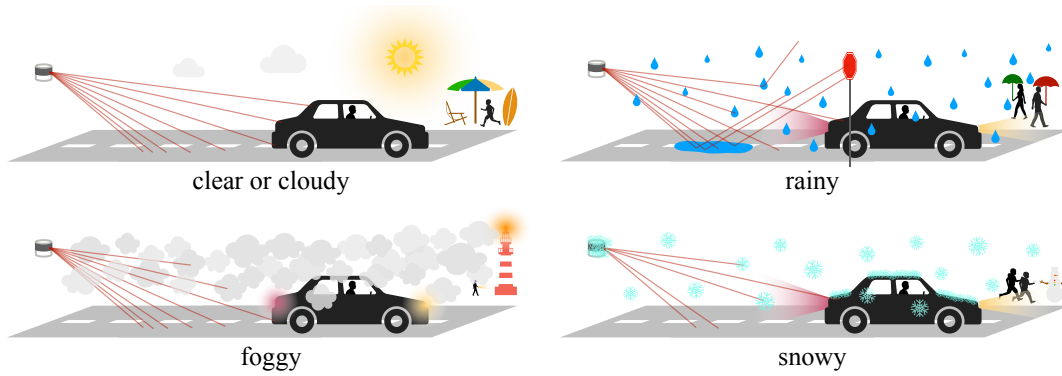


Figure 5.6. Representation of LiDAR measurements in different weather conditions. Problems occur with adverse weather since the LiDAR rays are disturbed (i.e., absorbed or scattered) by raindrops, wet surfaces, water drops inside fog, or snowflakes. Learning depth from such a signal causes the propagation of artifacts into depth estimates, such as holes in the ground in the presence of rain [86]. The absorption of the rays causes sparser point clouds that reduce the efficacy of LiDAR in these scenarios. Absorbed rays result in erroneous measurements, which are detrimental to downstream tasks. These issues do not occur at night with clear or cloudy weather. Instead, radar is not affected by adverse weather thanks to the longer wavelength of its signal, yet inherently too sparse and noisy to learn depth solely from it [83].

While this last point may seem less problematic, it becomes an issue when the environmental conditions are at the boundary between the triggering of two branches, such as at dusk.

With rain, the root causes are different, but they still lead to erroneous pixel correspondences. As shown in Figure 5.6, wet surfaces cause reflections, and raindrops introduce a vertical flow, violating the moving camera in a static world assumption [307]. Yet, no prior work has focused on rain. Only ourselves with R4Dyn [83] and a few other researchers tested the depth models with rain but did not propose anything to address it. However, as represented in Figure 5.6 and shown in the paper [86], rain is exceptionally problematic for supervised methods because the LiDAR from which they learn is highly unreliable. Wet surfaces cause it to produce artifacts in the measurements as the rays bounce and reach more distant surfaces. The problem is that supervised approaches copy such artifacts in their estimates. This behavior was not discussed in prior works despite constituting a significant issue.

Analogously, LiDAR has issues with snow and fog, too. As shown in Figure 5.6, both snowflakes and fog particles cause false detections, which are learned by supervised works. Instead, a self-supervised method would struggle with textureless areas, low visibility, and the vertical flow caused by falling snow. These issues for depth estimation were also ignored in the literature, as no method was proposed to address such conditions.

We propose a general approach to deal with such challenging conditions. We apply the method on monocular depth estimation under both self- and fully-supervised settings with the name of md4all, but the same approach can be extended to other tasks, too. The idea is simple yet effective. It is based on training on challenging samples while learning only from ideal conditions. We achieve this by translating the images with good visibility to adverse conditions, which we share open-source. Then, we train by feeding a mix of the conditions and having the training signal only from the easier samples corresponding to the problematic ones. By doing so, the model learns to extract robust features regardless of the samples given in input.

With a single monocular encoder-decoder model, the proposed strategy achieves remarkable results in all conditions, reducing depth errors by up to 60% over the approach it is applied on [94]. It even improves in standard settings, thanks to the mix of inputs acting as data augmentation and regularization. These simple techniques significantly outperform complex prior works dedicated to addressing the issues at night, achieving substantially better scores at nighttime than they do at daytime and performing better at daytime, too. While the main experiments are with sun, clouds, darkness, and rain, we include preliminary results with snow and fog in the supplementary material.

The following includes the main paper, while the supplementary material can be found in Section B.4 and the paper’s license in Appendix C.

Robust Monocular Depth Estimation under Challenging Conditions

Stefano Gasperini^{*,1,2} Nils Morbitzer^{*,1} HyunJun Jung¹ Nassir Navab¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² VisualAIs ³ Google

* equal contribution

IEEE/CVF International Conference on Computer Vision (ICCV), 2023

Summary. In this paper, we uncover reliability issues of monocular depth estimators occurring with challenging weather and illumination conditions and tackle them with md4all: a simple and effective solution that works reliably under both adverse and ideal conditions, as well as for different types of learning supervision. We achieve this by exploiting the efficacy of existing methods under perfect settings. Therefore, we provide valid training signals independently of what is in the input. First, we generate a set of complex samples corresponding to the normal training ones. Then, we train the model by guiding its self- or full-supervision by feeding the generated samples and computing the standard losses on the corresponding original images.

Contributions. I initiated this project to address the issues with night and rain seen while working on R4Dyn [83]. I supervised the project and its implementation, which Nils carried out as part of his Master's. I proposed ideas to address the problem, including crucial parts of the method (e.g., distillation from a frozen baseline model and image translation to swap the inputs). Nils picked up on these ideas, brought them forward, and made them more concrete via the implementation. I proposed to apply the method to supervised settings to tackle the artifacts of LiDAR with adverse weather for the first time. I proposed and guided the experiments, including those to validate the claims, which Nils executed for the self-supervised setting and HyunJun ran for the supervised setup and additional baselines. I drove the analysis and discussions, wrote the paper, and drew the figures.

Due to copyright restrictions, the following is the accepted but not the published version of the paper.

Published version: https://openaccess.thecvf.com/content/ICCV2023/html/Gasperini_Robust_Monocular_Depth_Estimation_under_Challenging_Conditions_ICCV_2023_paper.html.

Copyright Statement. ©2023 IEEE. Reprinted, with permission, from Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari, 'Robust Monocular Depth Estimation under Challenging Conditions', IEEE/CVF International Conference on Computer Vision (ICCV), October 2023, Paris, France.

The supplementary material can be found in Section B.4 and the license in Appendix C.

Robust Monocular Depth Estimation under Challenging Conditions

Stefano Gasperini^{*,1,2} Nils Morbitzer^{*,1} HyunJun Jung¹
Nassir Navab¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² VisualAIs ³ Google

Abstract

While state-of-the-art monocular depth estimation approaches achieve impressive results in ideal settings, they are highly unreliable under challenging illumination and weather conditions, such as at nighttime or in the presence of rain. In this paper, we uncover these safety-critical issues and tackle them with *md4all*: a simple and effective solution that works reliably under both adverse and ideal conditions, as well as for different types of learning supervision. We achieve this by exploiting the efficacy of existing methods under perfect settings. Therefore, we provide valid training signals independently of what is in the input. First, we generate a set of complex samples corresponding to the normal training ones. Then, we train the model by guiding its self- or full-supervision by feeding the generated samples and computing the standard losses on the corresponding original images. Doing so enables a single model to recover information across diverse conditions without modifications at inference time. Extensive experiments on two challenging public datasets, namely *nuScenes* and *Oxford RobotCar*, demonstrate the effectiveness of our techniques, outperforming prior works by a large margin in both standard and challenging conditions. Source code and data are available at: <https://md4all.github.io>.

1. Introduction

Estimating the depth of a scene is a fundamental task for autonomous driving and robotics navigation. While supervised monocular depth estimation approaches have achieved remarkable results, they rely on ground truth data which is expensive and time-consuming to produce [19, 12]. This requires costly 3D sensors (e.g., LiDAR) and significant additional data processing [19, 12].

To circumvent these issues, geometrical constraints on stereo pairs or monocular videos have been widely explored

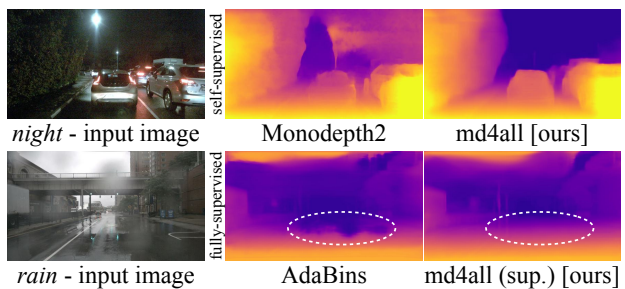


Figure 1. Predictions in challenging settings [3] for self-supervised [11] and supervised [1] methods. Standard approaches fail due to training assumptions or sensor artifacts. Under both supervisions, our *md4all* makes the same models robust in all conditions.

to learn depth estimation in a self-supervised manner [11, 25, 34, 7, 10]. Monocular training solutions are the most inexpensive and rely on the smallest amount of assumptions on the sensor setup, as they require only image sequences captured by a single camera.

Self-supervised methods rely on photometric assumptions and pixel correspondences [11, 34]. State-of-the-art approaches [11, 41, 32] deliver sharp and accurate estimates in standard conditions (i.e., sunny and cloudy), but suffer from a variety of inherent issues, such as scale ambiguity and difficulties with dynamic objects. While prior works have already proposed robust methods to address these problems [12, 8], there is still a major issue preventing the wide applicability of self-supervised depth estimators in safety-critical settings, such as autonomous driving. Darkness and adverse weather conditions (e.g., night, rain, snow, and fog) introduce noise in the pixel correspondences. As displayed in Figure 1, this is detrimental to the effectiveness of such methods, thereby requiring ad hoc solutions.

As shown in Figure 2, this problem is particularly severe at nighttime due to reflections (e.g., caused by streetlights and vehicle headlights), noise, and the general inability of the embedded cameras to capture details in dark areas. This leads to wrong depth estimates, which can be dangerous in safety-critical settings. A few pioneering works have already explored this problem, albeit with highly-

* The authors contributed equally.

Contact author: Stefano Gasperini (stefano.gasperini@tum.de).

complex pipelines and significant architecture changes affecting inference as well [38, 37, 22, 33, 36], such as illumination-specific branches. Additionally, prior methods that can operate both at night- and daytime introduce a significant trade-off concerning the standard daytime performance [22, 37], highlighting the need for a new solution.

In adverse weather conditions such as rain, monocular models are similarly fooled by reflections and decreased visibility. However, rain introduces another problem. While radars are robust in such conditions, LiDARs become unreliable, as they introduce multi-path and the so-called blooming effects (Figure 2). In autonomous driving, since supervised depth estimation approaches learn from LiDAR data, this causes them to learn also such erroneous measurements, rendering them unreliable in rainy settings (Figure 1). Analogous issues occur with snow and fog. These problems are relatively unexplored, demanding new solutions.

Alarming, no general solution currently allows an image-based depth estimator to work reliably under all conditions. Since LiDAR can constitute a misleading training signal in adverse weather, and pixel correspondences are problematic too (e.g., at night), neither existing supervised [24] nor self-supervised [11, 34] techniques work well in such challenging settings. A straightforward solution for the supervised case would be using synthetic data [40, 30], as by simply not modeling the sensor issues, a simulator could produce perfect ground truth in adverse weather. However, this is not only unexplored, but it would introduce a series of problems, such as a substantial syn2real gap due to the difficulty of modeling challenging conditions realistically (requiring, e.g., domain adaptation).

In this paper, we address these open issues with a simple and effective solution that works reliably in a variety of conditions and for multiple types of supervision. We approach this challenging problem by considering the success of existing methods in standard illumination and weather settings [11, 13, 12, 8]. This motivated us to find a way for them to work also under challenging scenarios, exploiting what makes them learn depth effectively in ideal conditions. Our core idea is based on training the model by providing always valid training signals as if it was sunny or cloudy, even when samples with adverse conditions are given. We apply this general principle to both supervised and self-supervised depth estimation via a set of techniques to improve the model robustness and reduce the performance gap between standard and hard conditions. The main contributions of this paper can be summarized as follows:

- We show how estimating depth in adverse conditions (e.g., night and rain) is problematic for both self- and fully-supervised approaches, requiring new solutions.
- We propose md4all: a simple and effective technique to make standard models robust in diverse conditions.

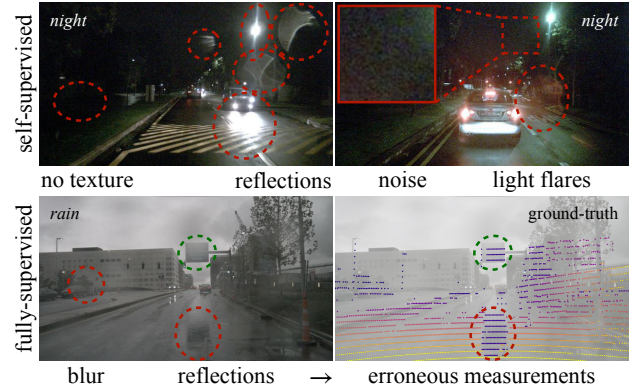


Figure 2. Detrimental factors to monocular depth estimation in difficult settings from nuScenes [3]. Self-supervised works have issues with textureless areas, reflections, and noise. Supervised ones learn artifacts from the ground truth sensor (LiDAR is shown).

- We apply our generic method to both fully- and self-supervised monocular settings.
- We generate and share open-source images in adverse conditions corresponding to the sunny and cloudy samples of nuScenes [3] and Oxford Robotcar [23].

With md4all, we substantially outperform prior solutions delivering robust estimates in a variety of conditions.

2. Related Work

2.1. Supervised Monocular Depth Estimation

The problem of estimating depth from a single color image is challenging due to the countless 3D scenarios that can produce the same 2D projection, making it an ill-posed problem. Nevertheless, significant progress has been made, thanks to the introduction of CNN-based architectures by Eigen et al. [5] and fully-convolutional networks with residual connections by Laina et al. [20] to estimate dense depth maps from monocular inputs. While many supervised methods have focused on directly regressing to depth measurements from LiDAR sensors (as in KITTI [9]) or RGB-D cameras (as in NYU-Depth v2 [31]), DORN [6] tackles the task in an ordinal manner. AdaBins [1] extended DORN via a linear combination of predictions across adaptive bins. Moreover, BTS uses a multi-stage local planar guidance [21] and P3Depth exploits coplanar pixels [24]. Others investigated the benefit of depth estimation while tackling other tasks, such as 3D object detection [17].

Issues While the supervision signal from 3D sensors is reliable in ideal conditions (e.g., sunny, cloudy), it severely degrades in photometrically challenging scenarios [19]. Outdoor, LiDAR sensors deliver erroneous measurements in adverse weather conditions, such as rain, snow and fog. As Jung et al. demonstrated indoor [19], training on an inexact ground truth leads depth models to learn the sensor ar-

tifacts and deliver wrong outputs. This problem is relatively unexplored outdoors, e.g., with rain. A few works investigated depth completion in simulated settings with LiDAR and radar in input [40] or event cameras and RGB [30]. In this paper, we explore this issue on AdaBins [1] and provide a simple solution to estimate depth reliably in diverse conditions, regardless of the sensor artifacts.

2.2. Self-Supervised Monocular Depth Estimation

To bypass the need for expensive LiDAR data, self-supervised methods employ view reconstruction constraints through stereo pairs [7, 10] or monocular videos [11, 46]. The latter utilizes motion parallax from a moving camera in a static environment [35] and requires simultaneous depth and camera pose transformation prediction. Significant advancements have been made since Zhou et al.’s pioneering video-based approach [46], including novel loss terms [11], network architectures that preserve details [12], the use of cross-task dependencies [18, 13], pseudo labels [25], vision transformers [44], uncertainty estimation [26], and 360 degrees depth predictions [15].

2.2.1 Solutions to Inherent Issues

Scale ambiguity Video-based methods predict depth up to scale, requiring median-scaling with ground truth data at test time [11]. Guizilini et al. [12] used the readily available odometry information to achieve scale awareness via weak velocity supervision on the pose transformation.

Dynamic scenes Due to the moving camera in a static world assumption [35], video-based methods have issues with dynamic objects, e.g., cars. To address this, Monodepth2 [11] uses an auto-masking loss on the static pixels, R4Dyn [8] adds weak radar supervision on the objects, and DRAFT [14] combines optical and scene flows.

Darkness Low visibility is detrimental to the losses used to learn depth because noise and lack of details prevent establishing pixel correspondences across the frames. DeFeat-Net [33] was among the first to mitigate this, with a cross-domain dense feature representation. ADFA [36] uses a generative adversarial network (GAN) to adapt nighttime features to daytime ones. R4Dyn [8] shows that radar is beneficial not only for dynamic objects but also at nighttime as a byproduct. RNW [38] reduces the irregularities at nighttime via, e.g., image enhancement and a GAN-based regularizer. ADIDS [22] uses separate networks for day and night images, partially sharing weights. ITDFA [43] is similar to ADFA, doing feature adaptation from night to day, with images generated with a GAN. WSGD [37] combines denoising with a lighting change decoder to predict per-pixel changes. While these works made significant steps towards solving the problem, they either have complex pipelines with dedicated branches for day and

night [22, 43], use additional sensors [8], suffer from a significant trade-off on the daytime performance [37], or are not meant to operate on multiple conditions, such as both day and night [36, 38, 43]. Therefore, an effective solution without inference complications is yet to be found.

Adverse weather As at nighttime, in adverse weather such as rain, fog, and snow, the limited visibility prevents establishing correct correspondences. Even fully-supervised approaches have issues in these settings [19]. So far, only a handful of works have explored depth estimation with adverse weather. ITDFA [43] requires an encoder for each condition and was not shown to work in both standard and adverse settings. R4Dyn [8] and MonoViT [44] are robust methods that delivered improvements also in adverse conditions as a side effect. Thus, this problem is largely unexplored, demanding a general solution.

Unlike prior works, in this paper, we propose a simple and effective solution enabling a standard monocular model to estimate depth in diverse conditions (e.g., day, night, and rain) without any difference at inference time compared to a common encoder-decoder pipeline [11]. Additionally, ours does not degrade the output quality in standard settings.

3. Method

In this paper, we enable a model to estimate depth reliably in diverse conditions (e.g., day, night, and rain). Displayed in Figures 3 and 4, our techniques exploit the effectiveness of existing approaches in standard conditions (e.g., daytime in good weather) to increase their robustness in adverse settings. Towards this end, we perform day-to-adverse image translation, train on the generated adverse samples, and learn only from valid training signals from the original day inputs. This simple idea is suitable to both self-supervised (Section 3.1) and supervised (Section 3.2) frameworks and is general to operate under various weather and illumination settings (including fog and snow).

3.1. md4all - Self-Supervised

We build upon a scale-aware video-based monocular method (Section 3.1.1). As described in Section 2.2.1, night and bad weather cause issues to self-supervised approaches. We address this with md4all by computing the losses only on the ideal samples corresponding to the hard ones given as input (Section 3.1.2). We then take this concept even further by distilling knowledge from a frozen self-supervised model trained only on the ideal samples (Section 3.1.3).

3.1.1 Self-Supervised Baseline

We build on a standard video-based monocular depth baseline equivalent to the framework shown in Figure 4 when considering $x = 0$ (i.e., no translation). We predict both the depth D_t of a target frame and the pose transformations

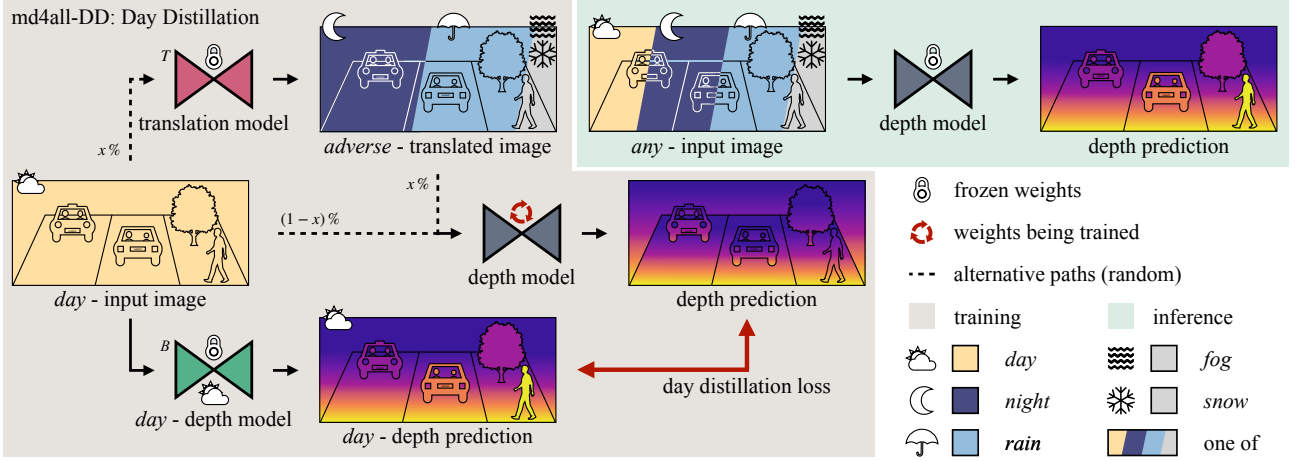


Figure 3. Our md4all-DD framework. The frozen *day* - depth model estimates on easy samples and provides guidance to another model fed with a mix of easy and translated inputs. Inference is done with a simple single model for both fully- and self-supervised md4all.

between the target I_t and source frames $I_{s \in \{t-1, t+1\}}$, with which we warp the source into a reconstructed target view. As in [11, 12], a loss is computed on the appearance shift between I_t and the reconstruction [46], alongside the structural similarity [39]. Following [11], we account for partial occlusions via the minimum reprojection error $\mathcal{L}_p$, and we ignore static pixels. Another loss $\mathcal{L}_s$ promotes smoothness and preserves edges [10]. $\mathcal{L}_p$ and $\mathcal{L}_s$ are calculated at all decoder scales, upsampled to the input size [11].

So far, this is equivalent to Monodepth2 [11]. Then, we add the weak velocity supervision $\mathcal{L}_v$ to achieve scale-awareness [12] and allow consistent predictions, beneficial when distilling knowledge between different models.

Architecture Unlike previous works having specialized branches [22, 43], we leave the architecture unchanged (e.g., [11]). Instead, we act on the training process. Our approach is general and not bound to a specific architecture.

3.1.2 md4all-AD: Always Daytime, No Bad Weather

Our md4all-AD configuration is shown in Figure 4. The core idea is learning from easy samples, even when given challenging ones (e.g., night) as if it was always daytime with good visibility (i.e., sunny or cloudy). This allows using the same established losses described in Section 3.1.1, which would otherwise fail with difficult inputs.

Day-to-adverse translation To achieve the above, we need easy samples corresponding to the challenging ones. This means having paired images (e_i, h_i^c) , with $e_i \in E$ and E being the set of easy samples (i.e., sunny or cloudy), $h_i^c \in H$ with H the set of the difficult samples from the conditions of interest $c \in C$ (e.g., snow). While an image translation method could convert the training H into easy ones, removing information is easier than adding it. Therefore, we generate H from E (e.g., turning e_i into nighttime).

Specifically, for each e_i and each condition c we aim to improve (e.g., night and rain), we obtain $h_i^c = T^c(e_i)$. We do this with c image translation models T^c trained at an earlier stage, increasing the training set size by $C \times E$.

Training scheme We then train depth and pose models as shown in Figure 4. During training, we feed to the depth model m_i , which is either h_i^c (for $x\%$ of the inputs, as a random mix of c) or e_i from the pre-existing training data. Additionally, we normalize the inputs depending on the recording time (i.e., day/night) to learn robust features agnostic of the input condition. The Appendix shows how performing this step only during training delivers similar results. Then, in the case of particularly noisy night samples (e.g., nuScenes [3]), we augment the inputs with heavy noise. The pose model always takes the sequence $[e_{i-1}, e_i, e_{i+1}]$, corresponding to m_i . If fed h_i^c , the pose network would have issues assessing the pixel correspondences.

Learning in all conditions Computing the losses $\mathcal{L}_p$ and $\mathcal{L}_s$ on h_i^c would lead to issues because of the difficulty of establishing correspondences in adverse conditions (Section 2.2.1). For this reason, training on E and deploying on H is more effective than training on both (Section 4.2), proving the limitations of standard methods. Our solution to this challenging problem is relatively simple: as shown in the figure, we provide a reliable training signal by always calculating the losses on E . Specifically, they are always computed on e_i , even when the depth model is fed with h_i^c ($x\%$). This constructed setting constitutes the ideal condition in which the losses $\mathcal{L}_p$ and $\mathcal{L}_s$ are already proven successful [11], eliminating the source of the issues. This leads the depth model to learn to extract robust features, regardless of whether the input belongs to E or H .

Inference After training depth and pose models, the latter is discarded, while our depth model is a simple encoder-decoder capable of estimating depth in multiple conditions.

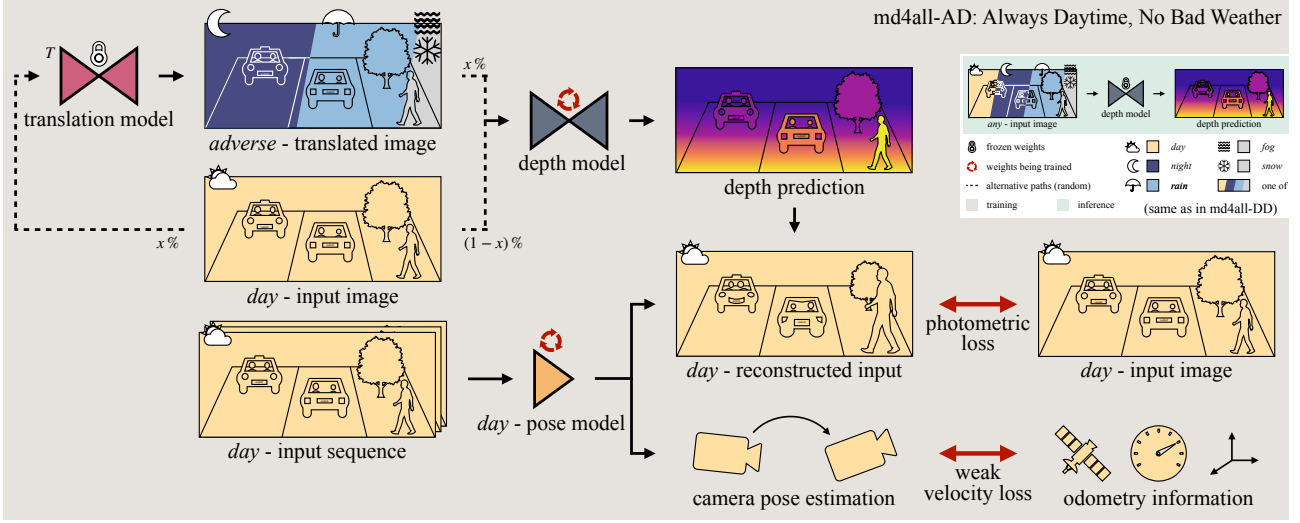


Figure 4. Our self-supervised md4all-AD framework. With $x = 0$, it is equivalent to the *day* - depth model in Figure 3 and the baseline. The depth model is trained with a mix of easy and translated samples, while the training signal is always from the easy ones.

As shown at the top of Figure 3, since we do not apply any architectural modification, at inference time, we predict depth with the same model through the same model parameters, regardless of the input condition. While dedicated models or branches may lead to better performance, switching between them is not always trivial, e.g., at dusk or with light rain. Therefore, we opted for a single monocular model, which does not penalize inference time compared to the same model trained only on E .

3.1.3 md4all-DD: Day Distillation

We take md4all-AD (Section 3.1.2) to the next level by simplifying the training scheme with md4all-DD. The core idea of md4all-DD is the same as for md4all-AD: we aim to learn depth only from E , pretending that the conditions C detrimental for the losses never occur.

Our md4all-DD framework mimics model estimates in ideal settings E , regardless of the difficulty of the input. As shown in Figure 3, we achieve this via knowledge distillation from a depth network B (baseline) trained at an earlier stage on E to a new depth model DD for both easy and adverse scenarios (i.e., E and H). The latter is fed m_i , i.e., the same mix of e_i and h_i^c as in md4all-AD (Section 3.1.2), while the former is given only e_i . DD is optimized solely through the following objective:

$$\mathcal{L}_d = \frac{1}{N} \sum_{j=1}^N \frac{|DD(m_i)_j - B(e_i)_j|}{DD(m_i)_j} \quad (1)$$

where N is the number of pixels, $DD(m_i)$ is DD 's depth prediction on m_i (i.e., an easy or hard sample), and $B(e_i)$ is B 's estimation on e_i (i.e., an easy sample). DD learns to

follow B at the output level, even when fed the problematic h_i^c , without being affected by the detrimental factors occurring in adverse settings. Inference is unchanged.

3.2. md4all - Supervised

Learning depth from a 3D sensor in adverse conditions exposes issues inherent to the sensor and the way it measures depth [19]. With bad weather (e.g., rain), LiDARs provide erroneous measurements (Figure 2), so learning from their signal means copying their artifacts as well (Figure 1). This has been ignored so far for monocular depth.

Regardless of the input, we address the sensor issues by learning from E . Analogously to the self-supervised setting, we use image pairs (e_i, h_i^c) and specify our method as md4all-AD following the self-supervised definition (Section 3.1.2), except for the supervision signal. Thus, we train the depth model with m_i and learn from the LiDAR signal of e_i . Thus, the supervision is from artifact-free data in ideal conditions E , such that the models never experiences the sensor issues. As in the self-supervised setup, the inference is unchanged. While md4all-DD (Section 3.1.3) also applies to the supervised case, using AD is more reasonable since reliable ground truth data from e_i is available.

4. Experiments and Results

4.1. Experimental Setup

Datasets and metrics We used two public driving datasets containing various illumination and weather conditions: nuScenes [3] and Oxford RobotCar [23]. **nuScenes** is a challenging large-scale dataset with 15h of driving in Boston and Singapore, diverse scenes, and difficult conditions. We distinguished good visibility (i.e., *day-clear*),

Method	sup.	tr.data	<i>day-clear</i> – nuScenes			<i>night</i> – nuScenes			<i>day-rain</i> – nuScenes		
			absRel	RMSE	δ_1	absRel	RMSE	δ_1	absRel	RMSE	δ_1
Monodepth2 [11]	M*	<i>a: dnr</i>	0.1477	6.771	85.25	2.3332	32.940	10.54	0.4114	9.442	60.58
Monodepth2 [11]	M*	<i>d</i>	0.1374	6.692	85.00	0.2828	9.729	51.83	0.1727	7.743	77.57
PackNet-SfM [12]	Mv	<i>d</i>	0.1567	7.230	82.64	0.2617	11.063	56.64	0.1645	8.288	77.07
R4Dyn w/o r in [8]	Mvr	<i>d</i>	0.1296	6.536	85.76	0.2731	12.430	52.85	0.1465	7.533	80.59
R4Dyn [8] (radar)	Mvr	<i>d</i>	0.1259	6.434	86.97	0.2194	10.542	62.28	0.1337	7.131	83.91
RNW [38]	M*	<i>dn</i>	0.2872	9.185	56.21	0.3333	10.098	43.72	0.2952	9.341	57.21
[ours] baseline	Mv	<i>d</i>	0.1333	6.459	85.88	0.2419	10.922	58.17	0.1572	7.453	79.49
[ours] md4all-AD	Mv	<i>dT(nr)</i>	0.1523	6.853	83.11	0.2187	9.003	68.84	0.1601	7.832	78.97
[ours] md4all-DD	Mv	<i>dT(nr)</i>	0.1366	<u>6.452</u>	84.61	0.1921	8.507	71.07	<u>0.1414</u>	<u>7.228</u>	<u>80.98</u>
AdaBins [1]	GT	<i>a: dnr</i>	0.1384	5.582	81.31	0.2296	7.344	63.95	0.1726	6.267	76.01
[ours] md4all-AD	GT	<i>dnT(r)</i>	0.1206	4.806	88.03	0.1821	6.372	75.33	0.1562	5.903	82.82

Table 1. Evaluation of self- and GT-supervised methods on the nuScenes [3] validation set. Supervisions (sup.): M: via monocular videos, *: test-time median-scaling via LiDAR, v: weak velocity, r: weak radar, GT: via LiDAR data. Training data (tr.data): *d*: *day-clear*, *T*: translated in, *n*: *night* (incl. *night-rain*), *r*: *day-rain*, *a*: *all*. Visual support: 1st, 2nd, 3rd best. More conditions and metrics in the Appendix.

night (including *night-rain*), and *day-rain*. We used the official split following R4Dyn [8], with 15129 training images (with synced sensors), and 6019 validation ones (of which 4449 *day-clear*, 602 *night*, and 1088 *rain*). **RobotCar** was collected in Oxford, UK, by traversing the same route multiple times in a year. It features a mix of *day* and *night* scenes. We followed the split and setup of WSGD [37], with 16563 *day* training samples and 1411 test ones (with synced sensors, of which 709 *night*). While we focused on night, rain, sun, and overcast, the Appendix shows preliminary results with fog and snow from the **DENSE** dataset [2]. We report on the standard metrics and errors up to 50m for RobotCar as in [37], and 80m for nuScenes as in [8]. More results can be found in the Appendix.

Implementation details Our self-supervised models use a ResNet-18 backbone [16] and learn from an image triplet sized 576x320 for nuScenes and 544x320 for RobotCar. The supervised model and md4all-DD are given only one keyframe. At inference time, all models take a single RGB input. We set $x = |C|/(|C|+1) \%$, with $|C|$ being the number of the adverse conditions of interest C , e.g., $x = 66\%$ for a model to work with *rain*, *night* and *day*, and within $x\%$ we used equally distributed data among C . So, our models see an equal amount of inputs for each condition. We used the same hyperparameters as Monodepth2 [11] and AdaBins [1] for self- and fully-supervised models, respectively. All models were trained on a single 24GB GPU.

Image translation We translated each e_i image to h_i^c . Diffusion models [27, 28] are not suitable due to the lack of already paired images. Datasets with multiple drives on the same roads [4, 29, 23] do not solve this issue due to the lack of synchronization and environmental changes. So we opted for GANs. For each condition c , we used a ForkGAN model [45] T^c to translate all *day-clear* training samples E of nuScenes, with $c \in C = \{\text{night}, \text{rain}\}$. We trained Fork-

GAN on BDD100K [42] and fine-tuned it on the nuScenes training set. For RobotCar, we used T^c to translate all *day* samples E into *night* ones. RobotCar contains more *night* samples than nuScenes, so we trained T^c directly on its training set. We share publicly all generated h_i^c images.

Prior works and baselines We compared ours with a variety of works [8, 11, 12, 38, 37, 22, 33, 1]. We applied ours on the self-supervised Monodepth2-based baseline of Section 3.1.1 and the fully-supervised AdaBins [1].

4.2. Quantitative Results

Night – nuScenes In Table 1, we report results for nuScenes [3] across various settings. Night samples present strong noise levels and reflections that are detrimental for self-supervised models (Figure 2), causing the absRel errors of most methods to double from ideal conditions (i.e., *day-clear*) to *night*. The difficulty of learning from night inputs is evident comparing Monodepth2 [11] trained only on *day-clear* (*d*) against *all* conditions (*a*), with the latter severely underperforming. PackNet [12] improved at *night* and *rain*, albeit doing worse in standard settings, possibly due to its large model and the relatively small dataset. PackNet’s velocity supervision also helped over Monodepth2 (md2) with our baseline. Thanks to the extra radar signal, R4Dyn [8] delivered significant improvements, although at *night*, only adding radar in input was beneficial over md2. md2 trained only on *day-clear* data outperformed RNW’s complex pipeline [38]. We retrained RNW on the official split (the authors reported an absRel of 0.3150 at *night* on their split [38]). Remarkably, despite being based on the same model as md2, at *night*, our simple techniques reduced absRel by 32% and relatively increased δ_1 by 37% (DD). Our md4all also outperformed the radar-based R4Dyn at *night*. This is thanks to the ability of our method to extract robust features from monocular data even in the dark.

Method	source	sup.	tr.data	day – RobotCar				night – RobotCar			
				absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
Monodepth2 [11]	[ours]	M*	d	0.1196	0.670	3.164	86.38	0.3029	1.724	5.038	45.88
DeFeatNet [33]	[37]	M*	$a: dn$	0.2470	2.980	7.884	65.00	0.3340	4.589	8.606	58.60
ADIDS [22]	[37]	M*	$a: dn$	0.2390	2.089	6.743	61.40	0.2870	2.569	7.985	49.00
RNW [38]	[37]	M*	$a: dn$	0.2970	2.608	7.996	43.10	0.1850	1.710	6.549	73.30
WSGD [37]	[37]	M*	$a: dn$	0.1760	1.603	6.036	75.00	0.1740	1.637	6.302	75.40
[ours] baseline	[ours]	Mv	d	0.1209	0.723	3.335	86.61	0.3909	3.547	8.227	22.51
[ours] md4all-DD	[ours]	Mv	$dT(n)$	0.1128	0.648	3.206	87.13	0.1219	0.784	3.604	84.86

Table 2. Evaluation of self-supervised works on the RobotCar [23] test set. Trailing 0 added to the values from [37]. Notation from Table 1.

Night – RobotCar In Table 2, we report results for RobotCar [23]. Here we compare with various approaches that also target depth estimation in challenging conditions [33, 22, 38, 37]. They all focus on *night* issues, tested here. Our md4all outperforms them all across the board, with substantially better estimates at *night* than theirs during the *day*: the previous best WSGD [37]’s *day* absRel error is 45% higher than ours at *night*. This is thanks to the simplicity of our approach, which does not rely on complex architectures, but makes existing models robust in adverse conditions by changing their input and training signals.

Rain – nuScenes Rain is less problematic than darkness due to the lack of cues in the latter. Results are shown in Table 1, with all methods performing better with *rain* than at *night*. Our self-supervised monocular md4all-DD significantly improved over Monodepth2 and the baseline, performing close to the radar-based R4Dyn [8].

Fully-supervised Table 1 reports also results in supervised settings. LiDAR data is reliable in the dark, so *night* scenes are less of an issue. Instead, *rain* inputs are particularly interesting for supervised works due to the reflection issues shown in Figures 1 and 2. For supervised settings, we applied our method on AdaBins [1]. It is to be considered that LiDAR artifacts may have an impact on the *rain* values, such that perfect estimates would not score perfectly because the ground truth is wrong (Figure 2). So, while we can assess the improvements of md4all at handling the blur caused by raindrops, we cannot correctly quantify its impact on eliminating the artifacts. Therefore, these comparisons are more meaningful when considered alongside qualitative outputs (Figure 5). Our supervised md4all performed better than AdaBins both quantitatively and qualitatively, eliminating the dependency on the sensor artifacts. Additionally, thanks to the strong regularization introduced by the translated samples, our model generalizes significantly better than the standard AdaBins, leading to vast improvements across the board, also at *night*. Training on the sparse LiDAR signal of nuScenes [3] (Figures 2 and 5) can lead to overfitting. Ours is a beneficial data augmentation technique, adding diversity to the training, as the model is shown $|C| + 1$ variations of each *day-clear* input.

Day(-clear) While we do not include any modification addressing standard conditions, we still see improvements over the baselines across both datasets and supervision types (Tables 1 and 2). This is due to the training mix of easy and translated samples acting as a strong data augmentation and regularization technique. Since the same weights are optimized on all conditions, they learn to extract robust features which are beneficial also with good visibility. Instead, RNW [38] is meant for operating only at *night*.

All conditions Remarkably, across all tested conditions, md4all significantly improves over Monodepth2 and AdaBins on which we applied it, without the need for specialized branches (Tables 1 and 2). There is no trade-off introduced when training our unique md4all model for multiple conditions, as the scores and errors remain equivalent or even improve compared to training only in ideal settings. This proves the effectiveness and generality of our simple ideas. The Appendix includes preliminary results with *snow* and *fog* on the challenging DENSE dataset [2].

AD and DD Our md4all delivers improvements both as DD and AD (Tables 1 and 2). While the two are applicable under both supervisions, available and reliable ground truth alongside the *day-clear* data makes AD more suitable for supervised setups. DD works better than AD in self-supervised settings thanks to the simplified training scheme and the guidance of our strong baseline.

Robustness against translations In Table 3, we assess the impact of the quality of image translation on our method. While the selected ForkGAN [45] translates better than CycleGAN [47], it does not give perfect outputs either (Appendix). Since we use the translations to learn robust features, their imperfections even help our model’s ro-

Method	avg/all	day	night
[ours] w/ CycleGAN [47]	0.1244	0.1159	0.1328
[ours] w/ ForkGAN [45]	0.1174	0.1128	0.1219
[ours] w/ degraded ForkGAN	0.1213	0.1159	0.1266

Table 3. Robustness of md4all-DD against translations from different GANs. Evaluation of absRel on the RobotCar [23] test set.

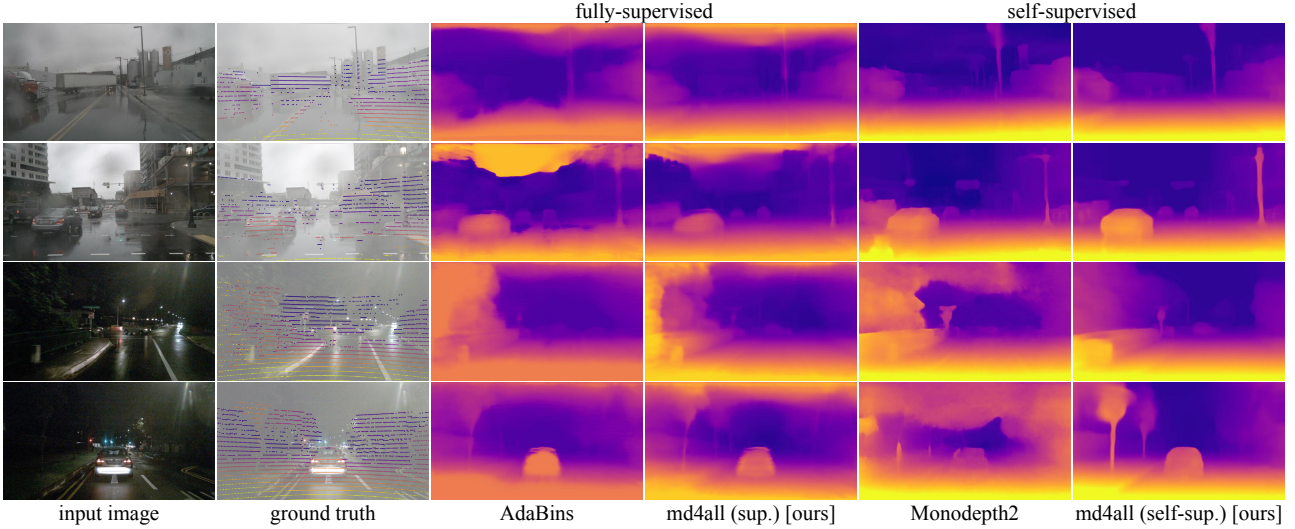


Figure 5. Comparison on nuScenes [3] between fully-sup. AdaBins [1] w/o and w/ ours, and self-sup. Monodepth2 [11] w/o and w/ ours.

bustness by making it harder to recover information for the depth task, as the translations act as data augmentation and regularization. The table confirms the robustness of md4all, performing similarly regardless of which GAN is used, even when degrading 10% of the inputs via random erasing.

4.3. Qualitative Results

Qualitative comparisons in Figures 5 and 6 confirm the quantitative findings, with our md4all delivering improved estimates in both adverse and standard conditions. On nuScenes [3] (Figure 5), unlike the baselines, both our models correctly identified the truck in the first rainy sample. As shown in Figure 2, rain leads to artifacts in the LiDAR ground truth, which cause the standard fully-supervised AdaBins [1] to learn them and estimate the road wrongly. Our supervised md4all exhibits no such artifacts as it was not trained with the problematic rainy samples but rather on our translated ones, which have reliable ground truth. Instead, self-supervised methods have issues at *night*. While Monodepth2 [11] could identify critical elements of the scenes (e.g., car and sign), its difficulties in extracting information in the dark are evident. Monodepth2 had fewer issues

with brighter *night* samples, as shown in the Appendix. Our self-supervised md4all delivered sharp estimates, identifying even the two trees on the left side of the bottom input, which are particularly hard to see. For RobotCar [23] (Figure 6), we compared on the same samples displayed by WSGD in their paper [37]. As in Table 2, our md4all delivered better and sharper estimates in both conditions, correctly estimating the people’s distance.

Limitations md4all improves in all tested conditions, but DD may propagate errors from the baseline. Thus, a stronger baseline would help. Despite the robustness against translations (Table 3), GANs [45] could be problematic. Better translations would help eliminate the domain gap, as seen with RobotCar (Table 2). GANs require many adverse images for training. Hard-to-distinguish data distributions (e.g., light snow vs. overcast) may create problems. md4all is applicable to stereo-based models too, but only given consistent translations for the stereo images. Future work may focus on eliminating the dependency on the GAN. Furthermore, md4all does not address the issue of dynamic objects, so flow [14] or weak radar supervision [8] may be beneficial, albeit adding complexity. The core ideas of this work can be extended to other tasks.

The **Appendix** includes a variety of extra results, e.g., experiments with *snow* and *fog*, and sample translations.

5. Conclusion

We presented the simple and effective md4all, enabling a single monocular model to estimate depth robustly in both standard and challenging conditions (e.g., night, rain). We showed md4all delivering significant improvements under both fully- or self-supervised settings, overcoming the detrimental factors that make adverse conditions problematic.

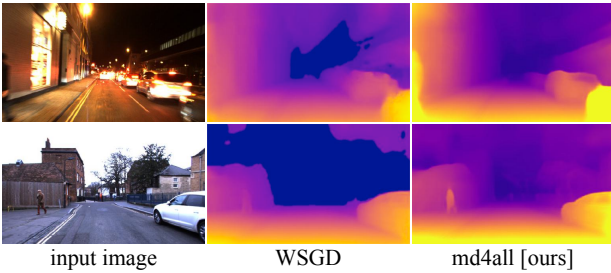


Figure 6. Comparison on RobotCar [23] samples between ours self-supervised and WSGD [37]. Outputs of WSGD are from [37].

References

- [1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 1, 2, 3, 6, 7, 8
- [2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 6, 7
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 2, 4, 5, 6, 7, 8
- [4] Carlos A Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, et al. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21383–21392, 2022. 6
- [5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014. 2
- [6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 2
- [7] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision*, pages 740–756. Springer, 2016. 1, 3
- [8] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021. 1, 2, 3, 6, 7, 8
- [9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2
- [10] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 1, 3, 4
- [11] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1, 2, 3, 4, 6, 7, 8
- [12] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 1, 2, 3, 4, 6
- [13] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations*, 2020. 2, 3
- [14] Vitor Guizilini, Kuan-Hui Lee, Rareş Ambruş, and Adrien Gaidon. Learning optical flow, depth, and scene flow without real-world labels. *IEEE Robotics and Automation Letters*, 7(2):3491–3498, 2022. 3, 8
- [15] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters*, 7(2):5397–5404, 2022. 3
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6
- [17] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. MonoDTR: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4012–4021, 2022. 2
- [18] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision*, pages 53–69, 2018. 3
- [19] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, Aleš Leonardis, Nassir Navab, and Benjamin Busam. On the importance of accurate geometry data for dense 3D vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023. 1, 2, 3, 5
- [20] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 2
- [21] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2
- [22] Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised monocular depth estimation for all day images using domain separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12737–12746, 2021. 2, 3, 4, 6, 7
- [23] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2, 5, 6, 7, 8

- [24] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022. 2
- [25] Andra Petrovai and Sergiu Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1588, 2022. 1, 3
- [26] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 3
- [27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 6
- [28] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *Proceedings of the ACM SIGGRAPH Conference*, pages 1–10, 2022. 6
- [29] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 6
- [30] Peilun Shi, Jiachuan Peng, Jianing Qiu, Xinwei Ju, Frank Po Wen Lo, and Benny Lo. EVEN: An event-based framework for monocular depth estimation at adverse night conditions. *arXiv preprint arXiv:2302.03860*, 2023. 2, 3
- [31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. *Proceedings of the European Conference on Computer Vision*, 7576:746–760, 2012. 2
- [32] Xibin Song, Wei Li, Dingfu Zhou, Yuchao Dai, Jin Fang, Hongdong Li, and Liangjun Zhang. MLDA-Net: Multi-level dual attention-based network for self-supervised monocular depth estimation. *IEEE Transactions on Image Processing*, 30:4691–4705, 2021. 1
- [33] Jaime Spencer, Richard Bowden, and Simon Hadfield. DeFeat-Net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020. 2, 3, 6, 7
- [34] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019. 1, 2
- [35] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 3
- [36] Madhu Vankadari, Sourav Garg, Anima Majumder, Swagat Kumar, and Ardhendu Behera. Unsupervised monocular depth estimation for night-time images using adversarial domain feature adaptation. In *Proceedings of the European Conference on Computer Vision*, pages 443–459. Springer, 2020. 2, 3
- [37] Madhu Vankadari, Stuart Golodetz, Sourav Garg, Sangyun Shin, Andrew Markham, and Niki Trigoni. When the Sun Goes Down: Repairing photometric losses for all-day depth estimation. In *Proceedings of the Conference on Robot Learning*, pages 1992–2003. PMLR, 2023. 2, 3, 6, 7, 8
- [38] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing Nighttime Weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16055–16064, 2021. 2, 3, 6, 7
- [39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4
- [40] Mengchen Xiong, Xiao Xu, Dong Yang, and Eckehard Steinbach. Robust depth estimation in foggy environments combining RGB images and mmWave radar. In *International Symposium on Multimedia*, pages 34–41. IEEE, 2022. 2, 3
- [41] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *International Conference on 3D vision (3DV)*, pages 464–473. IEEE, 2021. 1
- [42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. 6
- [43] Chaoqiang Zhao, Yang Tang, and Qiyu Sun. Unsupervised monocular depth estimation in highly complex environments. *Transactions on Emerging Topics in Computational Intelligence*, 6(5):1237–1246, 2022. 3, 4
- [44] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. MonoViT: Self-supervised monocular depth estimation with a vision transformer. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 668–678. IEEE, 2022. 3
- [45] Ziqiang Zheng, Yang Wu, Xinran Han, and Jianbo Shi. ForkGAN: Seeing into the rainy night. In *Proceedings of the European Conference on Computer Vision*, pages 155–170. Springer, 2020. 6, 7, 8
- [46] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 3, 4
- [47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 7

Part III

Conclusion and Outlook

Summary and Findings

This dissertation explored the problem of scene understanding for autonomous driving and its reliability challenges around consistency, robustness, and generalization. Various methods were proposed in this thesis to tackle open problems and improve the generalization to out-of-distribution data and corner cases. Such approaches span various tasks of scene understanding, from object detection to segmentation and depth estimation, and encompass a variety of techniques to address robustness, including adversarial methods and uncertainty estimation.

In contrast to mainstream approaches that aim to improve scores on established benchmarks and outperform the competition, this thesis focused on critical but often overlooked challenges in achieving full autonomy. Specifically, we aimed to enhance the applicability of state-of-the-art methods to the real world. Given the immense variability of the world, new solutions are needed to deploy perception algorithms reliably and safely.

Due to the benchmark-driven research permeating the computer vision domain, most works entirely ignore the data outside the distribution of popular datasets. However, reliability is of utmost importance in safety-critical settings, and models must be robust against new inputs and corner cases. Towards this end, we uncovered and mitigated systematic issues that learning-based methods exhibit with challenging scenarios. These include problems with out-of-distribution data in general and scenes with dynamic objects or low visibility for depth estimators.

First, we investigated reliability in terms of consistency across multiple outputs corresponding to different scene understanding tasks through panoptic segmentation. Towards this end, we proposed the first end-to-end panoptic segmentation method, Panoster, which delivers instance and semantic segments directly from a single neural network, significantly reducing runtime. As demonstrated by many subsequent research works, Panoster opened a new direction for panoptic segmentation. The proposed method departed from pre-existing approaches, which were either multi-stage or required costly grouping steps to form instances.

Then, for out-of-distribution data, we proposed various beneficial strategies to improve generalization and robustness. These include adversarial augmentation and uncertainty estimation. Regardless of the available training data, the key is expanding the distribution to include new plausible samples. While these could be theoretically captured during data collection, especially for rare scenarios, the difficulty of acquiring them as they happen leads to insufficiently diverse data. To circumvent this issue, we generate new samples artificially, starting from the already available data. In this context, we proposed a new effective technique for point clouds.

Furthermore, understanding the performance boundaries of a model is fundamental for ensuring an informed and reliable deployment. Uncertainty estimation is a powerful tool to do this. In this thesis, we showed how estimating the uncertainty is beneficial to assessing such boundaries and extending them. In particular, we showed how making the model aware of its limitations improves its performance

on out-of-distribution data. This is thanks to uncertainty estimation improving the confidence of the model's predictions, enhancing the overall output quality.

Nevertheless, with completely unknown objects and scenarios, such as with unseen object categories outside the pool of known classes, the performance boundaries are already clear when designing the model. If unknown classes are not addressed during the development, the model will be unreliable when they appear in input, raising significant safety concerns. This is even more problematic for semantic or panoptic segmentation, where models predict a known class for every input unit so that when fed an unknown object, they systematically fail, giving a wrong estimate in output.

In this thesis, we addressed this issue with a new method capable of segmenting known areas and completely unknown objects individually without prior knowledge about them. This raises scientific and philosophical questions about the concept of unknown objects and the role of external knowledge, especially in relation to the use and function of foundation models trained on such objects. To keep the settings as unconstrained as possible and reduce assumptions, we excluded any prior knowledge about unknowns by introducing a new setting called holistic segmentation.

For the task of depth estimation, we investigated systematic failures of popular models and learning paradigms due to violations of their training assumptions. For self-supervised approaches, such problems include the wrong estimation of dynamic objects and the difficulty of learning in challenging conditions such as at night or with rain. For supervised methods, we uncovered the issues derived from the dependency on LiDAR data with adverse weather. For example, LiDAR provides erroneous measurements with rain due to reflecting wet surfaces and multi-path effects. Towards this end, we also proposed solutions to these issues, again improving the models' robustness and applicability.

We proposed using radar to mitigate the problem of dynamic objects, fixing the wrong predictions with a weak supervision signal from the sparse radar measurements. Then, we addressed challenging conditions in both supervised and self-supervised settings with a monocular strategy exploiting image translation and the success of existing depth estimators in ideal settings, providing a reliable training signal regardless of the condition given in input.

While this dissertation proposed several methods to advance scene understanding for autonomous driving, it also shed light on the limitations and potential areas for further exploration. The next chapter delves into potential opportunities for future work.

In conclusion, by mainly focusing on real-world applicability and providing innovative solutions to previously unexplored problems, this work uncovered critical issues for autonomous driving and computer vision in general and proposed novel methods to address them, significantly increasing the performance of state-of-the-art approaches across various tasks and setups. Thus, this dissertation contributes to the ongoing global effort to reach autonomy, paving the way for future research and serving as a stepping stone toward safer and more reliable scene understanding for autonomous driving.

Future Work and Discussion

Contents

7.1	Methods to Make Models More Reliable	135
7.1.1	Specific Improvements to the Proposed Methods	135
7.1.2	Foundation Models	137
7.1.3	Best Use of Reliability Information	138
7.1.4	Temporal Models and End-to-End Frameworks	139
7.2	Data Generation and Processing	139
7.2.1	Adversarial Methods for Data Generation	139
7.2.2	Scene Manipulation	140
7.2.3	Synthetic Data	141

While autonomous driving has seen astonishing progress in recent years, many issues still need to be solved before the same autonomous vehicle can be deployed safely throughout entire countries, continents, and the world. Such open challenges span across various domains, from technical to regulatory. Focusing on the technical aspects, open problems are not only related to scene understanding but also include infrastructure [183, 227], controls [282, 334], sensors [168], path planning [125], and trajectory prediction [52]. This section further discusses possible directions to improve reliability in scene understanding for autonomous driving.

We identify two macro sources of improvements: methods and data. While having access to more diverse data helps with generalization and robustness, its availability and collection may be problematic. Therefore, developing methods to improve robustness, generalization, and, more generally, reliability is fundamental to mitigate the dependency on the training data. Furthermore, new strategies can be developed to generate new data starting from what is already available. Many works have been proposed tackling reliability [119, 122, 260], also in the directions described in this chapter. However, much work remains to be done to further strengthen reliability toward production-ready systems operating in safety-critical settings.

7.1 Methods to Make Models More Reliable

7.1.1 Specific Improvements to the Proposed Methods

The proposed methods would all benefit from improvements, also considering the significant advancements in the field since they were presented. Panoster [84] (Section 5.1.1) would significantly benefit from a better, more recent semantic model, such as SphereFormer [155], as that would directly impact the overall performance. Additional improvements can be applied to the clustering procedure [87].

To learn instance segmentation, Panoster optimizes jointly opposite objectives, which could lead to troubles converging to a global minimum. Instead, a unified clustering objective could ease the optimization. The same end-to-end ideas could be applied to the imaging domain. Furthermore, Panoster exhibits issues with scenes containing a high amount of objects, which is a rare occurrence in the training data. To overcome this limitation, the proposed instance segmentation could be adapted to work in sync with data augmentation strategies, such as copy and paste or object sampling [159]. A simple loss function for clustering could also aid U3HS in the context of unknown objects and the proposed holistic segmentation setting [85] (Section 5.2.3), thereby simplifying its complex optimization.

The proposed works using uncertainty estimation would directly benefit from better uncertainty estimates [82, 85]. While we modified the uncertainty estimation techniques towards improved estimates in the tasks at hand, we did not dig deep into this. Deeper and more theoretical investigations could be worth pursuing to find better uncertainty estimators that operate at higher sync with the task of interest. In both cases, we adapted existing techniques proposed for image classification [275, 310] to operate at the pixel-level for more complex tasks such as semantic segmentation [85] and center regression for object detection [82]. For semantic segmentation in the context of the proposed holistic segmentation [85], we did not explicitly use the information carried by neighboring pixels, which could be of help to better identify unknown objects by considering higher-level features.

With the holistic segmentation setting and the proposed U3HS [85], we only scratched the surface of what can be achieved for unknown objects without using external knowledge. The method would benefit from a wide range of improvements. First, we could distinguish unknown *things* from unknown *stuff* and only segment instances of the former. This could be done by learning to distinguish the two categories. As already mentioned, better uncertainty estimates would allow the method to deliver better masks in output. A limitation of using uncertainty in this context is the inability to differentiate between completely unknown and unusual yet known objects. This causes known areas to be treated as unknown. While it is possible that the amount of uncertainty indicates this difference, relying on a fixed threshold to identify unknowns can be limiting. Therefore, investigating an auto-tunable or context-dependent threshold may provide valuable insights. Furthermore, as seen in Section 5.2.3, the highly challenging task makes the output masks not always ideal. These could be improved in many ways. For sequential data, temporal information may help refine the unknown segments. Also, post-processing steps or integrating more context-aware information (e.g., via bilateral filtering [299]) may be beneficial. Finally, given the importance of the latent space for the method, a better regularization of it or the use of self-supervised contrastive learning techniques, such as SymCLR [43], could aid the construction of a more organized embedding space. With Prior2Former [269], we already further explored the assumption-free holistic segmentation setting proposed in this dissertation (Section 5.2.3) with the first evidential method for mask transformers, delivering strong performance both in- and out-of-domain.

In our journal article [163], we already extended 3D-VField [162] (Section 5.2.2) to address 3D semantic segmentation, incorporating a series of improvements and significantly more profound insights into the method. We added the LiDAR intensity as an extra dimension for the adversarial attack, thereby generating adversarial examples that make the model more robust against intensity shifts due to the sensor being used or the distance from it. We also explored using targeted adversarial examples to shift points to fool the model into thinking that the object of interest belongs to a different specific class. This strategy can be used to improve specific knowledge boundaries of a model. Nevertheless, lots of work can be done to explore these directions further. First, the proposed

adversarial augmentations could be extended to *stuff* classes (i.e., those which are not objects). This would prevent using the object-centric approach of bounding boxes and axis-aligned boxes, possibly with distance- and view-dependent vectors for *stuff* points and vector fields for the objects. Adversarial augmentation offers lots of opportunities to improve generalization and robustness. As shown, the key is constructing plausible and diverse examples. Therefore, new techniques to enhance the realism of the synthesized data, as well as methods to generate more diverse samples, would be highly beneficial. Similarly to works exploring the use of GANs for data augmentation [7], these techniques could be extended to the imaging domain towards new plausible hard data points and more reliable models.

Like every other method in this work, R4Dyn [83] (Section 5.3.1) would benefit from an improved baseline architecture, such as MonoViT [366]. Nevertheless, absolute performance was not the main goal of this thesis. Furthermore, the proposed techniques to incorporate radar could be improved by better expanding the radar influence on the objects or exploiting radar also on static areas, which R4Dyn discards by considering only the radar points that project to vehicles and pedestrians. However, noise and multi-path effects would be more difficult to tackle for static elements. Additionally, the radar expansion could be enhanced using object masks from an off-the-shelf model such as SAM [151], avoiding part of the point filtering that R4Dyn uses and leading to better supervision from the radar.

md4all [86] (Section 5.3.2) would also improve its performance by simply using a better architecture, such as MonoViT [366], especially as a baseline model for distillation. Depth artifacts should be removed as much as possible from the depth estimates of the baseline, so addressing dynamic objects, e.g., with radar as in R4Dyn [83], would be important towards even more reliable estimates. Furthermore, md4all could be enhanced by eliminating the dependency on the GAN or improving the generation of the translated images. The latter could be achieved by conditioning the image generation on the depth estimated in the ideal settings by the baseline. To properly evaluate the performance with rain, the LiDAR ground truth should be manipulated to remove the artifacts. Additionally, more thorough investigations are needed for snowy and foggy conditions, and so far, have only been explored with preliminary experiments in the supplementary material (Section B.4). Specifically, such preliminary investigations highlighted the need for a new benchmark with real data for monocular depth estimation in adverse weather conditions due to the inadequacy of existing datasets for this task. Therefore, future work could focus on collecting such data. The variability of snowy settings highlighted the need for fine-grained and precise metadata to properly categorize the inputs and allow the evaluation under different conditions.

7.1.2 Foundation Models

Given the widespread success of foundation models [39, 151, 215, 243], e.g., vision-language models, many researchers have used them as part of their methods [27]. However, many implementations act as mere interfaces, treating these models as infallible black boxes [92, 346]. Such uninformed usage can lead to unexpected failures of the wrapping systems [267], making them unreliable.

The problem is that foundation models are still learning-based systems relying on a training set [363], which is by definition limited [85, 162, 163]. Radford et al. [243] show that even foundation models exhibit issues with long tail and out-of-distribution samples. However, the performance boundary is significantly broader for foundation models compared to standard methods. Therefore, the gener-

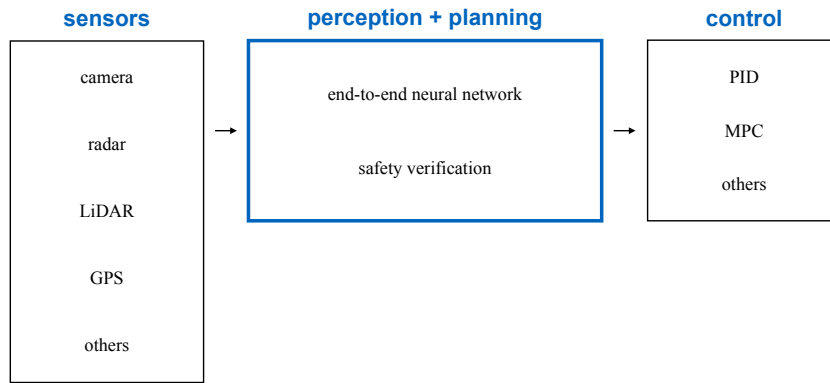


Figure 7.1. Representation of the end-to-end autonomous driving software stack. Compared to the standard stack shown in Figure 2.5, in this end-to-end setup, the perception, localization, and planning components are merged without needing an HD map. So, sensory inputs are processed by a unified perception and planning module, which implicitly performs scene understanding and directly outputs trajectories for the ego vehicle to follow, simplifying the stack.

alization issues still occur [363], but only on a subset of the data that causes failures to traditional methods.

The performance boundary is related to the training data distribution. Foundation models are often trained on proprietary, extensive, and annotated datasets, including data scraped from the web [39, 151, 215, 243]. However, users outside the organizations that create these models often lack visibility into the specifics of this distribution. This opacity raises concerns about how to effectively and responsibly use these models within autonomous driving systems and beyond.

Many approaches entirely rely on the predictions of foundation models and silently fail whenever the underlying model gives a wrong prediction [92, 346]. Instead, similarly to our work [85], future research could develop a method to handle robustly the objects that are unknown to the foundation model being used. These methods could enhance the transparency and interpretability of foundation models' decisions, further building trust in their reliability.

Moreover, how to best exploit the effectiveness and generalization ability of foundation models in the context of autonomous driving is an open problem. In this context, multimodality, for example, would also concern 3D data (e.g., from LiDAR and radar). Since task-specific models tend to perform better in their setting than general foundation models, there are opportunities for improvements in this domain.

7.1.3 Best Use of Reliability Information

In this thesis, we proposed new techniques to assess and improve the reliability of a model through generalization and robustness via adversarial methods [162, 163], uncertainty estimation [82, 85], and out-of-distribution evaluations. Many works are limited to the estimation of the uncertainty [82, 117, 203], the creation of adversarial examples [191, 306], or the computation of interpretability metrics [73, 146, 365]. Instead, we also used such information to improve the method [85, 162, 163]. However, given that these techniques have only started to gain traction in recent years, it is still unclear how to

best use reliability information to enhance the capability of a model. Future work could investigate this problem, thereby leading to better methods.

Additionally, while most works explore a certain topic or task in isolation, complex systems such as robots and autonomous vehicles require the orchestration of different components, including multiple perception and downstream modules (e.g., prediction and planning). In this thesis, we did not investigate how to use the reliability information extracted online for downstream tasks, e.g., uncertainty estimation for path planning [82, 85]. While end-to-end solutions from perception to control [144] circumvent this completely (Figure 7.1), it remains an open problem that could be explored with future research.

7.1.4 Temporal Models and End-to-End Frameworks

All of the methods proposed in this thesis operate on single frames at inference time [83, 84, 86, 162, 163]. The same holds for the vast majority of the approaches being researched because the benchmarks are typically set up in a single-frame manner. So, relatively few works explore temporal information to improve a model's predictions [208, 352]. However, in autonomous driving and robotic settings, the data is naturally temporal and composed of sequences of frames.

Such temporal data could be used to improve reliability by refining any predictions, e.g., object detection and semantic segmentation. Single-frame predictions at previous timestamps could be combined with an ensemble of models, or the model itself could consume multiple frames to provide more consistent predictions. The plurality of predictions at different timestamps could also be used to extract uncertainty estimates.

As shown in Figure 7.1, end-to-end autonomous driving approaches feature a single model from sensory inputs to trajectory estimations [144]. As these models gain attention and popularity in the field, to avoid single-point-of-failure scenarios, their reliability in terms of robustness and generalization is also of interest [5]. Future work could incorporate the proposed reliability techniques into end-to-end models.

7.2 Data Generation and Processing

As discussed in Section 4.4, data significantly impacts every learning-based model. In general, increasing the diversity of the data helps for generalization. However, in the real world, the challenge is collecting such diverse data. Most popular datasets [18, 89, 90] contain mainly average cases and only a few corner cases, if any. In this section, we discuss possible future directions to mitigate this issue, e.g., by generating corner cases artificially.

7.2.1 Adversarial Methods for Data Generation

As seen in ours [162, 163] and prior works [316], adversarial examples can be used for data augmentation. A significant advantage of this strategy is that new complex samples are generated automatically based on the knowledge gaps of the model under scrutiny. Essentially, adversarial methods offer

corner case discovery and generation. Therefore, there is theoretically no need to collect extensive data to sample the long tail of the underlying distribution. However, new challenges arise as adversarial methods often produce examples that are either too unrealistic [343] or too similar to the original data [191] to increase diversity in a valuable and substantial manner, introducing a delicate trade-off.

As discussed in Section 5.2.2, the key is to expand the training data distribution in a plausible way [162, 163]. Thus, developing new strategies and constraints is crucial to keep the adversarial examples plausible while introducing diversity in the data. With 3D-VField [162] and our journal extension [163], we proposed such a method for 3D point clouds. However, the realism and diversity of the adversarial examples can be further improved with new techniques and constraints and extended to the imaging domain. This would help reduce the requirements for the data collection, albeit needing significant domain knowledge to design such constraints. A line of work has explored using GANs for data augmentation [7]. This, in contrast to our 3D-VField, can possibly introduce more variation, with the risk of adding unrealistic samples. The realism could be mitigated with diffusion models [253].

7.2.2 Scene Manipulation

Following the idea of generating new samples from existing data, digitally manipulating existing natural scenes offers various advantages, especially in simulating corner cases that are challenging to collect in the real world.

Implicit representations [240, 280, 285], Neural Radiance Fields (NeRF) [201], and 3D Gaussian Splatting (3DGS) [145] present a promising direction in this context. NeRF and 3DGS enable the synthesis of novel views of a scene from input images [96, 246, 250, 302, 328, 344] even sparsely sampled [303], storing the scene information in a neural network with NeRF (i.e., implicitly) or an explicit 3D representation with 3DGS. Many applications have been explored with these technologies, including autonomous driving [45] and medical data [340]. Scene understanding can be done on the reconstructed 3D scene, e.g., via language queries in an open-vocabulary fashion [173]. Then, to improve the performance of scene understanding models, novel views can serve as augmentations. Furthermore, by leveraging these methods and decomposing the scene into individual objects and background [153], researchers and engineers can manipulate scenes to create new, unseen scenarios [45, 137], which can be used to enrich the diversity of the training data [44]. For example, nerf2nerf [97] enables the registration of two NeRFs and could be used to compose challenging yet plausible autonomous driving scenes. Such approaches would allow for the simulation of various environmental conditions, object placements, and scene layouts, offering a versatile tool for generating data that covers a wide range of possible real-world situations.

For instance, the scene of a clear, sunny day with sparse traffic could be manipulated to simulate a foggy day with high traffic density and pedestrians crossing the road unexpectedly. A standard straight road ahead may be edited for a curved street. A vehicle could be added on the side of the road, simulating a breakdown. Such manipulations would help add diversity to the training data, improving the model's reliability with different driving conditions and unexpected events.

Scene manipulation can be executed via scene graphs [61, 265]. A scene graph is a compact data structure representing the semantic configuration of a scene, capturing the objects present, their attributes, and the relationships between them [318, 337, 339, 359]. By directly altering the nodes

of the graph (i.e., the objects), their attributes (e.g., color), or their relation (e.g., proximity), new scenes can be generated by decoding the information of the graph [60]. Future work could learn to generate plausible scene graphs automatically and use them to create realistic novel scenes with corner cases. This could be done in autonomous driving settings with an adversarial method [277] to drive the exploration of corner cases, enforcing plausibility constraints on the graph and the visual output.

7.2.3 Synthetic Data

Synthetic data offers numerous advantages. Simulators can provide large amounts of data cost-efficiently with the ability to control the entire scene [63, 291]. However, there is often a disconnect between synthetic or generated environments and real-world scenarios [198]. This is due to the difficulty of modeling reality precisely, encompassing physics, textures, and lighting. To bridge this gap effectively, future research could explore new strategies to enhance the realism of synthetic data and style transfer techniques [182].

A complementary direction is that of domain adaptation techniques [9, 122, 140, 257] (Section 4.8). Domain adaptation can be employed to adapt the model's features to the real data while mainly learning from the synthetic data. While many strategies already exist [9], more research is needed to fully close the gap between synthetic and real data, reducing the need for large amounts of real data. The ability of foundation models [39, 151] to estimate adequately across various domains could be beneficial in this context. Diffusion models [25, 253] can also help improve the quality of the generated data.

Diffusion models [253] have changed how people think about data generation and artificial intelligence in general. However, there are still open problems concerning the conditioning of such models [48, 255]. Specifically, there are challenges to overcome before complex yet realistic driving scenes can be generated in a multi-view or temporally consistent manner, as required by difficult scene understanding tasks. Recently, Wayve announced GAIA-1 [124], a generative model for autonomous driving. GAIA-1 leverages video, text, and action inputs to generate driving scenarios in the image space. It predicts the next token or frame of an input video, delivering somewhat temporally consistent generated data. Despite this technology being in its infancy and suffering from various artifacts, especially at further distances, the generated data with GAIA-1 is already helpful towards improved generalization [124]. Yet, it is to be investigated how this can be extended to multi-view consistent videos and how the quality can be improved to match real data. Even more recently, Stability AI released Stable Video Diffusion [25], a latent video diffusion model for high-resolution text-to-video and image-to-video generation. For multi-view consistency, a research direction is that of 3D generative models [81]. Researchers are focusing on improving the generation quality at the object level, removing artifacts with promising results [278], but have not yet reached the ability to create highly complex multi-view scenes, such as driving scenarios. Moreover, 3D data is more diverse, more dependent on the specific 3D sensor, and significantly less available than images or text, making it more challenging to collect, annotate, and process. For these reasons, generative and foundation models have lagged in the 3D domain, demanding new solutions.

Part IV

Appendix

Abstracts of Publications not Discussed in this Thesis

This section includes the abstract of the publication that is not discussed in this dissertation [163]. Specifically, [163] is an extension of our CVPR 2022 work on adversarial augmentations [162] to an IJCV article, where we provide deep insights into the proposed method through various experiments, especially around the gained reliability with our method. We also extend it from 3D object detection to 3D semantic segmentation, LiDAR intensity, and targeted attacks.

3D Adversarial Augmentations for Robust Out-of-Domain Predictions

Alexander Lehner^{*,1,2} Stefano Gasperini^{*,1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
Nassir Navab¹ Benjamin Busam¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² BMW Group ³ Google

* equal contribution

International Journal of Computer Vision (IJCV)

Published in October 2023 and included in Volume 132, Number 3 of March 2024.

Published article: <https://doi.org/10.1007/s11263-023-01914-7>.

Abstract. *Since real-world training datasets cannot properly sample the long tail of the underlying data distribution, corner cases and rare out-of-domain samples can severely hinder the performance of state-of-the-art models. This problem becomes even more severe for dense tasks, such as 3D semantic segmentation, where points of non-standard objects can be confidently associated to the wrong class. In this work, we focus on improving the generalization to out-of-domain data. We achieve this by augmenting the training set with adversarial examples. First, we learn a set of vectors that deform the objects in an adversarial fashion. To prevent the adversarial examples from being too far from the existing data distribution, we preserve their plausibility through a series of constraints, ensuring sensor-awareness and shapes smoothness. Then, we perform adversarial augmentation by applying the learned sample-independent vectors to the available objects when training a model. We conduct extensive experiments across a variety of scenarios on data from KITTI, Waymo, and CrashD for 3D object detection, and on data from SemanticKITTI, Waymo, and nuScenes for 3D semantic segmentation. Despite training on a standard single dataset, our approach substantially improves the robustness and generalization of both 3D object detection and 3D semantic segmentation methods to out-of-domain data.*

This article is licensed under a Creative Commons Attribution 4.0 International License: <http://creativecommons.org/licenses/by/4.0/>.

Supplementary Materials of the Publications

Contents

B.1	Appendix: Domain Generalization with Adversarial Augmentations (CVPR 2022)	149
B.2	Appendix: Segmenting Known and Unknown Objects and Regions (ICCV 2023)	161
B.3	Appendix: Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)	177
B.4	Appendix: Monocular Depth Estimation in Challenging Conditions (ICCV 2023)	185

B.1 Appendix: Domain Generalization with Adversarial Augmentations (CVPR 2022)

This section includes the supplementary material associated with [162], presented and discussed in Section 5.2.2.

Supplementary Material of:

3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection

Alexander Lehner^{*,1,2} Stefano Gasperini^{*,1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
 Mohammad-Ali Nikouei Mahani² Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

* equal contribution

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022

Published version: https://openaccess.thecvf.com/content/CVPR2022/supplemental/Lehner_3D-VField_Adversarial_Augmentation_CVPR_2022_supplemental.pdf.

Supplementary Material of: 3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection

Alexander Lehner^{*,o,1,2} Stefano Gasperini^{*,1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
Mohammad-Ali Nikouei Mahani² Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

A. Supplementary Material

In this supplementary material we include further details and results. Specifically, Section A.1 describes the proposed CrashD out-of-domain dataset to a greater extent, Section A.2 provides additional implementation details, Sections A.3 and A.4 report more quantitative and qualitative results on outdoor data, while Section A.5 provides results on ToF camera data in indoor settings.

A.1. Details on the Proposed Dataset: CrashD

In this section we further describe the proposed dataset: CrashD. We refer the reader to the dataset webpage to see examples of the generated accidents and scenes.

A.1.1 Intended Use

This dataset was designed to evaluate the performance of LiDAR-based 3D object detectors on out-of-domain data. It is meant to serve as a test benchmark for 3D detectors trained on KITTI [1], Waymo [8], or similar datasets.

It should be noted, that CrashD is not intended for training and evaluating an object detector directly, since the generated LiDAR scenes do not include anything other than ground and cars. Therefore, training and evaluating on this dataset would be rather trivial, since the detector could learn that anything rising from the ground is a car, except for the relatively small spare parts separated by the accidents (e.g., the tire in Figure 1).

Nevertheless, reasonable uses of the proposed CrashD could include domain adaptation, transfer learning, and domain generalization [9], as well as synthetic-to-real trans-



Figure 6. LiDAR scene setup of CrashD. For each car, a black arrow indicates its damaged area, which is ensured to be visible from the sensor viewpoint. Image used with courtesy of BeamNG GmbH.

fers. Furthermore, it could be used to assess the damage of a vehicle, and also for uncertainty estimation or similar methods to detect out-of-distribution samples. Moreover, it could serve for point cloud reconstruction, or anomaly segmentation approaches comparing damaged and undamaged cars, since for each crashed vehicle in a scene we provide its repaired counterpart at the same location.

A.1.2 Driving Simulator

CrashD was generated using a driving simulator developed by BeamNG [4], which includes a realistic physics engine, allowing for realistic damages. It offers a Python interface to setup the scenarios programmatically. Furthermore, it features a variety of sensors, including a LiDAR with customizable settings. Therefore, we equipped the ego vehicle with a LiDAR that imitates the one used in KITTI [1].

* The authors contributed equally.

^o Contact author: Alexander Lehner (alexander.lehner@tum.de).

Work partly sponsored by the German Federal Ministry for Economic Affairs and Energy (grant 19A19005B), VDA KI-Absicherung project.

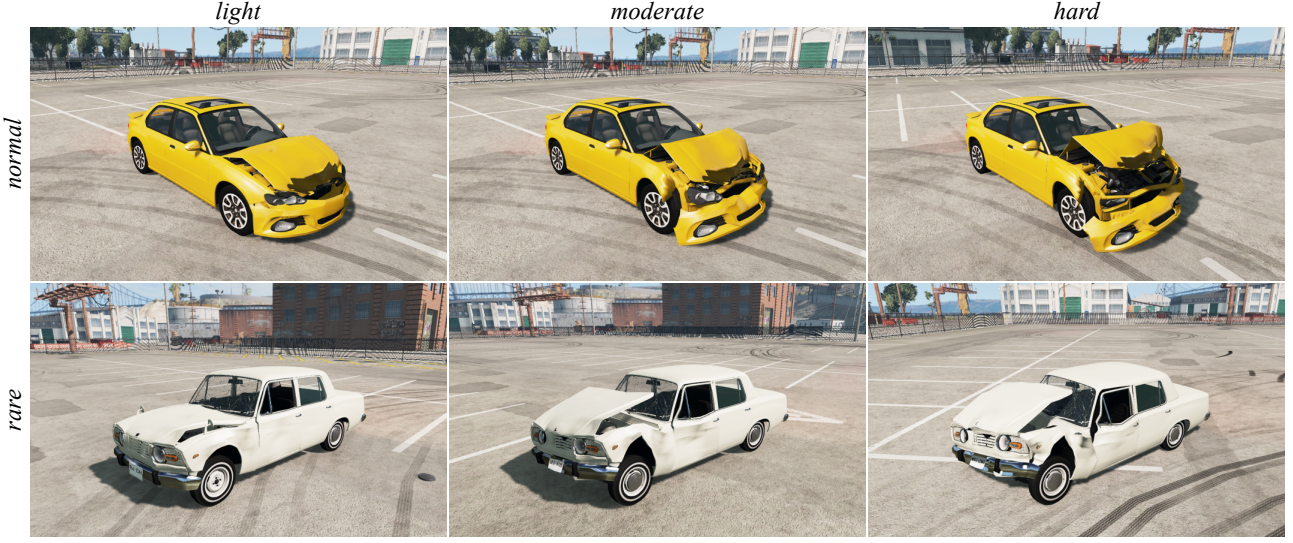


Figure 7. Comparison of *linear* damage intensities for *normal* and *rare* cars of CrashD. For each type of car, the accidents were created by the same hitting vehicle, coming from the same angle. It can be seen that the *hard* crash compromised the structure of the weaker *rare* car, while the *normal* car absorbed the impact differently, leaving the cabin unchanged.

A.1.3 Data Generation and Collection

We generated random accidents with random settings (e.g., hitting angle, distance, type of hitting car, type of hit car), and placed the cars randomly in the LiDAR scene. On each type of car (i.e., *normal* and *rare*), we applied 2 types of accidents (i.e., *linear* and *t-bone*), with 3 intensities each (i.e., *light*, *moderate* and *hard*). That results in 12 different categories of damaged cars and their 12 undamaged counterparts (i.e., *clean*), resulting in 24 categories overall. As the undamaged cars were placed at the exact same locations in the LiDAR scenes, they can be used as control group, to check the performance drop of a 3D detector when introducing the damages on the same cars.

We generated the accidents as follows. For each of the 12 categories of damages, we randomly selected 5 cars of the corresponding vehicle type (i.e., *normal*, *rare*), and 1 hitting vehicle. The hitting vehicle crashed into each of the 5 cars, getting repaired before each crash. We then repeated this process at least 64 times for each of the 12 categories, generating more than 3840 different accidents.

Furthermore, within each category, we used several random parameters, resulting in a high amount of possible damages. The intensity was determined by the distance from which the hitter starts, so the higher the distance, the higher the speed at which it will hit the target (i.e., one of the 5 cars). The effect of different intensities on the two types of cars for a *linear* crash can be seen in Figure 7. For each intensity type, there was a random variable determining a variation of the distance at which the hitter was placed. Then, the hitting angle and the side (i.e., front or back for

linear, and left or right for *t-bone*) were also randomized. Overall, this covered 360 degrees for each type of car and intensity.

Each batch of 5 cars, after being hit, was randomly placed in the LiDAR scene, such that the damaged area was visible from the sensor viewpoint, as shown in Figure 6. We considered a crash visible if the sensor was within 35 degrees from the hitting angle. This ensured that a car classified as damaged is represented by a deformed point cloud. Moreover, if the damaged part was not visible from the sensor, the car was discarded from the batch.

This was due to a series of reasons, resulting in the lack of control over the rotation of the damaged car within the LiDAR scene. In particular, BeamNG setup the simulator [4] such that if a vehicle is rotated programmatically, it gets automatically repaired. Plus, depending on the dynamics of a crash, a damaged vehicle could rotate following the impact. So, as we reduced the LiDAR scene to the front 180 degrees, we had to discard some cars to be sure that they were not classified as damaged if their impacted area was not visible. To avoid that crashes with a set of hitting angles could systematically not be placed in the scene, we randomly rotated the whole accident scenarios.

For each batch of 5 cars, we recorded 10 frames with the cars with visible damages (between 1 and 5), where we randomized the distance from the sensor, as well as the angle around it. Moreover, again to avoid that a vehicle is considered damaged if the affected area is not visible, we excluded occlusions considering only 25 angles around the sensor, and preventing two cars from occupying the same one. This resulted in 750 possible different locations in the



Figure 8. *Normal* cars of CrashD. These were classified as *normal* as they resemble the vast majority of cars on the road today in Germany, USA, and other locations where popular LiDAR datasets, such as KITTI [1] and Waymo [8], have been recorded.



Figure 9. *Rare* cars of CrashD. These were classified as *rare* as they complement the *normal* (i.e., common) cars shown in Figure 8. In particular, *rare* ones resemble old cars from various regions, and also include a wedge-shaped sports car. * indicates cars that cannot hit other vehicles (due to their low speed and weight), but can only be hit by others.

scene. With this setup, a given vehicle might be discarded in one frame if the angles from which its damage is visible are occupied by other cars, but might appear in a subsequent frame if it gets placed beforehand.

Furthermore, we put the objects only in the front, motivated by the front-facing setup of KITTI [1], thereby facilitating transfers from KITTI to the proposed CrashD. Towards this end, we positioned the vehicles from 10 to 40 meters away from the LiDAR, around its front 180 degrees. As shown in Figure 6, the scene features a large parking lot, where no object is located, other than the cars. We selected a totally empty parking lot (lacking poles, trees, or anything else), to fully focus on the task at hand, providing test data for evaluating the generalization capability of a method to different object shapes. Instead, having distracting elements

(e.g., trees) in the scene, could have led to a different kind of transfer evaluation (e.g., the ability of recognizing cars compared to other objects in the scene), which goes beyond the scope of this dataset. Nevertheless, in the main paper, as well as in additional results in this supplementary material, we also show a transfer from KITTI [1] to Waymo [8], which features real complex scenes, with trees and other objects, thereby challenging the 3D detector in a different way compared to transferring to the proposed CrashD.

A.1.4 Vehicles

The simulator offers a variety of fictional vehicles, which are shown in Figures 8, 9 and 10. In particular, the 12 *normal* cars used are shown in Figure 8, resembling the



Figure 10. These vehicles can only hit others and are not detectable objects, as they do not fit the KITTI [1] criteria for being a car, so they would not get recognized by a model transferred from KITTI.

vast majority of vehicles on the road today in the countries where common LiDAR datasets were recorded, such as Germany and USA, for KITTI [1] and Waymo [8] respectively. Figure 9 shows the 7 *rare* cars used for CrashD, including older cars from Europe, USA and Asia, as well as a wedge-shaped sports car. Among older cars, the simulator features different muscle cars, and also a very small car (at the top left of Figure 9).

The significant gap between the two types of cars can be seen by comparing the *normal* and *rare* vehicles in Figures 8 and 9 respectively. Specifically, considering the *normal* cars resemble those from KITTI and Waymo, the shapes of the *rare* ones are rather different, posing a substantial challenge for any LiDAR-based 3D object detector transferring on this dataset from those two others. Analogously, detecting the cars with the various deformations resulting from the accidents, which can be seen in Figure 7, pose a different, but also significant challenge for a detector trained on KITTI, Waymo, or a similar dataset.

Since the KITTI [1] *car* annotations do not include vans, trucks, pickups and busses, we excluded these from the detectable vehicles of CrashD. Nevertheless, these vehicles were part of the pool of hitting vehicles, and they are shown in Figure 10. Hitting vehicles also included all the ones shown in Figure 8, as well as those in Figure 9. However, we excluded the 2 cars marked with * due to their relatively low speed and weight, which would have not provided an accident as intense as those caused by the other vehicles, thereby altering the data distribution along the intensity types (i.e., *light*, *moderate*, *hard*). In spite of that, the 2 with * were part of the detectable vehicles.

A.1.5 Dataset Statistics

In total, the proposed CrashD includes 46936 cars, half of which are damaged and half are not, as the LiDAR scenes were repeated with and without damages. *Normal* cars are 23314, while *rare* ones are 23622, again half of each is damaged. 8124 cars were hit by *light* accidents, 7453 *moderate* and 7891 *hard*. 11530 were affected by a *linear* crash, while 11938 by a *t-bone*. Due to the vehicle placement in the LiDAR scene being dependent on the damage visibility, cars undergoing a *linear* crash were more likely to be included

from a frontal or rear perspective (including 3/4 views), while *t-bone* ones were only included from the sides.

A.2. Additional Implementation Details

Iterative gradient L2 attack For this attack [10] we minimize our adversarial loss $\mathcal{L}_{\text{adv}}$ constraining the deformation $\mathbf{m}$ for each point $\mathbf{p}$ with $\|\mathbf{m}\|_2 < \epsilon$, with $\epsilon = 30$ cm.

Chamfer attack For the Chamfer attack [3] we used the Chamfer distance to measure the gap between the original and perturbed point clouds, which is given by:

$$\mathcal{C}(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \|x - y\|_2 \quad (1)$$

for two sets X and Y . We perturb by minimizing:

$$\mathcal{L}_{\text{cha}} = \mathcal{L}_{\text{adv}} + \lambda \mathcal{C}(\mathbf{p} + \mathbf{m}, \mathbf{p}) \quad (2)$$

with λ set to 0.1 and the amount of deformation constrained by $\mathcal{C}(\mathbf{p} + \mathbf{m}, \mathbf{p}) < \epsilon$, with $\epsilon = 30$ cm. It should be noted that single deformations vectors could lead to perturbations larger than 30 cm, since what is bounded is the overall Chamfer distance and not single vectors. This attack led to only a small amount of perturbed points, but the ones that moved showed large displacements.

Adversarial removal For the removal attack we follow [12] and remove 10% of the *critical points* of an object. These are those input points that if removed, the prediction changes. We estimate them as those with the highest deformation magnitude from the iterative gradient L2 attack [10].

Adversarial generation We follow [10] adding 10% of the objects points. We initialize their location as that of the *critical points* (see removal). We then perform the iterative gradient L2 attack [10] solely on the added points. Thus shifting them to decrease the detection quality.

Transfer to Waymo To evaluate the transfers to Waymo [8], we used the standard KITTI evaluation. Therefore, the LiDAR scene was cut until 70 m in front of the ego vehicle and 40 m to both sides. We also lowered the whole point cloud and ground truth bounding boxes by 1.6 m, to match the KITTI coordinates and ground plane.

A.3. Additional Outdoor Quantitative Results

A.3.1 Transferability of the Vector Fields

Adv.aug.	PointP. [2]		Second [11]		Part-A ² [6]	
	AP	ASR	AP	ASR	AP	ASR
none	77.1	63.4	79.2	54.9	79.2	50.5
w/o $\mathcal{L}_{adv}$	76.4	60.0	77.2	52.5	79.3	47.4
[ours]	77.1	21.8	78.1	18.3	79.3	18.7

Table 4. *Moderate* AP and ASR ↓ across different models, showing transferability and efficacy of our deformations, on the validation set of KITTI. ASRs on Second and Part-A² are measured on vector fields trained on the defended PointPillars, to report the transferability. Adv.aug.: adversarial augmentation; w/o $\mathcal{L}_{adv}$: ours not learned.

Table 4 shows the high transferability of our adversarial deformations to other 3D object detectors. It can be seen that perturbations learned on PointPillars [2] are highly effective also on rather different architectures such as Second [11] and Part-A² [6], maintaining up to 86% ASR across the models. Table 4 reports also the benefit of our adversarial augmentation strategy against our deformations. The perturbed point clouds targeting PointPillars are effective also to defend the other models.

A.3.2 Robustness against noise

Method	-10%	-5%	0%	+5%	+10%
PointP. [2]	70.51	70.88	77.11	67.36	65.27
[ours]	71.45	71.75	77.13	69.57	65.86

Table 5. KITTI validation *moderate* AP under various % of removed and added points within the cars bounding boxes.

In Table 5 we report the performance of PointPillars [2] with and without our adversarial augmentation strategy. For this set of experiments, at inference time we randomly added and removed points within the cars bounding boxes according to the percentages reported in the table. Both models were the same as in the rest of this work, simply evaluated with this setup. Thanks to the improved generalization provided by our vector fields, the augmented model was more robust against such noise. Our augmentation acts as regularization during training, allowing the model to learn more meaningful features independent of specific points. This led to a constant gap between 5 and 10% removal. Conversely, randomly adding points is not realistic

from the sensor perspective, since occlusions and its physical properties are not respected. Due to this reason, both models suffered more when adding points, than removing.

A.3.3 Detailed transfer to CrashD

Evaluation by categories In Table 6, we show a detailed evaluation of the various 3D object detectors along the different sub-categories of the proposed CrashD, with various kinds of damages, different intensities and types of cars. Our adversarial augmentation strategy outperformed all detectors [2, 6, 11] across the board by a significant margin, especially on *rare* cars. In particular, with high intensity crashes (*hard*), the baselines [2, 6, 11] severely underperformed, reducing by half their APs on cars undergoing a *t-bone* accident. This can be due to the large point displacement introduced by the impacts, especially with weaker old cars. Conversely, our 3D-VField, as it was trained on sensor-aware deformations, was more robust against these damages, delivering a smaller decrease from the *clean* cars to their *crash* counterparts. Interestingly, *rare* vehicles were often more challenging to be detected than *normal crash* ones. This can be attributed to an accident typically affecting only a local region of a vehicle, leaving the rest of it untouched and detectable, compared to a rare design which has an impact on the whole object point cloud, making it in general harder to be recognized. Comparing the same cars with and without damages (*crash* and *clean*) shows that the former are significantly more difficult for every detector, due to the different resulting shapes. All detectors substantially benefited from our adversarial augmentations, despite training the vector fields solely against PointPillars [2]. The values also confirm the superiority of Part-A² [6] over the other 3D detectors, as seen in Table 1.

Correct and wrong detections on CrashD Table 7 reports a comparison of PointPillars [2] without and with our adversarial augmentations on CrashD, according to the number of true positives, false positives and false negatives, depending on the main categories of the proposed dataset, at different IoU thresholds. It can be seen that the baseline [2] had a strong tendency towards over-predicting the amount of objects in the scene, resulting in a high number of false positives. In fact, even with a low IoU threshold of 0.1, over 30% of the boxes predicted by the baseline did not match any car in the scene. At the same time, it completely ignored several cars, both damaged and undamaged, resulting in false negatives. On the other hand, as seen already in the main paper showing the APs, the proposed 3D-VField delivered a significantly better detection rate, vastly reducing the amount of false positives and negatives, despite being based on the same architecture and settings as the baseline [2].

→ CrashD			<i>normal, linear</i>			<i>normal, t-bone</i>			<i>rare, linear</i>			<i>rare, t-bone</i>		
			<i>light</i>	<i>mod.</i>	<i>hard</i>	<i>light</i>	<i>mod.</i>	<i>hard</i>	<i>light</i>	<i>mod.</i>	<i>hard</i>	<i>light</i>	<i>mod.</i>	<i>hard</i>
PointP. [2]	clean	baseline [2]	59.6	64.4	60.6	65.5	73.7	67.3	33.5	33.8	27.7	37.5	35.1	37.3
		3D-VF [ours]	61.8	64.2	62.0	72.4	76.7	70.6	39.6	41.1	35.0	49.6	47.4	47.7
PointP. [2]	crash	baseline [2]	46.5	33.8	28.6	57.9	54.9	40.2	26.7	22.9	15.4	31.2	23.3	15.4
		3D-VF [ours]	54.3	46.6	40.6	65.3	60.2	50.2	33.4	31.0	21.5	41.7	33.0	22.1
Second [11]	clean	baseline [11]	67.0	68.6	68.7	76.1	81.1	75.0	39.3	43.8	37.5	43.7	42.5	44.4
		3D-VF [ours]	71.3	75.4	73.1	79.3	82.4	77.7	40.9	47.5	41.5	52.8	49.2	53.0
Second [11]	crash	baseline [11]	60.1	46.4	43.0	72.0	65.6	53.3	36.1	37.8	28.8	40.1	31.4	22.9
		3D-VF [ours]	64.8	50.4	44.9	75.5	69.4	58.1	38.4	37.0	29.1	49.3	37.7	25.4
Part-A ² [6]	clean	baseline [6]	77.9	82.7	78.4	86.6	87.6	85.2	71.5	72.7	73.7	78.3	72.9	75.1
		3D-VF [ours]	85.6	86.2	86.0	91.3	93.2	90.5	80.0	81.6	79.8	83.7	79.3	82.2
Part-A ² [6]	crash	baseline [6]	71.1	58.6	49.3	79.7	64.3	56.5	61.7	55.5	49.0	67.0	48.6	32.2
		3D-VF [ours]	81.1	69.4	63.3	87.3	75.8	65.9	74.9	69.1	59.0	74.5	53.8	36.7

Table 6. Detailed AP comparison of PointPillars [2], Second [11], and Part-A² [6] trained on KITTI [1] and transferred to the proposed CrashD without any fine-tuning. The evaluation is shown according to the various accident types, and intensities, as well as the kinds of car. Baseline indicates the standard method, while [ours] shows the impact of our adversarial augmentation strategy.

→ CrashD		IoU 0.1		IoU 0.5		IoU 0.7	
		baseline [2]	3D-VF [ours]	baseline [2]	3D-VF [ours]	baseline [2]	3D-VF [ours]
<i>normal, clean</i>	TP ↑	11547	11651	11539	11638	8571	8894
	FP ↓	4069	419	4077	432	7045	3176
	FN ↓	110	6	118	19	3086	2763
<i>normal, crash</i>	TP ↑	11485	11642	11391	11562	6770	7620
	FP ↓	4550	772	4644	852	9265	4794
	FN ↓	172	15	266	95	4887	4037
<i>rare, clean</i>	TP ↑	11761	11805	11747	11790	6091	7528
	FP ↓	4700	316	4714	331	10370	4593
	FN ↓	50	6	64	21	5720	4283
<i>rare, crash</i>	TP ↑	11724	11804	11566	11680	4688	6011
	FP ↓	4742	590	4900	714	11778	6383
	FN ↓	87	7	245	131	7123	5800

Table 7. Impact of our adversarial augmentation on the main categories of the proposed CrashD according to true positives (TP), false positives (FP) and false negatives (FN) at different IoU thresholds. The models were based on PointPillars [2], trained on KITTI [1] and transferred to CrashD without any fine-tuning. For reference, the total amount of cars in CrashD is 46936.

A.3.4 Ablation Studies

Amount of deformed objects In Table 8 we report the effect of augmenting various amounts of objects during training. Specifically, augmenting more objects in each scene did not help generalization, as it made difficult to recognize standard objects. Augmenting all cars means the detector never learns a normal vehicle, making it rather hard to identify one at inference time. This can be seen in the AP drop

on KITTI [1] from augmenting half of the cars, to all of them. Instead, augmenting a single object allowed to retain the same AP on KITTI, while significantly improving it on the out-of-domain Waymo [8] and the proposed CrashD.

Grouping strategies In Table 9 we show the impact of varying amounts of learned vector fields on the ASR, according to different distinguishing criteria. We compare the chosen relative rotation (Section 3.2) with selecting by dis-

# augm. objects	KITTI	→ W.	→ CrashD	
	<i>mod.</i>		<i>n.,clean</i>	<i>r.,crash</i>
[ours] 1 obj.	77.13	44.61	67.95	30.37
[ours] 50% obj.	76.31	39.60	53.99	23.63
[ours] 100% obj.	59.30	32.84	38.29	14.83

Table 8. Models trained on KITTI, augmented with our adversarial technique. In each row, the amount of objects augmented at training time in each scene changes. The chosen number of augmented objects was 1. *mod.*: moderate difficulty; →: transfer without any fine-tuning; W.: Waymo; *n.*: normal; *r.*: rare.

tance of the object to the sensor or number of object points. Relative rotation delivered superior ASR, as it favors the mutual alignment between neighboring vectors. In contrast, less vector fields (i.e., 1 and 6) or different criteria resulted in contrasting vectors, reducing the object deformation.

Aggregation strategies Table 10 shows the effect of different aggregation strategies of vectors when applying the deformations on the cars of KITTI [1]. It can be seen how the different amount of groupings (G) and neighboring vectors (k) considered for each point shift affected the adversarial performance of the method (ASR). In general, all deformations in the table were restricted to a maximum of $\epsilon = 30$ cm. The amount of learned vector fields G had an impact on the ASR of each aggregation strategy. For example, sum was more effective with 12 G than 1 G , since the vectors of the 12 fields were better aligned to each other than those of the single field (Section 4.2), so summing them increased the deformation magnitude. In fact, the high ASR of sum with 12 G , was due to larger perturbations.

Grouping	1-ASR	6-ASR	12-ASR	18-ASR
distance	55.1	56.2	57.3	57.6
nr. points	55.1	56.9	56.0	57.1
rel. rotation	55.1	59.2	63.4	63.7

Table 9. ASR $\uparrow$ on the validation set of KITTI for different grouping strategies and amount of vector fields.

Grid step size In Table 11 we show the impact of different step sizes t of the vector field grid. A larger step size, results in a coarser grid, which in turn means less vectors for each field. Intuitively, with more vectors, each would be more specific for a given point shift, but less generalizable to others. So, each vector would overfit to its training points. There is in fact a trade-off between the amount of vectors and the generalizability of the learned vector field, as seen in Table 2. That can be seen by the ASR, as the vectors were learned on the training set of KITTI [1], and applied to

k	Aggregation					
	-	sum		average		distance
	1	2	3	2	3	2 3
$G = 1$	46.3	44.4	33.4	45.4	52.0	50.3 47.0
$G = 12$	59.6	76.5	80.3	61.9	62.4	63.4 59.6

Table 10. ASR $\uparrow$ on the validation set of KITTI [1] for different aggregation strategies and number of neighbors (k) involved in each deformation, for both number of groups $G = 1$ and $G = 12$. All configurations are based on PointPillars [2].

its validation set, on which the values are reported. Down-scaling t from 20 to 5 cm, significantly reduced the ASR. Conversely, increasing t to 30 cm worsened their generalization. Therefore, $t = 20$ cm was chosen as the grid step size, offering a good trade-off between the vector specificity and generalizability, as shown by the ASR.

Step size	5 cm	10 cm	20 cm	30 cm
ASR $\uparrow$	44.1	46.3	53.0	49.6

Table 11. ASR $\uparrow$ on the validation set of KITTI [1] for different step sizes of the vector field grid. A smaller step size increases the amount of vectors. All configurations are based on PointPillars [2], with $G = 1$.

A.4. Additional Outdoor Qualitative Results

In this section we provide qualitative results of the learned deformations.

A.4.1 Deformations on KITTI

Figure 11 shows a comparison of the deformations applied by each method to a set of cars from KITTI [1]. We included related works, such as the Chamfer attack [3] and the iterative gradient L2 approach [10], as well as variations of the proposed 3D-VField. The Chamfer attack [3] shifted some points far away while many remained close to the original location, resulting in an almost perfect ASR. However, this came at the cost of rather obvious perturbations. The iterative gradient L2 [10] method also achieved a highly effective ASR (Table 1), but with significantly less evident deformations. As expected from the high ASR (Table 3), our unconstrained (unleashed) method delivered substantially perturbed objects, even more distorted than those produced by the Chamfer attack. Applying the ray constraint allowed for less perturbed (and less effective ASR), but more recognizable objects. It can be seen how this constraint alone impacts the realism of the deformations, by

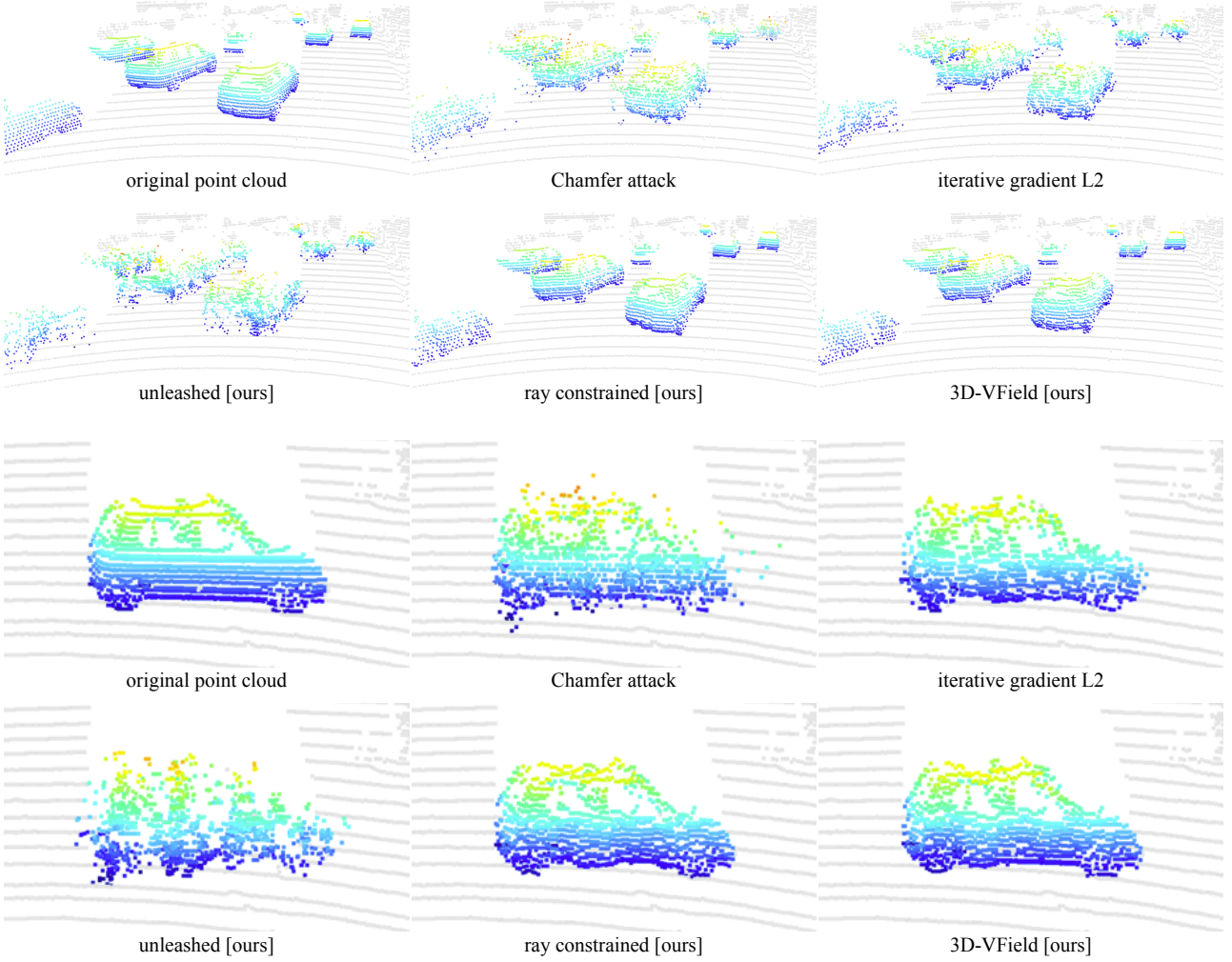


Figure 11. Comparison of adversarial perturbations on a set of cars from two different point clouds of KITTI [1]. The effect of the Chamfer attack [3], the iterative gradient L2 [10], and multiple variations of our approach are shown. It can be seen that our 3D-VField preserves the shape of the original point cloud better than the other approaches.

comparing it to the unleashed version. Moreover, aggregating neighboring vectors via distance weighting in our full approach (Section 3.2), further improved the resemblance of the object to the original point cloud. Although the difference is subtle, this can be appreciated comparing the rear wheel, the floor, and the windows of the car in the bottom half of Figure 11. Thanks to the realism and the smooth alterations of the points visible in the figure, training with our deformations allowed for superior transfer performance to challenging out-of-domain data (Table 1).

Figure 12 shows the deformations learned by our method. It can be seen that only local areas are affected, and the cars preserved their overall shapes with smoothly deformed parts.

Figure 13 shows the effect of each vector of the adver-

sarial field to the ASR. It can be seen that the most affected was the front bumper, which can easily be deformed with an accident. The side of the car is mostly unaffected, prob-



Figure 12. Color-coded deformations in cm learned by the proposed method. The perturbation does not affect every point, its magnitude is relatively low, and local smoothness is preserved.

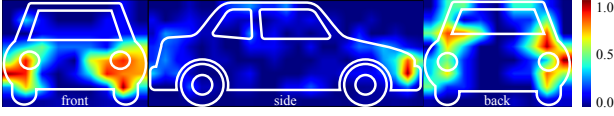


Figure 13. Color-coded contribution of each vector to the ASR, in percentage.

ably due to the relatively limited amount of vehicles visible from the side in KITTI. Interestingly, the model has learned to avoid the areas without points (e.g., the windows).

A.5. Results on Indoor Data

A.5.1 Experimental setup

In these experiments we used the SUN RGB-D dataset [7], which posed a completely new set of challenges compared to the three driving datasets. SUN RGB-D contains indoor furniture objects captured by depth cameras such as time-of-flight (ToF), as opposed to driving scenes captured by a LiDAR. We trained on all 10 classes, but we selected one at a time for learning our vector fields. In particular, we report on the classes *bed*, *sofa* and the highly diverse *chair*, as they are the ones where deformations are more plausible compared to others (e.g., *table*). In this setting, we apply our method on a VoteNet [5] architecture. Moreover, we followed the same setup as for the outdoor experiments, except that we reduced the maximum deformation ϵ to 10 cm, making it more plausible in indoor settings.

A.5.2 Quantitative Results

Adv.aug.	<i>beds</i>		<i>sofas</i>		<i>chairs</i>	
	AP	ASR	AP	ASR	AP	ASR
none	85.6	49.7	67.4	70.6	77.4	70.9
w/o $\mathcal{L}_{adv}$	85.2	41.1	67.5	65.4	76.9	62.1
[ours]	86.0	19.7	68.5	34.8	77.5	39.6

Table 12. AP and ASR $\downarrow$ on the validation set of SUN RGB-D [7], with a VoteNet [5] architecture. Adv.aug.: adversarial augmentation; w/o $\mathcal{L}_{adv}$: ours not learned.

Table 12 shows the wide applicability of our deformation and augmentation strategies when applied to point clouds from depth sensors capturing furniture objects from SUN RGB-D [7]. Shifting the points with our 3D-VField produced a strong ASR against the not adversarially augmented models (none), especially on *sofas* and *chairs*. Using the deformations as augmentation even improved the AP on the validation set, confirming the benefit of our techniques towards the generalization to unseen data, despite the rather

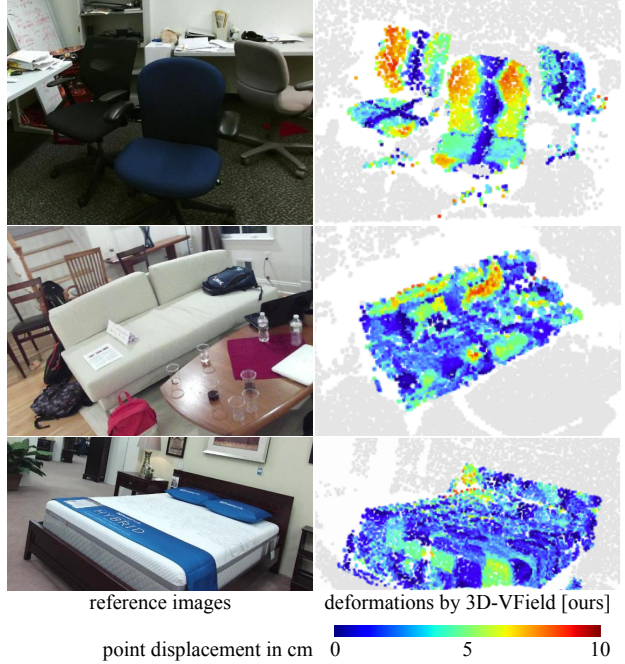


Figure 14. Color-coded deformations applied by the proposed 3D-VField on various objects of the SUN RGB-D dataset [7]. The color corresponds to the shift of each point in centimeters, limited to a maximum of 10 cm. Adversarial deformations learned against VoteNet [5].

different setting, sensor, objects, and architecture. Furthermore, defending with our adversarial augmentations significantly reduced the ASR, showing the gained robustness against deformed objects.

A.5.3 Qualitative Results

In Figure 14 we show the deformations learned by our method against VoteNet [5] on three different categories of objects from SUN RGB-D [7], namely *chairs*, *sofas*, and *beds*. It can be seen that the overall shape of each object is preserved, with minor perturbations applied. In this indoor setting, such alterations could resemble the presence of pillows, a blanket, or simply a different design of the object.

References

- [1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 3, 4, 6, 7, 8
- [2] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12697–12705, 2019. 5, 6, 7

- [3] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3D point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 88–104. Springer, 2020. 4, 7, 8
- [4] Pascale Maul, Marc Mueller, Fabian Enkler, Eva Pigova, Thomas Fischer, and Lefteris Stamatogiannakis. BeamNG.tech technical paper, 2021. 1, 2
- [5] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 9
- [6] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5, 6
- [7] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 9
- [8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 3, 4, 6
- [9] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4627–4635, 2021. 1
- [10] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3D adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9128–9136, 2019. 4, 7, 8
- [11] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, Oct. 2018. 5, 6
- [12] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019. 4

B.2 Appendix: Segmenting Known and Unknown Objects and Regions (ICCV 2023)

This section includes the supplementary material associated with [85], presented and discussed in Section 5.2.3.

Supplementary Material of:

**Segmenting Known Objects and Unseen Unknowns
without Prior Knowledge**

Stefano Gasperini^{1,2} Alvaro Marcos-Ramiro² Michael Schmidt²
Nassir Navab¹ Benjamin Busam¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² BMW Group ³ Google

IEEE/CVF International Conference on Computer Vision (ICCV), 2023

Published version: https://openaccess.thecvf.com/content/ICCV2023/supplemental/Gasperini_Segmenting_Known_Objects_ICCV_2023_supplemental.pdf.

Supplementary Material of: Segmenting Known Objects and Unseen Unknowns without Prior Knowledge

Stefano Gasperini^{1,2,◦} Alvaro Marcos-Ramiro² Michael Schmidt²
Nassir Navab¹ Benjamin Busam¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² BMW Group ³ Google

A. Supplementary Material

In this appendix, we include further details and results. Specifically, Sections A.1, A.2 and A.3 provide deeper insights on the proposed setting, the method, and the experimental setup, respectively, while Sections A.4 and A.5 contain more results, both quantitative and qualitative.

A.1. Additional Details on the Setting

In this section, we describe the benefits of the proposed holistic segmentation setting in greater detail, considering both the impact on downstream tasks and the differences with other perception tasks addressing unknown objects.

The proposed holistic segmentation setting aims to segment any unseen, unknown objects without prior knowledge about the unknowns while segmenting known areas. In this context, "unseen unknowns" means any object of any category outside the known classes learned during training, such as the sheep in Figure 6 for a method trained on, e.g., Cityscapes [7], as well as unidentified and distorted parts following a car accident.

A.1.1 Motivation

The importance of identifying unseen unknowns arises from safety-critical scenarios, such as autonomous driving, where ignoring them can lead to dangerous consequences when simply using the predicted segments for downstream tasks, e.g., path planning. This is shown in the top right of Figure 6. Since even large-scale datasets are limited representations of the real world, there will always be corner cases and long tail samples which are problematic for standard models [15]. Therefore, it is crucial to identify these cases and then deal with them safely via downstream tasks.

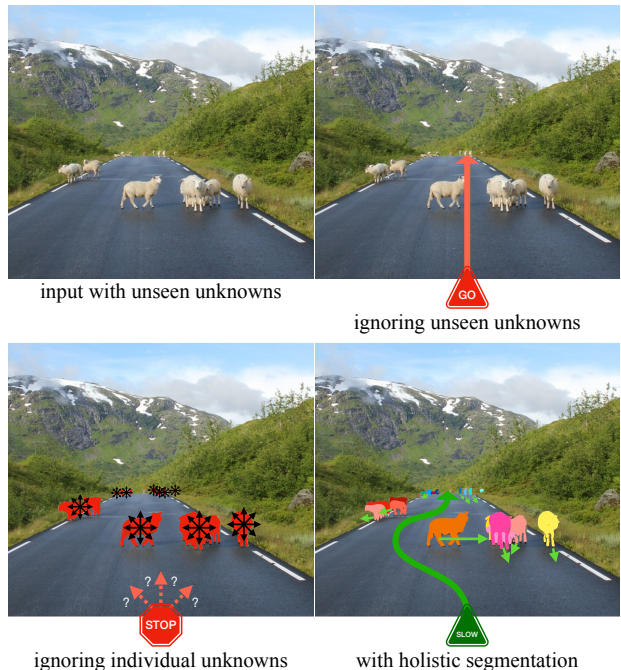


Figure 6. Motivation diagram for identifying unseen, unknown objects on a sample of [3] considering path planning as a downstream task, and hypothesizing that *sheep* is not part of the training data (i.e., unseen unknown). Shown segments are not predictions. State-of-the-art approaches dangerously ignore unknowns (top right) [5]. OOD segmentation does not identify instances of unknowns (bottom right) [14], making it difficult for downstream tasks as the unknowns cannot be tracked, and their trajectory cannot be predicted. The proposed setting (bottom right) identifies individual unseen unknowns, leading to a safe path.

OOD segmentation [14], i.e., segmenting unknown areas as a whole and not identifying individual instances of unknowns, flags the presence of something unknown in the input. In the context of downstream tasks, such as path planning, a single OOD segment (bottom left in the figure) could trigger an alert state, leading to a potential stop, which is a

[◦] This work was conducted while working at BMW Group.
Contact author: Stefano Gasperini (stefano.gasperini@tum.de).

Setting	Data Assumptions	Identifiable Objects	Not Identifiable Objects
open-set panoptic segm. [13, 22]	unknowns are already in the training data, within <i>void</i> areas	known and unlabeled objects present in the training data as <i>void</i>	categories outside of the training data [13]
open-vocabulary, zero-shot [23, 12]	the underlying language model knows about every unknown	objects known by the language model [19]	categories outside of the training data of the language model [19]
holistic segmentation [ours]	none	any known and unknown (i.e., unseen) object	none

Table 4. Comparison of tasks and settings dealing with instances of unknown objects. The second column (Data Assumptions) is related to unknowns. The 2 rightmost columns represent the objects that are theoretically identifiable or not, given the setting.

safe state. However, given that OOD segmentation does not separate unknown objects into instances, once found, it is unclear whether they are moving or static, which means that it would be difficult for path planning. Instead, by segmenting instances of unseen unknowns (bottom right in the figure), holistic segmentation allows tracking unseen objects and estimating their trajectory, leading to a safe path. This motivates the instance segmentation of unknowns, which brings benefits similar to those of instance segmentation compared to semantic segmentation for known objects.

Also critical is the ability to deal with any unseen, unknown object category and not be restricted to a limited subset of them. This is of utmost importance to address the wide variability of objects and scenarios encountered in the real world. While previous settings focused on re-identifying already-seen objects [13, 23], we design holistic segmentation specifically to address any unseen category.

A.1.2 Comparison with Other Settings

As shown in Table 4, compared with other tasks and settings also dealing with unknown objects, the proposed holistic segmentation makes no assumptions about the unknown objects, allowing one to segment any objects. Instead, zero-shot and open-vocabulary approaches assume that text descriptions of unknown objects are available [12, 23]. Open-set panoptic segmentation methods assume unknowns are confined within *void* regions at training and test time [22, 13]. In the latter case, *void* may not be available or not sufficiently large and diverse (as in Cityscapes [7]), depending on the training data. Due to their construction, both of these setups inherently restrict the pool of recognizable objects to those for which text descriptions are available through a vision-language model (open-vocabulary) or to those present within their own training set (open-set panoptic).

For example, for the scene in Figure 6, because of its setup, EOPSN [13] cannot identify any *sheep* unless a vast

amount of images containing *sheep* is part of its training data (with *sheep* being labeled as *void*, or directly as a dedicated class *sheep*). Open-vocabulary methods would rely on the fact that a language model [19] already knows about *sheep* to be able to identify them in the image. While the concept of *sheep* is relatively simple and could be assumed to be known by a large language model, there is no guarantee that such a model would know about every possible object and scene that can be encountered in real life (e.g., unidentified pieces on the road following a car accident), meaning that open-vocabulary approaches cannot deal with long tail samples from the distribution of the natural world, simply because their language model cannot process them.

Again, given that datasets include by definition only a fraction of the diversity of the world [15], also datasets to test the ability of a model to identify unknowns are limited [18, 3, 1], containing only a small amount of the possible objects and situations that can be encountered in real life. Therefore, to operate reliably in real unconstrained scenarios, it is of utmost importance not to have limitations on the types of recognizable objects, which should go beyond those found in existing datasets. Instead, relying on a language model to identify unknowns is equivalent to shifting the unknown problem to a different model. As shown with CLIP by Radford et al. [19], large language models also have issues with OOD samples. For example, unidentified broken car parts lying on the ground after an accident would be difficult to describe, so it would be problematic for language models. Thus, when given inputs that are unseen and unknown to the underlying language model, open-vocabulary and zero-shot methods would fail to identify the unknown objects. Furthermore, existing open-set panoptic works rely on the presence of unknowns (intended as unlabeled) directly in the training data through the *void* class. This highlights the need for a new and unconstrained solution.

For these reasons, the critical differences between the proposed holistic segmentation setting and previous tasks

are that holistic segmentation is not constrained in terms of the types of unknown objects that are identifiable and that holistic segmentation does not assume the presence of unknowns in the training data, thereby segmenting **any unseen, unknown object without any prior knowledge about unknowns**. Limited by design by either the unknowns that are known to the underlying language model (e.g., open-vocabulary) or the unknowns that are directly present in the training data (e.g., open-set panoptic segmentation), previous tasks do not enable the identification of any instance of unknowns and rely on prior knowledge about unknowns and their data distribution (e.g., through CLIP [19] or by learning *void*).

A.2. Additional Details on the Method

Loss functions As described in Section 4.3, the proposed method is trained with a combination of losses: a semantic loss $\mathcal{L}_s$, an object detection loss $\mathcal{L}_o$, a prototype loss $\mathcal{L}_p$, and a discriminative loss $\mathcal{L}_d$. The discriminative loss is aimed at learning meaningful embeddings. It is composed of three different terms [8], namely variance $\mathcal{L}_{va}$ to attract elements towards the mean, distance $\mathcal{L}_{di}$ to push away different groups, and regularization $\mathcal{L}_{re}$ to prevent the divergence of clusters from the origin:

$$\begin{aligned}\mathcal{L}_d &= \lambda_{41}\mathcal{L}_{va} + \lambda_{42}\mathcal{L}_{di} + \lambda_{43}\mathcal{L}_{re} \\ \mathcal{L}_{va} &= \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \frac{1}{N_\omega} \sum_{a=1}^{N_\omega} [\|\mu_\omega - \phi_a\| - \delta_v]_+^2 \\ \mathcal{L}_{di} &= \frac{1}{|\Omega|(|\Omega|-1)} \sum_{\omega_A \in \Omega} \sum_{\omega_B \in \Omega} [2\delta_d - \|\mu_{\omega_A} - \mu_{\omega_B}\|]_+^2 \\ \mathcal{L}_{re} &= \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \|\mu_\omega\|\end{aligned}\quad (1)$$

where: $|\Omega|$ is the number of prototypes, N_ω is the number of embeddings associated to the prototype ω , μ_ω is the mean embedding of the cluster related to ω , $\|\cdot\|$ is the L2 distance, $[x]_+ = \max(0, x)$ is the hinge (i.e., until which threshold the terms are active [8]), $\omega_A \neq \omega_B$, and we follow [8] for the hyperparameters, e.g., $\lambda_{41} = \lambda_{42} = 1$ and $\lambda_{43} = 0.001$.

Clustering unseen unknowns As described in Section 4.2, we use DBSCAN [9] to cluster the embeddings of unknown regions into individual unknown objects. Specifically, DBSCAN has multiple advantages: it does not need the number of clusters as input (which is unknown in our case), it is effective and very fast, has a low memory footprint, and distinguishes outliers (Table 3 shows the impact of this feature with A3-A4). Although other traditional clustering methods (e.g., Mean Shift, Affinity Propagation, Birch) are theoretically applicable in our setting, they come with drawbacks (e.g., have high memory requirements, do not output outliers, are significantly slower, or tend to deliver sub-optimal results). On the other hand, popular approaches that require the number of clusters as input can-

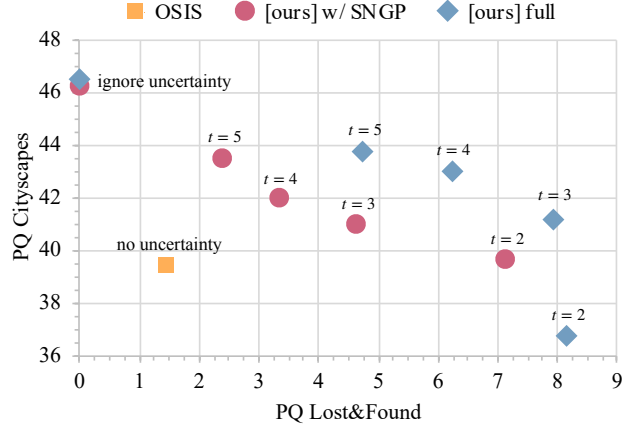


Figure 7. Trade-off between known (i.e., Cityscapes [7] validation set) and unknown (i.e., Lost&Found [18] test set) performance introduced by OSIS [22], compared to our approach, both using SNGP [17] and our improved DPN (i.e., [ours] full, in blue). The different data points are obtained by varying the parameter t .

not be applied in our settings (e.g., K-Means). Hence, DBSCAN was selected.

A.3. Additional Details on the Experimental Setup

Clustering parameters DBSCAN requires two parameters: $minPts$ (number of points in a neighborhood to count as a core point) and ϵ (maximum neighborhood size). To find such parameters, we trained a model (i.e., on Cityscapes [7] or MS COCO [16]), then selected ($minPts$, ϵ) with a simple grid search maximizing PQ on a random subset of the known dataset (i.e., without unknowns). Towards this end, we formed instances as follows: ignoring the detection output (i.e., using only the embeddings) and determining their class via majority voting from the semantic output. In particular, when finding ($minPts$, ϵ), this means treating the embeddings of knowns as if they were unknowns (apart from their semantic class), assuming that the model treats them similarly. It is essential to consider that the parameters were selected on the known objects (i.e., from Cityscapes or COCO), despite DBSCAN being used only for separating unknowns (i.e., in the Lost&Found [18] dataset or the held out classes of COCO). We did this to maintain the unknowns completely unseen (i.e., only as test set), as in real scenarios.

Comparisons with previous works As described in Section 5.1, we compared our uncertainty-based solution with prior works learning the *void* class and tested various uncertainty estimators within our proposed framework. The following paragraphs further detail how these other methods were trained.

Following our setup of training on Cityscapes [7] and transferring to Lost&Found [18] without any fine-tuning, prior works addressing open-set panoptic segmentation

Uncertainty method	L&F (<i>unseen</i>)		open CS mIoU
	AP	FPR ₉₅ ↓	
<i>softmax</i>	16.72	22.88	71.77
MC Dropout [10]	11.22	13.94	68.31
DML [2]	3.14	83.04	69.86
DUQ [21]	5.43	26.64	68.78
DPN [20]	5.43	19.79	66.99
SML [14]	16.91	51.67	70.69
SNGP [17]	22.70	12.02	70.68
improved DPN [ours]	25.44	19.10	70.10

Table 5. Comparison of open-set semantic segmentation on Lost&Found [18] test set of uncertainty estimators based on DeepLabV3+ [4] and trained only on Cityscapes (CS) [7].

(i.e., OSIS [22] and EOSPN [13]) were trained by learning the *void* class of Cityscapes, unlike our U3HS. This unlabeled class comprises all pixels that do not fulfill the requirements to be part of one of the standard 19 annotated classes. Some of these *void* pixels are systematic, e.g., the back side of traffic signs and street lights (excluding poles). By exploiting the variability within *void*, the models learn the extra class decision boundary as a fall-back covering anything far from the other classes. To do so, OSIS learns a constant U representing such boundary. In particular, we adapted OSIS from LiDAR point clouds to RGB images, applying it to each pixel instead of point and changing architecture accordingly. As for ours and all other models in this work, we used a ResNet50 [11] as backbone and decoders following the structure of DeepLabV3+ [4]. Moreover, to keep the GPU memory low, we used the same $F = 8$ for the embedding size as in our U3HS. For the experiments on MS COCO [16], we followed the K=5% setup of EOSPN [13], turning 4 classes into *void* to facilitate prior works learning on *void*, by ensuring a diverse distribution of its pixels, as they now cover a diverse set of classes (e.g., *pizza* and *car*).

For the other uncertainty estimation approaches evaluated in this work, we used the authors descriptions and implementations, adapting [21, 17] from image classification to semantic and panoptic segmentation. For DML [2], we used the authors best hyperparameters, therefore a variance loss weight $\gamma_{VL} = 0.01$, and weights $\beta = 20$ and $\gamma = 0.6$. For SML [14], we did not employ the boundary suppression, as it did not improve the results. This might be due to Lost&Found [18] being annotated only for the OOD objects and a coarse road segment. For DUQ [21], we used an embedding dimension of $m = 8$, due to constrained training resources, same as our $F = 8$. Then, we used length scale $\sigma^2 = 0.3$ and exponential smoothing factor $\gamma = 0.999$. For SNGP [17], we again used an embedding dimension $D = 8$ (due to the limited training resources), no layer norm for the embeddings, an exponential smoothing factor $\gamma = 0.99$ for

Configuration			L&F (<i>unseen</i>)		open CS mIoU
Ref.	Activ.F.	KL	AP	FPR ₉₅ ↓	
[20]	<i>exp</i>	yes	5.43	19.89	66.99
[ours]	<i>softplus</i>	yes	3.43	25.97	64.36
[ours]	<i>softplus</i>	no	25.44	19.10	70.10

Table 6. Ablation study on uncertainty estimates for open-set semantic segmentation. Models trained only on Cityscapes [7].

updating Σ and 50 samples for Monte Carlo averaging to estimate the uncertainty.

MS COCO As described in the main paper, given that there is no official set of unknown classes for MS COCO [16], we treat as unknown the least frequent 20% known classes. These classes are: *baseball bat*, *bear*, *fire hydrant*, *frisbee*, *hair drier*, *hot dog*, *keyboard*, *microwave*, *mouse*, *parking meter*, *refrigerator*, *scissors*, *snowboard*, *stop sign*, *toaster*, and *toothbrush*. We held out all training samples where any of these 16 classes appeared, such that they were completely unseen to the models.

Ablation study for holistic segmentation With reference to Table 3, A1 is our baseline, which was built upon OSIS [22]. As OSIS, A1 included learned instance-aware embeddings, but unlike OSIS, it featured a semantic decoder delivering semantic segmentation and uncertainty estimates based on the semantic output via our improved DPN. Moreover, as for all our models, A1 did not learn the *void* class (unlike OSIS). A2 featured the relaxed score for the embedding association (described in Section 4.1), which lets the variance be indirectly controlled by the final task (i.e., the loss $\mathcal{L}_p$, Section 4.3). Unlike A1 and A2, which had a shared head between embeddings and prototypes (i.e., as in OSIS), A3 introduced a dedicated prototype head. In practice, this meant having more layers fully dedicated to the embeddings and the prototypes separately instead of sharing the computation until a later stage. Therefore, this allowed for more expressive and purposed features. A4 did not reassign to the known classes the outliers obtained from clustering unknowns via DBSCAN. Therefore, these pixels were kept unknown and shared the same instance ID. A5 did not perform majority voting (Section 4.1). This meant directly assigning the semantic classes predicted by the semantic branch to all known instance pixels instead of enforcing coherence within an instance. This caused the instances to be fragmented according to how many semantic classes they contained, decreasing RQ. Finally, A6 predicted the semantic classes for *stuff* areas directly from the semantic prediction branch instead of matching the embeddings with *stuff* prototypes as in A1-A5 (Section 4.1).

A.4. Additional Quantitative Results

Trade-off between known and unknown Figure 7 shows the trade-off between the performance on known and

ResNet depth	F	PQ	RQ	SQ
18	2	33.0	42.3	77.9
18	4	38.9	49.8	78.0
18	8	41.3	52.7	78.3
18	16	42.3	53.8	78.7
18	32	<u>42.1</u>	<u>53.6</u>	<u>78.5</u>
50	8	47.7	60.4	79.0

Table 7. Different embedding dimensions F on closed-set panoptic segmentation on the validation set of Cityscapes [7]. The first column indicates the depth of the ResNet [11] backbone used (i.e., 18 for ResNet18).

unknown for our framework, both with SNGP [17] and our improved DPN, compared to that of OSIS [22]. The different data points were extracted by evaluating the outputs at different thresholds t , namely [2, ..., 5], and ignoring the uncertainty estimates entirely (i.e., closed-set, reported where PQ Lost&Found is 0). The hyperparameter t directly affects how high the uncertainty estimates must be for their associated pixels to be considered unknown. This has an impact on the performance on open-set Cityscapes [7] and Lost&Found [18], since changing in output what is considered unknown alters what is regarded as in-domain (i.e., known) as well. OSIS [22] does not have such a hyperparameter as it considers unknown everything predicted as *void*. Overall, it can be seen that our proposed framework offers a better trade-off in both configurations (red and blue) than that of OSIS [22] (yellow). Furthermore, using our full approach (i.e., our framework with our improved DPN) typically gave the best trade-off between known and unknown without compromising the metrics too much (blue).

Unknowns in semantic segmentation In Table 5, we compare the ability of a wide variety of uncertainty estimators (i.e., [20, 21, 17, 14, 2], and MC Dropout with 25 runs [10]) to find unknowns in a semantic setting on Lost&Found [18], after training on Cityscapes [7]. This meant retraining all methods under the same conditions while also extending DPN [20], DUQ [21], and SNGP [17] to semantic segmentation. Semantic models (Tables 5 and 6) used smaller crops sized 512×256 compared to the other experiments. For uncertainty estimation, we evaluated the ability to identify unknowns reporting the AP on the unknown class [18], as well as the false positive rate at the recall 95 (FPR₉₅). For semantic segmentation on Cityscapes, we computed the mIoU. As seen in Table 3, DUQ [21] and DPN [20] performed worse than SNGP [17]. MC Dropout [10] underperformed *softmax*, probably due to the contrasting opinions from 25 forward passes. Our method was the best at finding unknowns (AP) with high-quality uncertainty estimates (FPR₉₅). Table 5 also reports the mIoU on Cityscapes (CS), showing that all methods introduce a trade-off between OOD and in-domain outputs,

Method	Clustering	PQ	RQ	SQ
U3HS [ours]	Mean Shift	2.71	3.91	69.22
U3HS [ours]	DBSCAN	9.36	14.83	63.14

Table 8. Transfer from Cityscapes [7] to Lost&Found-300 [18] test set (i.e., on the first 300 samples, see Section A.4). DBSCAN (our choice) is compared to Mean Shift to cluster the embeddings of unknown areas.

as overestimating the uncertainty decreases the in-domain mIoU. Balancing these two complementary aspects is not trivial, with our approach and SNGP managing it best.

Ablation on uncertainty estimation Table 6 compares the DPN [20] we adapted from image classification to semantic segmentation with our extension. Our improvements were oriented to simplify the training process and help convergence. First we applied the *softplus* activation function to the last semantic layer, instead of *exp* as in DPN [20]. We chose *softplus* because it grows slower than *exp* and it is smooth, differentiable everywhere, and monotonic. This significantly improved the training stability at the cost of a reduced quality of the uncertainty estimates. Finally, due to the complexity of modeling the target distribution in our setting, omitting the KL term used by DPN [20] further stabilized training and boosted the performance on all metrics.

Impact of embedding size and architecture Table 7 shows the effect of different embeddings dimensions F on a smaller ResNet18 [11]. In the rest of this work, all experiments used $F = 8$ and ResNet50, as in the last line of the table, due to constrained training resources. The embedding dimension directly affects the learning capability of the model. Since the instance-aware embeddings are a critical part of the output, a smaller F is linked to inexpressive embeddings that cannot be as discriminative as those from a larger F . Therefore, increasing F improved all metrics except for the larger $F = 32$. This can be attributed to the small ResNet18 backbone being already saturated at $F = 16$, unable to extract rich and detailed features for the larger embeddings to exploit. With a larger model (e.g., ResNet101), even higher embedding dimensions F might be beneficial. Table 7 shows that our proposed approach, given less constrained resources, could deliver better results when using an embedding dimension higher than the $F = 8$ employed across this work. The table also shows the comparison between ResNet18 and ResNet50, with the latter delivering over 15% higher PQ at the same $F = 8$. This shows how our proposed approach would perform with a larger backbone.

Impact of the clustering method In Table 8 we compare two popular clustering methods within our U3HS framework, namely DBSCAN and Mean Shift [6]. Due to the very high computation effort and memory required by Mean Shift, we opted for the following setup for this ex-

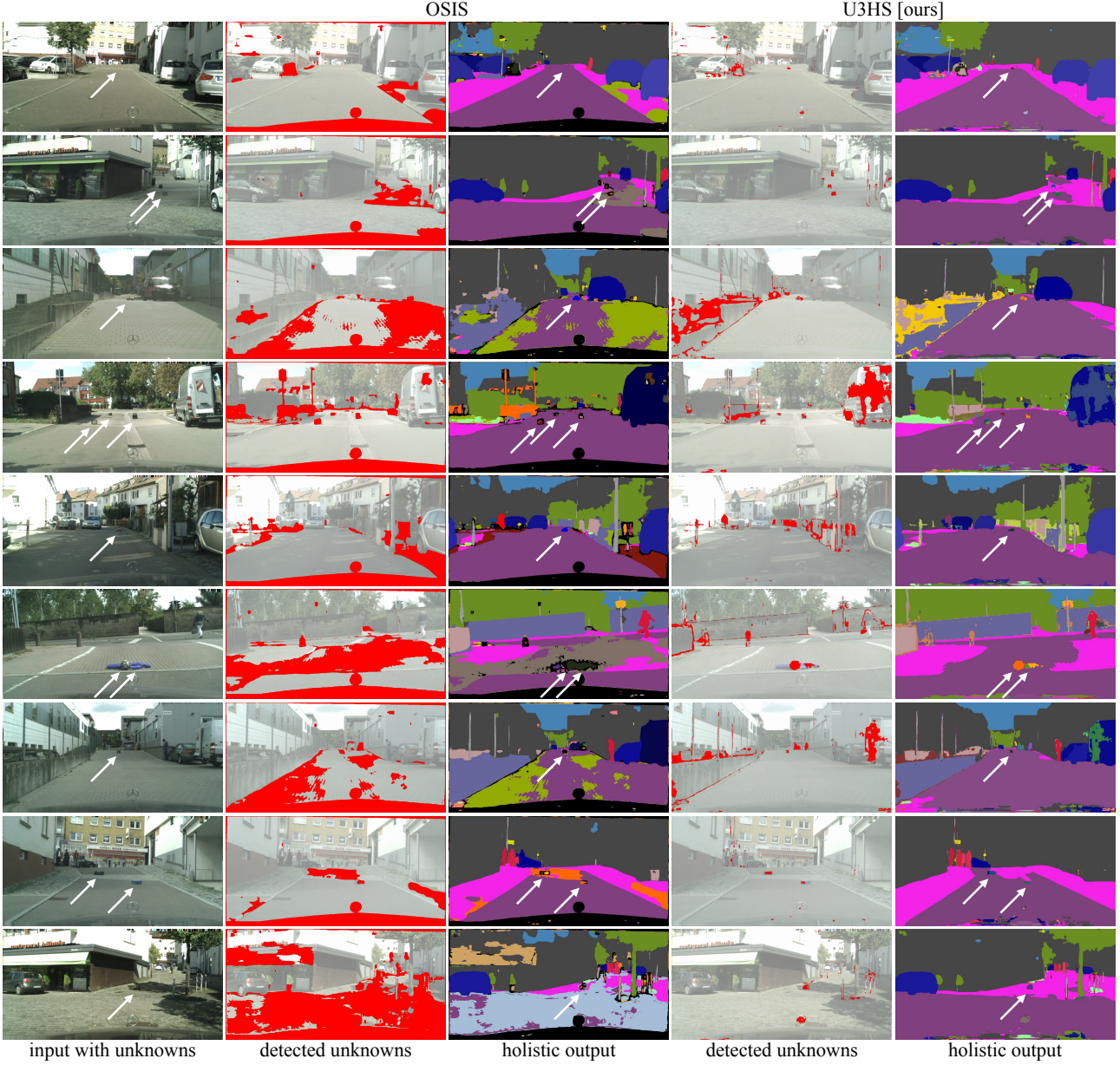


Figure 8. Example predictions of OSIS [22] and the proposed U3HS on unknown categories from the test set of Lost&Found [18]. The models were trained on Cityscapes [7] and transferred to Lost&Found without any fine-tuning. OSIS found unknowns as the *void* class (learned during training), while our U3HS discovered them via uncertainty estimation. Black regions in OSIS’s outputs, including around unknowns, represent pixels predicted as part of the unknown instance of the ego vehicle bonnet: since the bonnet is labeled as *void* in the training set, OSIS learned it as such and it turned it into an unknown instance at inference time. White arrows mark labeled OOD objects.

periment. First, instead of a standard CPU implementation, we used a parallelized CUDA version of the algorithm [24]. Then, due to the still very high memory requirements, specific samples of Lost&Found caused memory issues. Therefore, we reduced the size of the test set of Lost&Found [18] to its first 300 samples (Lost&Found-300), which were not problematic. These 300 samples are sufficient to indicate the effect of using Mean Shift instead of DBSCAN. Table 8

shows the superiority of DBSCAN for this setting, with a 3.5x higher PQ and 3.8x better RQ. In particular, RQ should be the focus as we compare instance segmentation of unknowns.

Additional details on EOPSN vs. OSIS As described in Section 5.2, EOPSN always diverged on Cityscapes despite numerous attempts, leading to null true positives, as shown in Table 1. OSIS did not suffer from this issue:

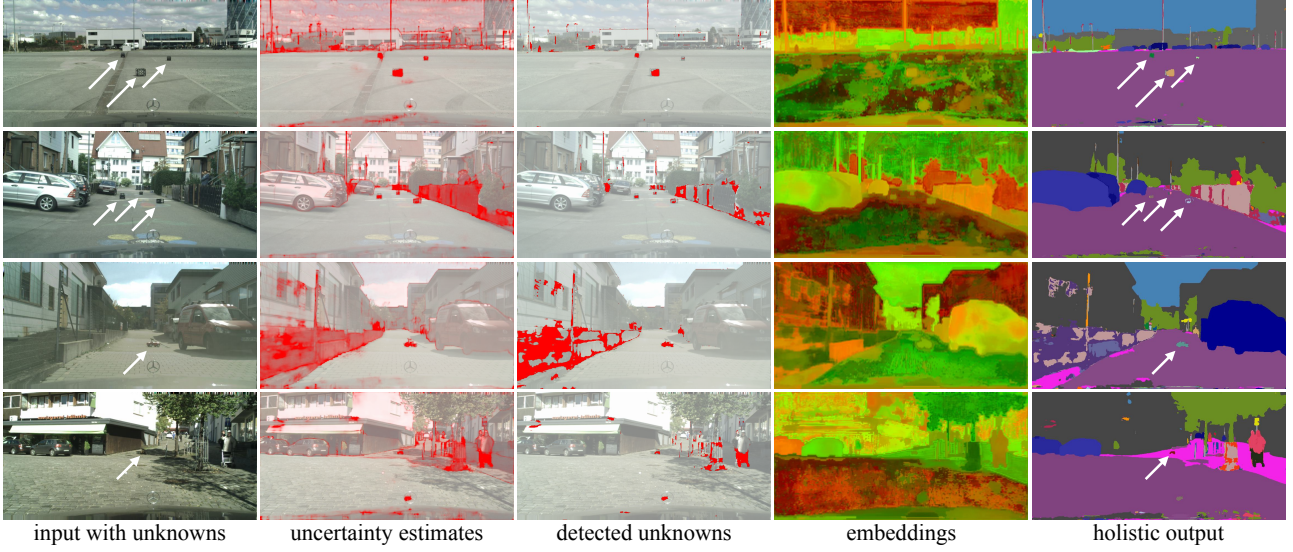


Figure 9. Additional example predictions of the proposed U3HS on unknown categories from the test set of Lost&Found [18]. The model was trained on Cityscapes [7] and transferred to Lost&Found without any fine-tuning. White arrows mark labeled OOD objects.

EOPSN’s mining strategy requires associating similar “unknowns” across different inputs, but OSIS operates frame-by-frame. Since *void* pixels are unstructured and undefined in Cityscapes, EOPSN’s association fails. As this process is a fundamental step of its training procedure, it makes EOPSN diverge and leads to null scores due to the absence of true positives. Instead, in EOPSN’s setup (i.e., re-identifying unlabeled objects seen during training), such associations can be made across the instances of the classes EOPSN’s authors treat as *void* or “unknown” (e.g., all cars in their setup, as in Table 2). Therefore, on MS COCO (Table 2), EOPSN managed to identify a few true positives, thereby scoring more than 0 for PQ and RQ and significantly more for SQ as it considers only the IoU of matched segments (TP) and not the wrong predictions (FP and FN).

Varying number of unknowns Both EOPSN and DDOSP showed that their performance drops across the board by increasing the amount of *void* classes to detect (i.e., K , pseudo-unknowns), especially for unknowns. As shown in Table 1, they perform poorly also with $K = 0$. Instead, our U3HS does not rely on *void*, so it is unaffected by K or what is assigned to *void*. This means that U3HS’s performance varies only slightly with different K s, as for random initialization. Moreover, whether turning known classes into *void* (MS COCO, Table 2, with $K = 5\%$) or not (Lost&Found, Table 1, $K = 0$), our method outperforms prior works despite letting the others learn from unknowns via *void*. Figure 8 shows this qualitatively. Thanks to uncertainty estimation, our setup has an edge with unseen categories (e.g., Table 1). Increasing K for open-set works (i.e., treating more classes as *void*) means reducing the number of classes that can be segmented semantically.

Therefore, the proposed setting is more practical than open-set panoptic and open-vocabulary because, for ours, no unknowns need to be part of the training of any model (simpler data collection), and ours detects any unseen categories.

A.5. Additional Qualitative Results

A.5.1 Qualitative Comparison

Figure 8 shows a comparison of the predictions of the proposed U3HS with the prior work OSIS [22], as well as the regions each predicted as unknown, on a set of samples from Lost&Found [18]. In particular, ours found unknowns as segments estimated as highly uncertain, and OSIS found them as the pixels predicted to be part of the learned *void* class.

From the images, it can be seen how for the most part, OSIS managed to learn a relatively good class boundary around the *void* class, as it was typically able to predict the OOD objects as unknown via *void*. This is interesting as it shows how OSIS can potentially work with challenging unseen unknowns. However, the same figure also shows the substantial limitations of learning and predicting *void* due to the assumptions about the data distributions this entails. In the first image, OSIS completely ignored the unknown object, assigning it to the *road* class, while in the fifth image, it detected the toy as *car*. In contrast, in the last picture, OSIS predicted almost everything as unknown. This proves how the binary aspect introduced by predicting the *void* class (a pixel is either unknown, by being *void*, or known, if another class) does not cope well with the diversity and unpredictability of the scenes in unconstrained real-world settings. Specifically, predicting the *void* class severely relies on the closed-set training data, as the suc-

cess of such a method is directly related to the diversity of the *void* class seen during training, which is limited as it cannot correctly sample the long tail of the data distribution [15].

Nevertheless, as shown already in Section 5, estimating the uncertainty allows to properly cope with unknown objects by adding an extra layer of prediction. Contrary to the idea of prior works (Section 2) of predicting unknowns via the *void* class, which directly competes with the other semantic classes for being part of the output, uncertainty estimates go on top of the standard semantic predictions. Although this complicates dealing with multiple network outputs, it offers a wider spectrum and deeper insights since the uncertainty could be ignored or considered with various thresholds depending on the situation (Figure 7), for the same trained model and output. Since estimating the uncertainty aims at smoothly quantifying the domain gap from the training data, we believe it is better suited to highly unpredictable unseen real-world scenarios as in holistic segmentation settings.

Furthermore, Figure 8 shows the capability of each method to identify instances of unknown OOD objects. For both approaches, this is related to the clustering of embeddings corresponding to those pixels predicted as unknown, via *void* (OSIS), or as highly uncertain (ours). In particular, OSIS tended to over-fragment unknown objects into several small instances, as seen in the fourth, sixth, eighth, and last images. This again proves our modifications’ effectiveness when dealing with the embeddings, as described in Section 4 and evaluated in Table 3. Additionally, OSIS could not distinguish the two neighboring OOD objects in the sixth image. Moreover, OSIS often improperly assigned large regions to the same unknown instance. Similarly to ours, OSIS considers every unknown segment as part of an instance. By learning and predicting the *void* class, during training, OSIS learned to precisely segment the bonnet of the ego car (labeled as *void* in Cityscapes [7]). However, at test time on Lost&Found it could not tell the ego vehicle bonnet apart from a wide variety of pixels. This was the case for the unknown object in the seventh image, which was entirely assigned to the same instance as the bonnet or many other segments around knowns and unknowns (colored in black). The ego vehicle bonnet unknown instance (black) often surrounded other predicted unknown instances (e.g., in the second, fourth, sixth, eighth, and last images).

A benefit of estimating the uncertainty is the ability to account for a wide array of unusual regions. This is valuable for downstream tasks, e.g., trajectory prediction and path planning. Specifically, uncertainty estimates by the proposed U3HS were high on the stroller in Figure 1, as well as in Figure 4 on the walking assistance device on the left of the upper image and the cart pushed by the man waving on the right of the bottom image in Figure 9, none of

which were labeled as unknown in the dataset [18], as they were not part of the objects manually placed by the authors. In Figure 8, this is repeated from a different perspective on the stroller in the background of the second image, the unusual van with the open doors in the fourth image, and the duffel bag in the sixth. By learning and predicting *void*, OSIS ignored these unusual regions as it lacks the flexibility and granularity that our U3HS offers by estimating the uncertainty.

A.5.2 Additional Results on Unknowns

Lost&Found Figure 9 shows additional qualitative outputs. Once again, it can be seen how challenging the proposed holistic segmentation setting is. As in the predictions of Figure 4, the model can distinguish most unknown objects. It can be seen how specific areas of the images trigger higher uncertainty estimates. This is the case of the fences in the second and third images of Figure 9, as well as unknown objects not part of the OOD labels of Lost&Found, such as the cart on the right of the bottom image, as previously mentioned. As previously seen, *stuff* structures (e.g., fences) are assigned to a single coherent instance ID throughout the whole image. At the same time, unusual objects (e.g., the cart in the last picture) have their dedicated ID. Figure 9 also provides some examples of unusual scenes present in the Lost&Found [18] dataset, posing significant challenges compared to Cityscapes [7].

MS COCO Figures 10 and 11 show qualitative outputs of the proposed U3HS on the held out classes of MS COCO [16]. The images report the vast diversity of the dataset, ranging from outdoor scenes to indoor close-ups. Remarkably, U3HS delivered precise segments for unknown objects, correctly segmenting instances of individual unknowns despite their similarity with other objects of the same type in the same input, with reasonable estimations of known classes too (Figure 10). Due to the difficulty of this problem, only a handful of segments are perfect, leaving room for improvements for known and unknown objects. Thanks to its strong uncertainty estimation capabilities, the proposed U3HS not only identified the held-out classes but also other unknown objects which are not part of the set of known classes, such as the rice cooker and the umbrella on the right of Figure 11 in the third and bottom rows respectively. This shows the efficacy of our method on a wide variety of scenarios typical of the real world.

A.5.3 Failure Cases

MS COCO While the proposed U3HS delivers reasonable estimates in various settings, from indoor to outdoor, the problem at hand is highly challenging, and its predictions are not perfect, as confirmed by the quantitative results. Figure 12 reports failure cases on a set of challenging samples

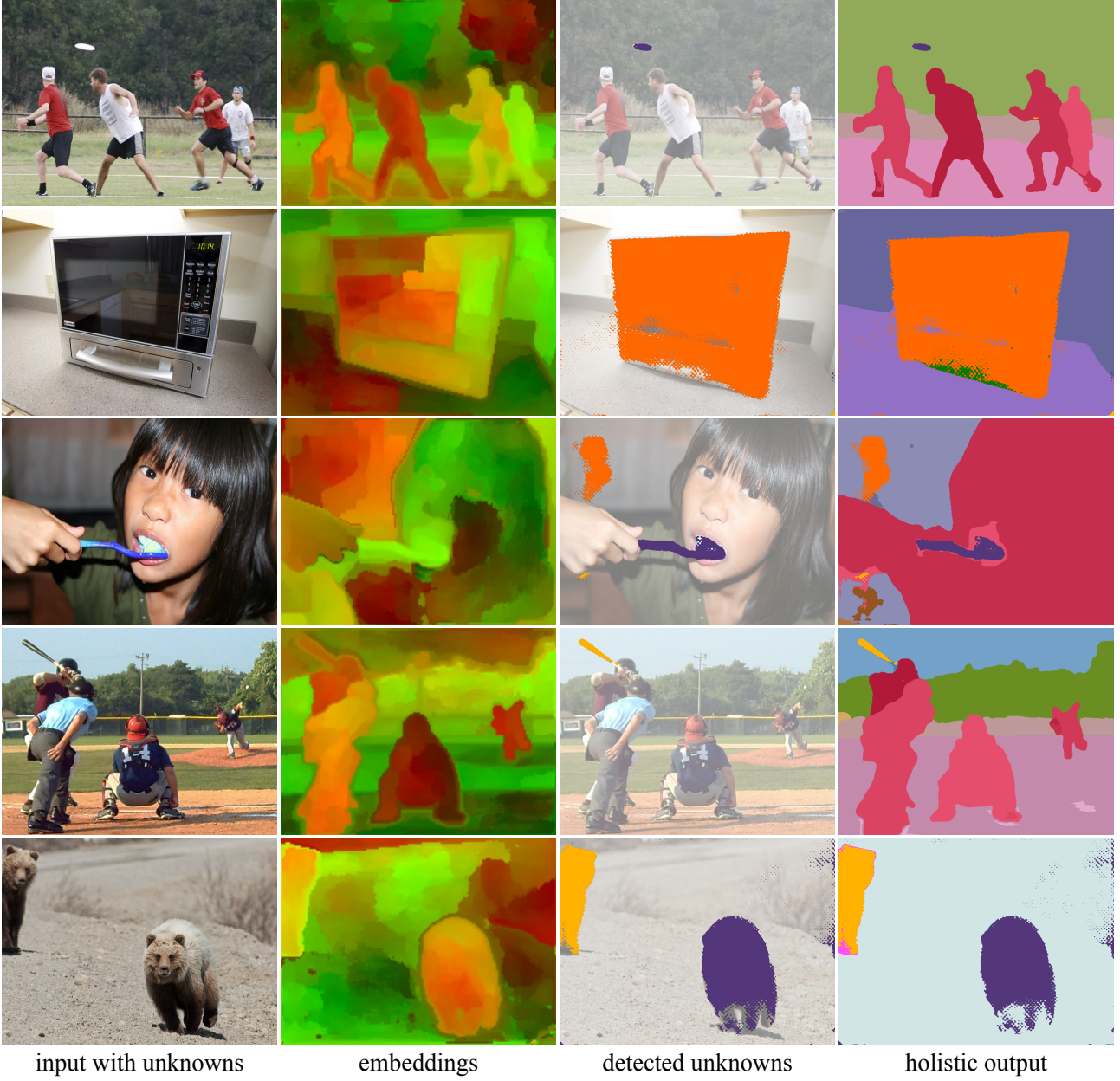
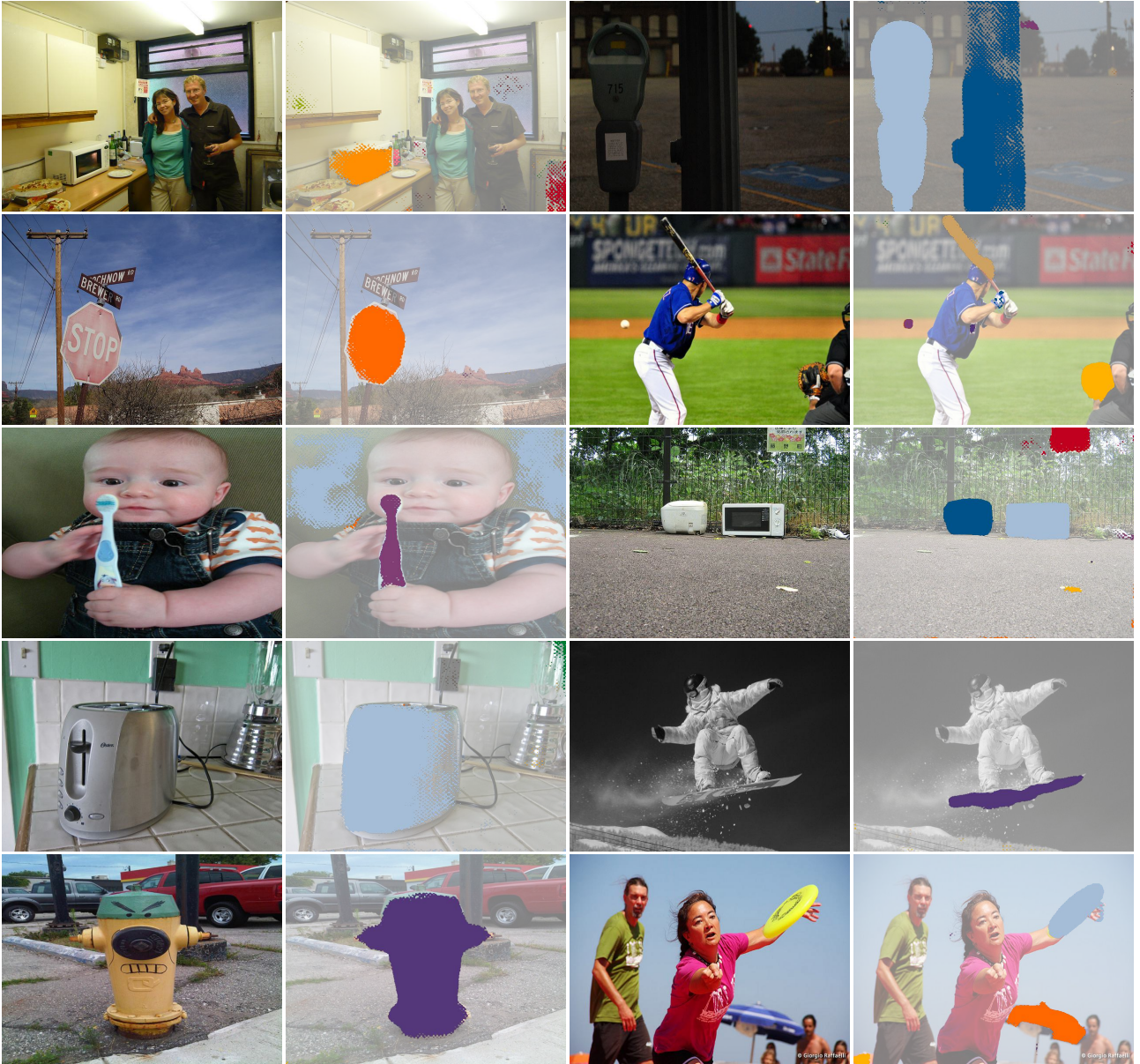


Figure 10. Example predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *frisbee*, *microwave*, *toothbrush*, *baseball bat*, and *bear*. All samples are equally resized. Input, embeddings, detected unknowns, and holistic output are shown.

of MS COCO. It can be seen that while U3HS identified unknowns reasonably, it often missed those objects that are part of the evaluated held-out categories. A series of issues cause this. In some instances, the integrated uncertainty estimates could not fully discover the unseen unknowns, e.g., only partially detected for both *refrigerators* at the left of the third and fourth rows. We also noticed systematic issues with certain classes, such as *keyboard*, *mouse* (both in the right of the fourth row), *hot dog* (in the fifth row), and

scissors (right of the last row). This could be attributed to these objects being semantically relatively close to known classes, such as *sandwich* for *hot dog*. Small objects were hard to see and therefore ignored by our method. This is the case of the *toothbrush* behind the *cat* in the last row of the figure. U3HS also had difficulties telling apart from one another very close and similar unknowns, e.g., the central *frisbees* in the right of the second row differing for the logo’s color. However, it could successfully separate neighboring



input with unknowns

detected unknowns

input with unknowns

detected unknowns

Figure 11. Example predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *microwave*, *parking meter*, *stop sign*, *baseball bat*, *toothbrush*, *toaster*, *snowboard*, *fire hydrant*, and *frisbee*. Other unknown objects are included in the samples, such as the *umbrella* and the *rice cooker* (i.e., not part of the known classes). All samples are equally resized. Input and detected unknowns are shown.

objects on multiple occasions, such as the containers on the ground of the left image in the fourth row (not evaluated in this experiment). Moreover, we noticed difficulties with particularly unusual inputs and cluttered environments, e.g., on the left of the second row. With other inputs, the uncertainty of U3HS was also triggered on known objects, such as the *dogs* on the top right (albeit correctly separated into two instances), or the *cabinets* on the right of the third row.

Since several held-out classes contained everyday kitchen-related items (e.g., *refrigerator*, *microwave*, and *toaster*), or typical desk objects (e.g., *keyboard* and *mouse*), the model could see only a handful of kitchens and offices (i.e., those images where none of these held out objects appeared), which severely impacted its ability to handle these situations appropriately. This is a limitation of holding out samples from MS COCO, compared to using a dedicated sepa-



input with unknowns

detected unknowns

input with unknowns

detected unknowns

Figure 12. Failure predictions of the proposed U3HS on OOD data containing the held out classes of MS COCO [16]. Held out classes (unseen unknowns) in the samples: *frisbee*, *keyboard*, *mouse*, *refrigerator*, *hot dog*, *toothbrush*, and *scissors*. Other unknown objects are included in the samples, such as bag and comb (i.e., not part of the known classes). All samples are equally resized. Input and detected unknowns are shown.

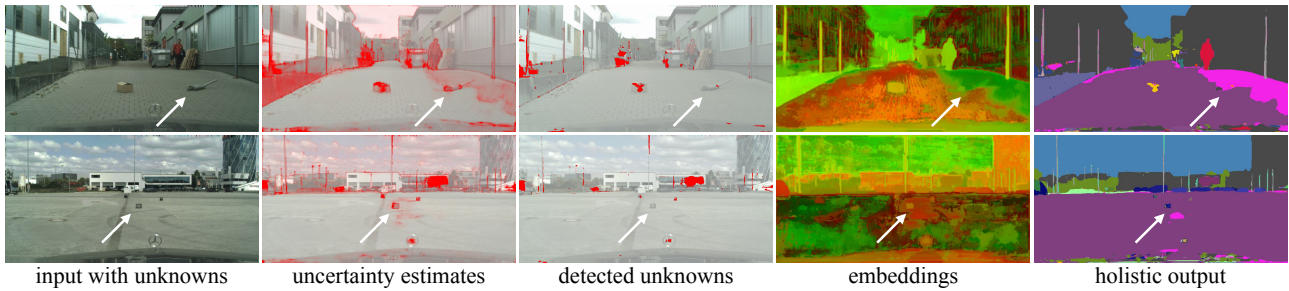


Figure 13. Failure predictions of the proposed U3HS on OOD data from the test set of Lost&Found [18], with a transfer from Cityscapes [7]. White arrows mark missed OOD objects as the estimated uncertainty was relatively low and filtered out.

rate dataset, such as Lost&Found [18]. Furthermore, given that U3HS estimates the model uncertainty, more training data covering a wider variety of scenarios could be beneficial to further reduce the uncertainty on the known classes and improve the known-unknown boundary of U3HS.

Lost&Found Figure 13 shows failure cases caused by the necessary filtering of the uncertainty estimates. While the uncertainty was triggered by a variety of unusual areas, including the vast majority of unknown objects, its a priori filtering (based on closed-set training data, Section 4.2) sometimes caused the unknown object to be completely undetected. Although this filtering is aimed at removing low uncertainty areas which are probably in-domain (e.g., the fence in the upper image), it could inadvertently remove proper OOD objects (e.g., those marked by the white arrows). This is related to the trade-off shown in Figure 7, so keeping more unknowns (i.e., lower threshold t) reduces the in-domain performance. Nevertheless, in the embeddings visualizations, the model correctly isolated the entire marked box in the lower image and precisely segmented the cardboard box in the upper one. However, the two unknown objects were not detected, due to the difficulty of merging multiple outputs and interpreting uncertainty estimates without access to OOD data. It should be considered that the proposed U3HS does not distinguish between the uncertainty for unknown objects and that of unusual known classes. The difference might lie in the amount of uncertainty corresponding to these regions, hence the filtering via the threshold t to attempt telling apart completely unknown from unusual, which remains highly challenging without using any information about unknowns at training time, as in our setup.

References

- [1] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes benchmark: Measuring blind spots in semantic segmentation. *Springer International Journal of Computer Vision*, 129(11):3119–3135, 2021. 2
- [2] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15333–15342, 2021. 4, 5
- [3] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMeIfYouCan: A benchmark for anomaly segmentation. In *Neural Information Processing Systems - Datasets and Benchmarks Track*, 2021. 1, 2
- [4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818. Springer, 2018. 4
- [5] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 1
- [6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 5
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1, 2, 3, 4, 5, 6, 7, 8, 12
- [8] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 3
- [9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. 3
- [10] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 4, 5
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

- ings of the *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4, 5
- [12] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7031, 2022. 2
- [13] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1184, 2021. 2, 4
- [14] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized Max Logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15425–15434, 2021. 1, 4, 5
- [15] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3D-VField: Adversarial augmentation of point clouds for domain generalization in 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17295–17304, 2022. 1, 2, 8
- [16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3, 4, 8, 9, 10, 11
- [17] Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020. 3, 4, 5
- [18] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting small road hazards for self-driving vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1099–1106, 2016. 2, 3, 4, 5, 6, 7, 8, 12
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3
- [20] Murat Sensoy, Lance M. Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing*, pages 3183–3193, 2018. 4, 5
- [21] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020. 4, 5
- [22] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *Proceedings of the Conference on Robot Learning*, pages 384–393. PMLR, 2020. 2, 3, 4, 5, 6, 7
- [23] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Proceedings of the European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2
- [24] Le You, Han Jiang, Jinyong Hu, C Hwa Chang, Lingxi Chen, Xintong Cui, and Mengyang Zhao. GPU-accelerated faster Mean Shift with Euclidean distance metrics. In *Proceedings of the Computers, Software, and Applications Conference*, pages 211–216. IEEE, 2022. 6

B.3 Appendix: Radar to Improve the Depth Estimation of Dynamic Objects (3DV 2021)

This section includes the supplementary material associated with [83], presented and discussed in Section 5.3.1.

Supplementary Material of:

R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes

Stefano Gasperini^{*,1,2} Patrick Koch^{*,1,2} Vinzenz Dallabetta²
Nassir Navab^{1,3} Benjamin Busam¹ Federico Tombari^{1,4}

¹ Technical University of Munich ² BMW Group ³ Johns Hopkins University ⁴ Google

* equal contribution

IEEE International Conference on 3D Vision (3DV), 2021

A. Supplementary Material

In this Section we include further details and additional results computed from the same models reported in the main manuscript.

A.1. LiDAR and Radar

In Figure 7 we show the significant difference between LiDAR and radar signals. As described in Section 3, due to the physical properties of the sensor, radar signals are noisy, sparse and often 1-dimensional. The complex multi-path noise can be seen affecting the two points marked by arrows in the Figure. Despite their mutual distance of 5.2 meters, the points are projected onto the same foreground object in the image space. This once again shows the importance of our modifications to handle radar data for weak supervision (Section 3.2.3).

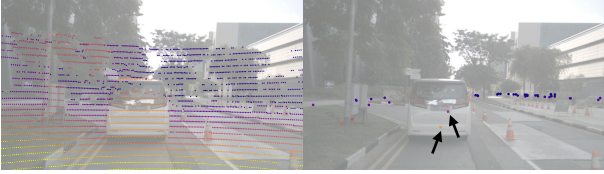


Figure 7. LiDAR (left) and radar (right) signals from nuScenes [3]. The two radar measurements marked by arrows are 5.2 meters apart, due to multi-path effects.

A.2. Self-supervised Framework

In the following, we further specify the loss functions described in Section 3.1. The photometric loss is a combination of $\mathcal{L}_1$ -loss and SSIM [32], as in [10]:

$$\begin{aligned}\mathcal{L}_1(I_t, \hat{I}_t) &= \|I_t - \hat{I}_t\|_1 \\ \mathcal{L}_{\text{SSIM}} &= 1 - \text{SSIM}(I_t, \hat{I}_t) \\ pe(I_t, \hat{I}_t) &= (1 - \alpha) \mathcal{L}_1(I_t, \hat{I}_t) + \frac{\alpha}{2} \mathcal{L}_{\text{SSIM}}(I_t, \hat{I}_t)\end{aligned}\quad (6)$$

where $\alpha = 0.85$ balances between the $\mathcal{L}_1$ and the SSIM term. Additionally, as in [11], we only consider the minimum reprojection error to account for partial occlusions:

$$\mathcal{L}_p(I_t, \hat{I}_{s \rightarrow t}) = \min_s pe(I_t, \hat{I}_{s \rightarrow t}). \quad (7)$$

Furthermore, we follow [11] by automatically masking out pixels which do not change appearance in between frames:

$$M_a = \min_s \mathcal{L}_p(I_t, I_s) > \min_s \mathcal{L}_p(I_t, \hat{I}_{s \rightarrow t}). \quad (8)$$

Hence, the photometric loss is only considered in regions where $M_a = 1$. Moreover, to encourage local smoothness

while preserving edges we use a specific term from [10]:

$$\mathcal{L}_s(I_t, d_t^*) = \frac{1}{N} \sum_{p \in N} \sum_{i \in x, y} |\partial_i d_t^*(p)| e^{-|\partial_i I_t|} \quad (9)$$

where $|\cdot|$ denotes the element-wise absolute value, ∂_x and ∂_y are gradients in x- and y-direction, and $d_t^* = d_t^*/\bar{d}_t^*$ the mean-normalized inverse of the depth prediction.

As described in Section 3.1, we follow [12] with a weak velocity supervision $\mathcal{L}_v$ for scale-awareness:

$$\mathcal{L}_v(\hat{\mathbf{t}}_{t \rightarrow s}, \mathbf{t}_{t \rightarrow s}) = \left| \|\hat{\mathbf{t}}_{t \rightarrow s}\|_2 - \|\mathbf{t}_{t \rightarrow s}\|_2 \right| \quad (10)$$

where $\hat{\mathbf{t}}_{t \rightarrow s}$ and $\mathbf{t}_{t \rightarrow s}$ are the predicted and ground truth pose translations respectively, easily obtainable from readily available velocity information (e.g. via odometry).

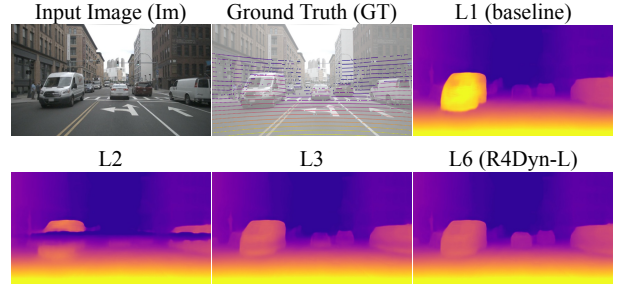


Figure 8. Qualitative comparison complementing the ablation study on the weak radar supervision (Table 2).

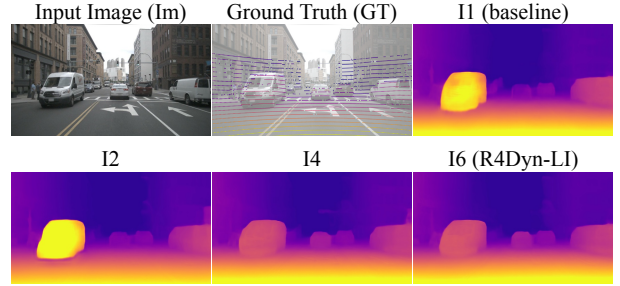


Figure 9. Qualitative comparison complementing the ablation study on the pre-processing for radar as additional input (Table 3).

A.3. Radar Accumulation

The accumulation of multiple radar measurements described in Section 3.2.1 is performed by mapping points from time $t - \tau$ to t , as proposed by [17]. Specifically, the ego- and target-motion compensation across multiple measurements is done considering the doppler velocity as:

$$\begin{aligned}x_t &= x_{t-\tau} + \Delta d_{x,t} + v_{x,t-\tau} \cdot \Delta \tau \\ y_t &= y_{t-\tau} + \Delta d_{y,t} + v_{y,t-\tau} \cdot \Delta \tau \\ z_t &= z_{t-\tau} + \Delta d_{z,t}\end{aligned}\quad (11)$$

where x_t , y_t and z_t denote the estimated target location at time t , Δd the ego pose transformation between measurements, and v the measured doppler velocity of the target.

A.4. Additional Implementation Details

As we focus on safety critical dynamic objects (e.g. traffic participants), when filtering the radar signal to use it as weak supervision (Section 3.2), we only consider 2D bounding boxes of the following classes of nuScenes [3]: *Vehicles* (including all sub-classes) and *Pedestrians* (including all sub-classes). As nuScenes provides only 3D bounding boxes, we obtained 2D boxes by considering the top, left, bottom, and right extremes of the projected 3D boxes. Furthermore, as we want to expand the radar signal over the objects, the binary association mask from Equation 2 should vary in size depending on the object, e.g. larger for a bus, than for a pedestrian. Additionally, we consider as most reliable detections those projected in the lower center of a bounding box. Towards this end, we set the smoothing parameter σ_d according to the bounding box dimensions, hence accounting for the object size: we set

Method	<i>all</i>	<i>clear</i>	<i>rain</i>	<i>night</i>
<i>Cars</i>				
Lin et al. [21]	0.217	0.191	0.225	0.433
Struct2Depth [4]	0.346	0.332	0.320	0.535
PackNet-SfM [12]	0.219	0.181	0.270	0.449
Monodepth2 [11]	0.232	0.199	0.261	0.486
baseline [ours]	0.212	0.186	0.240	0.391
R4Dyn-L [ours]	0.164	0.134	0.219	0.311
R4Dyn-LI [ours]	0.140	0.125	0.171	0.204
<i>Vehicles</i>				
Lin et al. [21]	0.229	0.208	0.233	0.428
Struct2Depth [4]	0.361	0.352	0.335	0.535
PackNet-SfM [12]	0.259	0.238	0.274	0.440
Monodepth2 [11]	0.240	0.211	0.270	0.483
baseline [ours]	0.231	0.209	0.259	0.384
R4Dyn-L [ours]	0.182	0.162	0.220	0.308
R4Dyn-LI [ours]	0.159	0.150	0.176	0.200
<i>Pedestrians</i>				
Lin et al. [21]	0.292	0.293	0.254	0.401
Struct2Depth [4]	0.298	0.299	0.276	0.367
PackNet-SfM [12]	0.252	0.247	0.284	0.355
Monodepth2 [11]	0.258	0.257	0.253	0.317
baseline [ours]	0.238	0.235	0.244	0.348
R4Dyn-L [ours]	0.229	0.223	0.255	0.409
R4Dyn-LI [ours]	0.218	0.215	0.234	0.332

Table 6. Evaluation of AbsRel on *Cars*, *Vehicles* and *Pedestrians* under adverse conditions of the validation set of nuScenes [3]. All methods were trained only on scenes with *day-clear* (*clear*) conditions. This Table complements Table 5, therefore shows the ability of each method to generalize to unseen settings.

$\sigma_{d,x} = c \cdot s \cdot bb_w/2$ and $\sigma_{d,y} = c \cdot s \cdot bb_h/2$, where bb_w and bb_h are the box width and height, and c is a constant scale factor set to 1.5. Moreover, s is a scale factor that depends on the position of the considered radar point with respect to the bounding box: $s = \delta_{side}/(bb_w/2) \cdot \delta_{top}/bb_h$, with δ_{side} and δ_{top} being the minimum distance of the radar point from the bounding box side and top edges respectively. The range smoothing parameter is fixed to $\sigma_r = 1 \times 10^{-5}$.

For prior works, as mentioned in Section 4.1, we used the original implementation and parameters, adapted as follows to accommodate the different dataset (i.e. nuScenes [3]). We trained Monodepth2 [11] for 40 epochs, which is the same as our R4Dyn. PackNet-SfM [12] was trained for a total of 200 epochs due to its slow convergence, and Struct2Depth [4] was trained for 75 epochs after the full KITTI [9] training performed by the authors. For fairness when comparing with other approaches, for Struct2Depth [4] we used the motion model (denoted by the authors with an M), but not their online refinement (indicated with an R). To ensure fair comparability with other methods, for the work of Lin et al. [21] we only used radar measurements from the past and present, but not from the future, which were used in the original implementation.

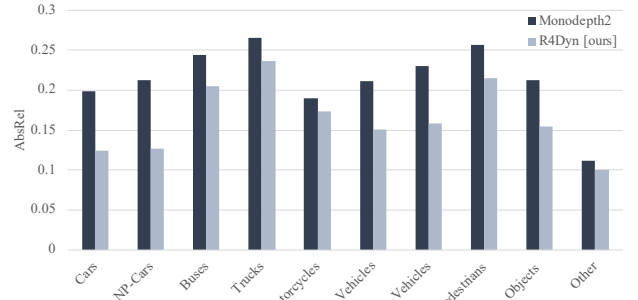


Figure 10. The histogram shows a class-wise evaluation of our R4Dyn and Monodepth2 [11] on the nuScenes *day-clear* validation set [3]. *Other* includes all remaining classes, that are not traffic participants, and *NP* indicates *Non-Parked Cars* or *Vehicles*. Lower absolute relative (AbsRel) error is better.

A.5. Additional Results

A.5.1 Qualitative Ablation Studies

In Figures 8 and 9 we show highlights of the qualitative results of our ablation studies on the weak radar supervision and additional radar input, respectively. These complement the quantitative results reported in Tables 2 and 3 with the sample being the same as the first one in Figure 6 for consistency. In the Figures, it can be seen how our image-only baseline (L1 and I1) performs similarly to Monodepth2 [11] on which it is based. Then, in Figure 8, adding a weak supervision by raw radar points (L2) deteriorated the estimations due to the considerable amount of noise caused by

Cl.	Method	Sup.	Input	AbsRel	SqRel	RMSE	RMSE _{log}	δ_1	δ_2	δ_3
<i>Cars</i>	Lin et al. [21]	GT	ImR	0.1907	2.399	6.922	0.2460	75.71	91.02	96.06
	Struct2Depth [4]	M*	Im	0.3323	7.436	9.353	0.3307	57.38	80.54	89.98
	PackNet-SfM [12]	Mv	Im	0.1814	1.936	6.313	0.2341	72.88	91.02	96.46
	Monodepth2 [11]	M*	Im	0.1983	2.100	6.635	0.2509	68.23	88.76	94.81
	baseline [ours]	Mv	Im	0.1862	2.115	6.735	0.2495	70.41	87.96	94.41
	R4Dyn-L [ours]	Mvr	Im	0.1343	1.481	5.713	0.1913	81.45	94.02	97.32
	R4Dyn-LI [ours]	Mvr	ImR	0.1250	1.371	5.395	0.1813	84.14	94.38	97.43
<i>NP-Cars</i>	Lin et al. [21]	GT	ImR	0.1898	2.485	6.793	0.2418	77.15	90.69	95.65
	Struct2Depth [4]	M*	Im	0.3703	9.063	10.05	0.3472	54.42	78.15	88.35
	PackNet-SfM [12]	Mv	Im	0.1902	2.050	6.397	0.2394	70.56	90.35	96.20
	Monodepth2 [11]	M*	Im	0.2130	2.368	6.905	0.2624	65.05	87.03	93.97
	baseline [ours]	Mv	Im	0.1862	2.115	6.735	0.2495	70.41	87.96	94.41
	R4Dyn-L [ours]	Mvr	Im	0.1356	1.518	5.651	0.1894	80.92	93.56	97.16
	R4Dyn-LI [ours]	Mvr	ImR	0.1274	1.399	5.302	0.1793	83.80	94.01	97.43
<i>Buses</i>	Lin et al. [21]	GT	ImR	0.2330	3.336	8.456	0.2728	65.46	85.48	93.44
	Struct2Depth [4]	M*	Im	0.3962	7.487	12.11	0.4057	41.22	69.19	85.83
	PackNet-SfM [12]	Mv	Im	0.3434	7.328	11.60	0.3662	52.96	77.66	88.94
	Monodepth2 [11]	M*	Im	0.2442	4.518	9.781	0.3007	63.13	83.44	91.84
	baseline [ours]	Mv	Im	0.2547	4.614	9.586	0.3080	62.85	82.28	91.73
	R4Dyn-L [ours]	Mvr	Im	0.2187	3.950	8.719	0.2779	68.08	85.45	93.14
	R4Dyn-LI [ours]	Mvr	ImR	0.2055	3.706	8.316	0.2600	70.35	86.28	93.46
<i>Trucks</i>	Lin et al. [21]	GT	ImR	0.2356	3.248	8.410	0.2829	63.70	83.46	92.67
	Struct2Depth [4]	M*	Im	0.3711	7.153	12.26	0.4027	45.32	71.41	84.74
	PackNet-SfM [12]	Mv	Im	0.3472	6.561	11.48	0.3758	49.82	75.84	87.82
	Monodepth2 [11]	M*	Im	0.2659	5.330	10.42	0.3221	60.76	81.03	91.48
	baseline [ours]	Mv	Im	0.2751	4.782	10.25	0.3340	55.26	80.55	90.05
	R4Dyn-L [ours]	Mvr	Im	0.2457	4.184	9.739	0.3077	58.71	83.26	92.09
	R4Dyn-LI [ours]	Mvr	ImR	0.2369	4.219	9.493	0.2997	63.17	82.85	92.64
<i>Motorcycles</i>	Lin et al. [21]	GT	ImR	0.2529	3.473	8.292	0.2757	62.84	84.27	93.50
	Struct2Depth [4]	M*	Im	0.2328	3.110	8.026	0.2854	58.39	82.58	90.00
	PackNet-SfM [12]	Mv	Im	0.2007	2.575	7.062	0.2529	67.38	88.33	93.28
	Monodepth2 [11]	M*	Im	0.1900	2.457	6.693	0.2409	72.86	89.33	93.23
	baseline [ours]	Mv	Im	0.1849	2.481	6.856	0.2486	72.11	89.26	92.93
	R4Dyn-L [ours]	Mvr	Im	0.1833	2.534	6.885	0.2496	69.71	89.43	93.05
	R4Dyn-LI [ours]	Mvr	ImR	0.1730	2.401	6.519	0.2389	75.26	89.56	92.64

Table 7. Class(Cl.)-wise evaluation on the main individual *Vehicle* classes on the nuScenes [3] *day-clear* validation set. *NP* stands for *Non-Parked*. Table to be considered in conjunction with Table 8.

multi-path effects [23], resulting in an overestimated set of pixels around the horizon line, which is the image location where most radar detections lie on (as can be seen in Figure 8). As reported in Table 2, filtering the radar points (L3) led to a significant improvement in the estimations, which then improved further with our R4Dyn-L (L6), e.g. for the leading vehicle in the image.

For the input ablation (Figure 9), a single frame of radar input measurements (I2) did not improve the predictions, while the weak radar supervision and accumulating multi-

ple radar measurements (I4) eliminate the severe underestimation, but delivers artifacts on the shape of vehicles (e.g. oncoming van), which are improved by R4Dyn-LI (I6).

A.5.2 Class-wise Comparison with Related Works

Additionally to the absolute relative error (AbsRel) on object classes *Cars*, *Vehicles*, *Non-parked Vehicles* and *Pedestrians* reported in Table 1, we provide extensive class-wise results in Tables 7 and 8. Moreover, we plot a compari-

Cl.	Method	Sup.	Input	AbsRel	SqRel	RMSE	RMSE _{log}	δ_1	δ_2	δ_3
<i>Vehicles</i>	Lin et al. [21]	GT	ImR	0.2082	2.637	7.400	0.2668	73.08	88.99	95.08
	Struct2Depth [4]	M*	Im	0.3516	7.179	10.13	0.3612	54.53	79.25	89.63
	PackNet-SfM [12]	Mv	Im	0.2382	3.393	7.927	0.2885	67.48	87.71	94.14
	Monodepth2 [11]	M*	Im	0.2110	2.809	7.617	0.2726	68.89	87.85	94.50
	baseline [ours]	Mv	Im	0.2091	2.680	7.597	0.2775	67.91	87.33	93.91
	R4Dyn-L [ours]	Mvr	Im	0.1618	2.047	6.681	0.2273	77.16	92.72	96.76
	R4Dyn-LI [ours]	Mvr	ImR	0.1504	1.922	6.371	0.2188	80.51	92.77	96.75
<i>NP-Vehicles</i>	Lin et al. [21]	GT	ImR	0.2088	2.892	7.396	0.2597	74.98	89.46	94.96
	Struct2Depth [4]	M*	Im	0.3739	8.440	10.58	0.3643	53.17	77.87	88.58
	PackNet-SfM [12]	Mv	Im	0.2508	3.699	8.044	0.2923	66.24	86.47	93.39
	Monodepth2 [11]	M*	Im	0.2300	3.372	7.917	0.2815	66.25	86.16	93.29
	baseline [ours]	Mv	Im	0.2254	3.219	7.885	0.2880	66.67	84.78	92.27
	R4Dyn-L [ours]	Mvr	Im	0.1686	2.461	6.728	0.2268	77.86	91.87	95.97
	R4Dyn-LI [ours]	Mvr	ImR	0.1589	2.311	6.375	0.2162	80.86	92.61	96.16
<i>Pedestrians</i>	Lin et al. [21]	GT	ImR	0.2930	4.496	9.507	0.2966	59.51	84.11	93.09
	Struct2Depth [4]	M*	Im	0.2993	5.489	11.48	0.3714	49.37	74.98	87.35
	PackNet-SfM [12]	Mv	Im	0.2473	4.171	9.301	0.2987	61.93	86.17	92.33
	Monodepth2 [11]	M*	Im	0.2572	4.420	9.831	0.3087	61.46	86.04	91.88
	baseline [ours]	Mv	Im	0.2351	4.004	9.117	0.2961	66.20	86.37	92.08
	R4Dyn-L [ours]	Mvr	Im	0.2231	3.670	8.806	0.2853	66.90	87.00	92.42
	R4Dyn-LI [ours]	Mvr	ImR	0.2146	3.613	8.560	0.2763	70.74	86.73	92.41
<i>Objects</i>	Lin et al. [21]	GT	ImR	0.2227	2.767	7.319	0.2735	72.33	88.68	95.10
	Struct2Depth [4]	M*	Im	0.3400	6.552	9.726	0.3543	56.16	80.38	90.92
	PackNet-SfM [12]	Mv	Im	0.2383	3.405	7.757	0.2861	69.02	88.50	94.75
	Monodepth2 [11]	M*	Im	0.2123	2.832	7.592	0.2700	70.11	89.14	95.37
	baseline [ours]	Mv	Im	0.2032	2.548	7.278	0.2693	70.90	88.97	95.11
	R4Dyn-L [ours]	Mvr	Im	0.1631	1.997	6.522	0.2279	78.42	93.22	97.16
	R4Dyn-LI [ours]	Mvr	ImR	0.1551	2.020	6.367	0.2222	81.20	92.73	97.09
<i>Other</i>	Lin et al. [21]	GT	ImR	0.0781	0.693	4.268	0.1542	93.07	97.35	98.73
	Struct2Depth [4]	M*	Im	0.1873	3.254	6.823	0.2645	78.61	90.80	95.42
	PackNet-SfM [12]	Mv	Im	0.1141	1.723	5.777	0.2051	88.68	95.54	97.70
	Monodepth2 [11]	M*	Im	0.1117	1.344	5.354	0.1913	89.50	96.54	98.15
	baseline [ours]	Mv	Im	0.1010	1.207	5.223	0.1916	90.73	96.32	97.97
	R4Dyn-L [ours]	Mvr	Im	0.1036	1.194	5.290	0.1931	90.20	96.21	97.93
	R4Dyn-LI [ours]	Mvr	ImR	0.1003	1.197	5.190	0.1886	91.13	96.41	98.00

Table 8. Class(Cl.)-wise evaluation on *Vehicles* (includes all individual classes from Table 7, plus *Bicycles*, *Construction Vehicles* and *Emergency Vehicles*), *Non-Parked Vehicles*, *Pedestrians*, *Objects* (all object classes together) and *Other* (pixels that do not correspond to objects) on the nuScenes [3] *day-clear* validation set. Table to be considered in conjunction with Table 7.

son between Monodepth2 [11] and our R4Dyn in Figure 10 across the various classes. From the Figure, it can be seen that our approach outperformed Monodepth2 on every object class by a significant margin, which was especially large for the most dynamic ones, namely *Cars*, *Vehicles* and their *Non-Parked (NP)* variants. Considering Tables 7 and 8, the LiDAR-supervised work by Lin et al. [21], which uses radar as input, was able to deliver superior estimates on non-object classes, denoted as *Other* (e.g. *Driveable*

Surface and *Vegetation*). Nevertheless, our R4Dyn could obtain significantly lower errors and better scores on all object classes, except for *Trucks*, probably due to trucks being often static in urban areas (e.g. for loading and for deliveries), such as in the nuScenes dataset [3]. The results re-emphasize and confirm the effectiveness and the robustness of our approach across various safety critical dynamic objects, as well as the benefit of using radar sensors for depth estimation, especially as weak supervision.

A.5.3 Class-wise Comparison on Adverse Conditions

In Table 6 we provide class-wise results of ours and related methods under adverse weather settings. As for Table 5, which reports general errors in the same weather conditions, all approaches were trained on *day-clear* scenes (indicated as *clear* in the Table), therefore the values represent the ability of each method to generalize to rather different inputs. In particular, in Table 6 we provide AbsRel errors on safety critical traffic participants: *Cars*, *Vehicles* and *Pedestrians*. For these classes, our R4Dyn-LI was able to outperform related methods by a significant margin, under most settings. Again, we attribute this to the benefit of radar and the effectiveness of our techniques to incorporate it.

Method	Supervision	Input	AbsRel
Monodepth2 [11]	nuScenes: M*	Im	0.1162
R4Dyn [ours]	nuScenes: Mvr	Im Ψ	0.1028

Table 9. Transfer comparison to the KITTI validation set of models trained on nuScenes. Ψ indicates LiDAR sub-sampled like a radar.

A.5.4 Transfer Comparison: nuScenes $\rightarrow$ KITTI

As written in Section 4.1, we used the nuScenes dataset [3] throughout our experiments, as it is to the best of our knowledge the only public dataset with data from a camera, an automotive radar, and a LiDAR to evaluate. On the other hand, the popular KITTI dataset [9] does not provide radar signals and has only limited dynamic scenes, which are the focus of our work. Nevertheless, radar sparsity and missing elevation can be simulated by sub-sampling a LiDAR signal, albeit leaving behind the highly complex radar noise modelling [23], as done also by Lin et al. [21]. Towards this end, we transferred our R4Dyn and Monodepth2 [11] models to the validation set of KITTI, without any fine-tuning, after training them on nuScenes [3]. We achieved this by cropping the KITTI images to the nuScenes format, and down-sampled the KITTI LiDAR to mimic the nuScenes radar: we kept only points within the radar field of view and sub-sampled them to reach the same low density (Section 3.2.1).

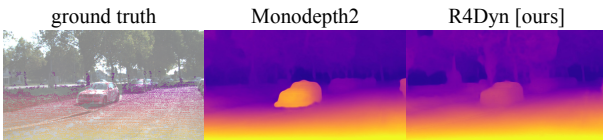


Figure 11. Qualitative example from a transfer from nuScenes to KITTI, without any fine-tuning of our R4Dyn, compared to Monodepth2 [11].

We provide the results of these experiments in Table 9, showing the generalization capability on significantly dif-

ferent data. Our R4Dyn outperforms Monodepth2 in this challenging transfer task. Qualitative results in Figure 11 confirm this and present similar outcomes to those seen in Figure 6, with Monodepth2 underestimating the depth of the oncoming vehicle. Despite the large domain gap and limited dynamic scenes (i.e. our focus), our 11.5% improvement on KITTI is substantial, again showing the effectiveness of our techniques to use radar in input and as weak supervision.

B.4 Appendix: Monocular Depth Estimation in Challenging Conditions (ICCV 2023)

This section includes the supplementary material associated with [86], presented and discussed in Section 5.3.2.

Supplementary Material of:

**Robust Monocular Depth Estimation
under Challenging Conditions**

Stefano Gasperini^{*,1,2} Nils Morbitzer^{*,1} HyunJun Jung¹
Nassir Navab¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² VisualAIs ³ Google

* equal contribution

IEEE/CVF International Conference on Computer Vision (ICCV), 2023

Published version: https://openaccess.thecvf.com/content/ICCV2023/supplemental/Gasperini_Robust_Monocular_Depth_ICCV_2023_supplemental.pdf.

Supplementary Material of: **Robust Monocular Depth Estimation under Challenging Conditions**

Stefano Gasperini^{*,1,2} Nils Morbitzer^{*,1} HyunJun Jung¹
Nassir Navab¹ Federico Tombari^{1,3}

¹ Technical University of Munich ² VisualAIs ³ Google

A. Supplementary Material

The source code of our method, the main trained models reported in the experiments, and the generated translated images are publicly available at <https://md4all.github.io>.

This appendix includes additional details and results. Sections [A.1](#) and [A.2](#) include additional information on the method and the experimental setup, while Sections [A.3](#) and [A.4](#) introduce more results, quantitatively and qualitatively, respectively.

In particular, this appendix is organized as follows:

- Section [A.1.1](#) includes details about our supervised framework.
- Section [A.1.2](#) adds details about our self-supervised baseline.
- Section [A.1.3](#) describes the noise and time-dependent normalization used for our self-supervised models.
- Section [A.2.1](#) further reports details about the experimental setup for image translation.
- Section [A.2.2](#) includes details on the setup used for the nuScenes dataset.
- Section [A.2.3](#) includes details on the setup used for the RobotCar dataset.
- Section [A.2.4](#) adds details about the experimental setup for prior works.
- Section [A.3.1](#) reports a detailed ablation study of our method on both nuScenes and RobotCar.
- Section [A.3.2](#) adds preliminary results with snow and fog on the DENSE dataset.

- Section [A.3.3](#) analyzes the effect of different data distributions among the conditions during training on RobotCar.
- Section [A.3.4](#) compares different configurations of our supervised framework on nuScenes.
- Section [A.3.5](#) looks into quantitative results with rain at nighttime and averages across the various conditions on nuScenes.
- Section [A.3.6](#) compares methods on the test set of nuScenes.
- Section [A.3.7](#) analyzes the performance of the methods at varying distances from the ego vehicle, both on nuScenes and RobotCar.
- Section [A.4.1](#) reports qualitative results of our self-supervised method on nuScenes.
- Section [A.4.2](#) adds qualitative results of our fully-supervised method on nuScenes.
- Section [A.4.3](#) discusses qualitative results of our self-supervised method on RobotCar.
- Section [A.4.4](#) looks into failure cases of our self- and fully-supervised methods, exemplified on nuScenes.
- Section [A.4.5](#) analyzes images generated via image translations for both nuScenes and RobotCar.
- Section [A.5](#) lists attempted and alternative approaches that did not work.

A.1. Additional Details on the Method

A.1.1 Supervised md4all

In the main paper, we mostly focused on the more complex self-supervised setting (Sections [3.1.2](#) and [3.1.3](#)), and we extended our method to the supervised setup (Section [3.2](#)),

* The authors contributed equally.

Contact author: Stefano Gasperini (stefano.gasperini@tum.de).

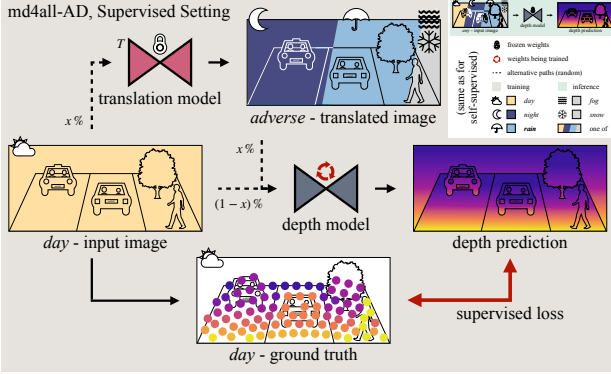


Figure 7. Apart from the type of supervision, our md4all-AD supervised framework works similarly to the md4all-AD self-supervised framework described in Section 3.1.2 and Figure 4, with Always Daytime, No Bad Weather. The depth model is trained with a mix of easy and translated samples, while its supervision is obtained from the ground truth data corresponding only to the easy samples. As in the self-supervised case, inference time is unchanged and performed with a single depth model without specialized branches (top right).

making it the first depth estimation work to explore and address bad weather in supervised monocular settings.

Supervised models learn directly from the ground truth (e.g., LiDAR data). However, in adverse conditions (e.g., rain), the ground truth becomes unreliable (Figure 2). Our supervised md4all aims to eliminate the sources of unreliability in the ground truth by providing a reliable signal in all conditions. As shown in Figure 7, we achieve this with the same principles described for the self-supervised settings: having a single depth model learn robust features agnostic of the condition in input by feeding a mix of easy and hard samples, with the ground truth always corresponding to the easy samples. Therefore, we use the same image translation model to generate adverse images corresponding to the easy ones in the training data. Then, we train the depth model with a mix of original easy and generated adverse inputs. Unlike the self-supervised settings where the training signal came from a pre-trained baseline model (md4all-DD) or the photometric losses (md4all-AD), the training signal is obtained directly from the ground truth data for supervised methods. When translating an image e_i to adverse conditions h_i^c , we use as ground truth for h_i^c the LiDAR data corresponding to e_i .

We associate this supervised method with our AD configuration since no distillation from a pre-trained model occurs (unlike for DD). Moreover, the depth model is trained in the same way as its baseline, i.e., via the ground truth, in an Always Daytime, no Bad Weather fashion, similarly to our self-supervised AD model (Section 3.1.2). Furthermore, the translated images should be used only for those conditions that render the ground truth unreliable. There-

fore, for the experiments, we translated the inputs from *day-clear* to *day-rain*, since the ground truth is unreliable with rain (Figure 2). Still, we used the original *night* inputs since the ground truth is reliable at night (Figure 5).

A.1.2 Self-Supervised Baseline

In this section, we further describe the loss functions used for the baseline of Section 3.1.1. Such baseline is equivalent to Monodepth2 [7] made scale-aware through weak velocity supervision from Guizilini et al. [8].

The photometric loss is the combination of $\mathcal{L}_1$ -loss and SSIM [14], as done in [6]:

$$\begin{aligned}\mathcal{L}_1(I_t, \hat{I}_t) &= \|I_t - \hat{I}_t\|_1 \\ \mathcal{L}_{\text{SSIM}} &= 1 - \text{SSIM}(I_t, \hat{I}_t) \\ pe(I_t, \hat{I}_t) &= (1 - \alpha) \mathcal{L}_1(I_t, \hat{I}_t) + \frac{\alpha}{2} \mathcal{L}_{\text{SSIM}}(I_t, \hat{I}_t)\end{aligned}\quad (1)$$

where $\alpha = 0.85$ is a weight to balance between the two terms. Furthermore, similarly to [7], we account for partial occlusions by only considering the minimum reprojection error:

$$\mathcal{L}_p(I_t, \hat{I}_{s \rightarrow t}) = \min_s pe(I_t, \hat{I}_{s \rightarrow t}). \quad (2)$$

Moreover, following the so-called auto-mask from Monodepth2 [7], we automatically mask out the pixels that do not change appearance across different frames:

$$M_a = \min_s \mathcal{L}_p(I_t, I_s) > \min_s \mathcal{L}_p(I_t, \hat{I}_{s \rightarrow t}). \quad (3)$$

Therefore, the photometric loss is only computed in the areas where $M_a = 1$. Additionally, to encourage local smoothness and preserve sharp edges, we use the following term from [6]:

$$\mathcal{L}_s(I_t, d_t^*) = \frac{1}{N} \sum_{p \in N} \sum_{i \in x, y} |\partial_i d_t^*(p)| e^{-|\partial_i I_t|} \quad (4)$$

where $|\cdot|$ is the absolute value computed element-wise, ∂_x and ∂_y are the gradients in x and y directions, and $d_t^* = d_t^*/\bar{d}_t^*$ is the inverse of the depth prediction normalized by the mean.

As already described in Section 3.1.1, we follow [8] by using a weak velocity supervision $\mathcal{L}_v$ to achieve scale-awareness. This is defined as:

$$\mathcal{L}_v(\hat{\mathbf{t}}_{t \rightarrow s}, \mathbf{t}_{t \rightarrow s}) = \left| \|\hat{\mathbf{t}}_{t \rightarrow s}\|_2 - \|\mathbf{t}_{t \rightarrow s}\|_2 \right| \quad (5)$$

where $\hat{\mathbf{t}}_{t \rightarrow s}$ and $\mathbf{t}_{t \rightarrow s}$ are the predicted and ground truth pose translations, respectively, which can be easily obtained from the available odometry information, through the ego vehicle speed and the time interval across frames.

ID	Method	tr.data	<i>day-clear</i> – nuScenes			<i>night</i> – nuScenes			<i>day-rain</i> – nuScenes		
			absRel	RMSE	δ_1	absRel	RMSE	δ_1	absRel	RMSE	δ_1
A0	md2 [7], <i>all</i>	<i>a: dnr</i>	0.1477	6.771	85.25	2.3332	32.940	10.54	0.4114	9.442	60.58
A1	md2, <i>n</i> real	<i>dn</i>	0.1345	6.575	85.47	2.4536	34.295	11.71	0.1753	7.701	77.13
A2	md2, <i>n</i> transl. _{15%}	<i>dT(n)</i>	0.1390	6.670	85.36	0.2655	9.892	54.44	0.1861	7.800	76.28
A3	md2, <i>day-c</i> only	<i>d</i>	0.1374	6.692	85.00	0.2828	9.729	51.83	0.1727	7.743	77.57
A4	+ v.-sup = b.line	<i>d</i>	0.1333	6.459	85.88	0.2419	10.922	58.17	0.1572	7.453	79.49
A5	+ noise, $\mathcal{L}$ clean	<i>d</i>	0.1428	6.609	84.43	0.2256	9.672	63.50	0.1592	7.619	78.95
A6	+ all <i>n</i> transl.	<i>dT(n)</i>	0.1624	7.042	80.50	0.2214	9.092	67.01	0.1752	8.272	76.41
A7	– pose transl.	<i>dT(n)</i>	0.1597	7.143	81.37	0.2184	8.754	66.90	0.1689	8.210	77.23
ADn	+ <i>day</i> loss only	<i>dT(n)</i>	0.1433	6.954	83.27	0.2230	9.001	68.63	0.1545	7.915	78.36
A9	– time norm.	<i>dT(n)</i>	0.1554	6.949	81.66	0.2121	8.502	67.43	0.1627	7.797	77.62
ADa	ADn + <i>r</i> transl.	<i>dT(nr)</i>	0.1523	6.853	83.11	0.2187	9.003	68.84	0.1601	7.832	78.97
A11	+ <i>day</i> distill.	<i>dT(nr)</i>	0.1387	6.621	84.11	0.1960	8.595	70.08	0.1444	7.355	80.20
DDn	ADn + distill.	<i>dT(n)</i>	0.1302	6.373	85.02	0.1959	8.471	70.14	0.1429	7.312	79.60
DDr	<i>r</i> distill.	<i>dT(r)</i>	0.1323	6.437	85.18	0.2502	11.847	57.02	0.1364	7.100	81.37
DDa	ADa + distill.	<i>dT(nr)</i>	0.1366	6.452	84.61	0.1921	8.507	71.07	0.1414	7.228	80.98
A15	– test time norm.	<i>dT(nr)</i>	0.1367	6.449	84.56	0.1881	8.524	70.65	0.1412	7.234	80.99

Table 4. Ablation study on our self-supervised method on the nuScenes [4] validation set. md2: Monodepth2 [7]. DDa is our self-supervised approach reported throughout this work for nuScenes: md4all-DD trained on *dT(nr)*. Instead, ADa is our self-supervised md4all-AD model. Notation reused from Table 1.

A.1.3 Noise and Normalization

We did not apply either of these two techniques in the supervised setting, i.e., where we applied our method on AdaBins [1], since the LiDAR ground truth provides a strong signal which already enables such robust feature extraction.

For the self-supervised models, we normalize the inputs at training time depending on the time of the day (i.e., day and night). Towards this end, we precompute the mean and variance of the pixel values across the two conditions throughout the dataset and normalize the inputs accordingly. In Tables 4 and 5, we show how this time-dependent normalization has a positive impact at training time, as it aligns the features in a condition-agnostic manner. Additionally, we show that this normalization can be avoided at inference time for similar results. Avoiding it ensures that the operations executed across all conditions are identical at inference time. Thus, after deployment, our method does not require any knowledge about the current weather and illumination settings, which may be hard to define and may intersect with other conditions (e.g., wet ground without rain or dusk). Nevertheless, due to the relatively small difference during inference, we used time-dependent normalization for self-supervised models unless otherwise noted.

Furthermore, in the case of camera sensors delivering significant noise levels (e.g., nuScenes [4]), we augment the inputs of self-supervised models with heavy noise. The noise is randomly applied to 50% of the inputs, regardless of their condition. This helps to learn more robust features. When the noise is used, we compute the losses on the sam-

ples without noise. Specifically, we generated the noise by adding to the image a random pattern following the uniform distribution [0.005, 0.05], then clamped the pixel values to [0, 1], thereby ensuring that the input remains within a valid range.

A.2. Additional Details on the Experimental Setup

A.2.1 Day-to-adverse Translation

For the experiments, we focused on two adverse conditions in nuScenes [4] (i.e., *rain* and *night*) and one in RobotCar [10] (i.e., *night*), alongside the standard conditions *day-clear* / *day*. Towards this end, we trained two different ForkGAN [16] models for nuScenes, one for each condition, and one for RobotCar, to enable translations from *day-clear* to each challenging condition, tailored to each dataset. For the RobotCar dataset, we trained the GAN using the 34128 daytime samples from the scene 2014-12-09-13-21-02 and the 32585 nighttime samples from 2014-12-16-18-44-24. The dataset offered enough samples to train the image translation model thanks to the high frame rate. Instead, the nuScenes dataset only provides 6951 samples for *day-rain* and 4706 for *night*, which are insufficient for the GAN to learn such day-to-adverse translation. Therefore, to learn the transition from *day-clear* to *day-rain*, we additionally used all *day-rain* samples from the nuImages dataset [4] resulting in a total number of 19857 *day-rain* frames. We balanced this with the 19685 *day-clear* images of the nuScenes training set. Since the nuImages dataset does not provide any metadata about the weather condition,

Method	tr.data	day – RobotCar				night – RobotCar			
		absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
Monodepth2 [7]	d	0.1196	0.670	3.164	86.38	0.3029	1.724	5.038	45.88
WSGD [12]	$a: dn$	0.1760	1.603	6.036	75.00	0.1740	1.637	6.302	75.40
[ours] baseline	d	0.1209	0.723	3.335	86.61	0.3909	3.547	8.227	22.51
[ours] md4all-AD	$dT(n)$	0.1113	0.707	3.248	88.02	0.1223	0.851	3.723	85.77
[ours] md4all-DD	$dT(n)$	0.1128	0.648	3.206	87.13	0.1219	0.784	3.604	84.86
[ours] md4all-DD w/o test time norm.	$dT(n)$	0.1129	0.640	3.190	87.02	0.1256	0.824	3.703	83.87
[ours] md4all-AD w/ LiDAR scaling	$dT(n)$	0.1192	0.747	3.184	86.81	0.1275	0.834	3.641	86.15
[ours] md4all-DD w/ LiDAR scaling	$dT(n)$	0.1133	0.642	3.052	87.45	0.1230	0.739	3.439	86.41

Table 5. Evaluation of self-supervised works on the RobotCar [10] test set up to 50 meters. Different configurations of our method are compared with Monodepth2 and WSGD. LiDAR scaling indicates the use of LiDAR data at test-time to scale the predictions (as done by Monodepth2 and WSGD), which is equivalent to * in the supervision notation of Table 1 and 2. The method highlighted in bold is reported throughout this work for RobotCar as md4all-DD trained on $dT(n)$. This Table extends Table 2.

we manually labeled all its samples with their respective weather condition. Nevertheless, *night* samples are insufficient in nuScenes and nuImages (14302) to train a GAN. For this reason, we first trained the *day-clear* to *night* translation model on BDD100K [15], which includes 36728 *day* and 27971 *night* images in its training set. Then, we fine-tuned it on the available nuScenes *night* samples from the training set.

A.2.2 nuScenes

For the depth experiments on nuScenes [4], we followed the setup of R4Dyn [5], using the official data splits and evaluating up to 80 meters comparing the predictions with a single LiDAR scan. As in R4Dyn, we discarded static frames (i.e., where the ego vehicle is stationary) for self-supervised models. While a single scan is highly sparse compared to the dense depth prediction, it limits the artifacts introduced by accumulating multiple scans over time for denser ground truth (e.g., due to moving objects and changing perspectives). We augmented the inputs with heavy noise for self-supervised models to mimic that in the night samples. For the supervised setting, learning from such a sparse signal means reducing the workload needed for producing the ground truth, albeit rendering it more challenging. As it is standard for supervised setups, the models do not learn the depth of the unreachable areas for the ground truth sensor (e.g., the sky for LiDAR). All qualitative images and quantitative results reported in the main paper and this supplementary material are from the validation set unless otherwise noted (e.g., test set in Table 14).

A.2.3 RobotCar

For the experiments on RobotCar [10], we followed the setup of WSGD [12] using the six sequences in the 2014-12-09-13-21-02 traversal as daytime samples, and the six

sequences in the 2014-12-16-18-44-24 traversal as night-time ones. Since the peculiarity of RobotCar is that it was recorded by driving over the same route multiple times over a year, a training-test split with non-overlapping drives is required to properly assess the models’ generalization capabilities. Therefore, we used the split provided by WSGD.

As in [13], we used the left images of the front stereo-camera (Bumblebee XB3), of which we removed the bottom 20% (i.e., ego vehicle bonnet), and the ground truth data from the LMS front LiDAR sensor. We used the official toolbox to accumulate multiple LiDAR scans and project them to the input images. Towards this end, we used visual odometry, as recommended by the official documentation of the dataset, and a time margin of $\pm 4e6$ from the origin timestamp, as in [13]. As commonly done for self-supervised methods, we discard static frames thresholding the translation provided by the visual odometry. We did not apply heavy noise for RobotCar as the night samples did not exhibit it. Furthermore, since the RobotCar camera occasionally suffers from inconsistent illumination across neighboring frames, we discarded these too. Specifically, we removed all triplets where the keyframe’s mean RGB value is ≥ 0.9 , or the RGB mean value difference between two consecutive frames is > 0.05 . In addition, only the images with a corresponding LiDAR ground truth could be evaluated.

For the experiment with degraded translations via random erasing (Table 3), we applied it randomly to 10% of the inputs, with a patch sized randomly between 5% and 10% of the input dimensions, placed randomly within the image, with an aspect ratio between 0.3 and 3.3. When applying random erasing, the performance of md4all marginally improved in terms of δ_1 by 0.32% on *all* (absRel slightly decreased as shown in Table 3), thanks to the augmentation and regularization effect introduced by the patches. Throughout the main paper and this supplementary mate-

Method	<i>day-clear</i>		<i>fog</i>		<i>snow</i>	
	absRel	δ_1	absRel	δ_1	absRel	δ_1
md2 [7]	0.1642	82.35	0.1698	81.97	0.1798	76.68
[ours]	0.1520	83.54	0.1524	83.36	0.1788	77.93

Table 6. These are only preliminary results (details in Section A.3.2). Evaluations with *snow* and *fog* on the DENSE dataset [2]. AbsRel and δ_1 are reported for each condition. Monodepth2 [7] (md2) trained on *day-clear* is compared with our md4all-DD trained on *day-clear* plus translated images to *fog* and *snow* ($x = 66\%$).

rial, all qualitative images and quantitative results are from the test set defined by WSGD.

A.2.4 Prior Works

For prior works on RobotCar, we reported the values computed by Vankadari et al. [12], who retrained RNW [13] on a non-overlapping split (which inherently reduced the scores), and also re-evaluated DeFeatNet [11] and ADIDS [9] on the same test split (again reducing the scores). Among works focusing on depth estimation in the dark, only RNW reported its results on the more challenging nuScenes dataset. However, since the authors reported their scores on a different, custom split, we retrained their model on the official split. For this reason, the results of RNW differ from those reported directly by Wang et al. in their paper [13]. Nevertheless, the difference is relatively small as RNW reported at *night*: absRel of 0.3150 (0.3333 from our experiment with RNW), sqRel of 3.793 (4.006), RMSE of 9.6408 (10.098), and δ_1 of 50.81 (43.72). While this performance gap should be attributed to the different data splits used, it does not affect the comparisons since our models performed significantly better than what Wang et al. reported in their paper [13], both quantitatively and qualitatively. Furthermore, on nuScenes, we also report the values of R4Dyn [5] and PackNet-SfM [8], as provided to us by the authors of [5]. Additional related works tackling adverse conditions exist (Section 2.2.1). Still, their lack of open-source code or their use of unconventional and unclear experimental setups prevented us from directly comparing with their methods.

A.3. Additional Quantitative Results

A.3.1 Ablation Study

In Table 4, we report an ablation study over the main components of our method.

We started from a Monodepth2 [7] trained on the entire training set of nuScenes [4] (A0), meaning all available conditions. A0 performed poorly under adverse conditions due to the difficulty of establishing pixel correspon-

Method	<i>avg/all</i>		<i>day</i>		<i>night</i>	
	absRel	δ_1	absRel	δ_1	absRel	δ_1
md2 [7]	0.2122	65.92	0.1196	86.38	0.3029	45.88
70d - 30n	0.1189	86.39	0.1138	87.80	0.1239	85.01
50d - 50n	0.1174	85.99	0.1128	87.13	0.1219	84.86
30d - 70n	0.1221	85.86	0.1168	87.16	0.1273	84.59

Table 7. Impact of different training data distributions between *day* (d) and *night* (n) samples by varying the parameter x . AbsRel and δ_1 are reported on the test set of the RobotCar dataset [10]. Different distributions are shown for md4all-DD, a . In the rest of this work, the balanced 50d-50n configuration (i.e., $x = 50\%$) was used.

dences across consecutive *night* and *rain* frames. A0 delivered scores and errors similar to those reported for Monodepth2 by prior works in their papers, such as RNW [13] and WSGD [12]. Furthermore, the outputs of A0 exhibit the same issues shown by RNW and WSGD in their qualitative comparisons (e.g., holes in the ground), which are not present and much improved when training Monodepth2 only on *day-clear*, as reported throughout this work (A3). In particular, with A0-A3, we show how the standard Monodepth2 performs substantially better than the complex RNW overall and significantly better than WSGD in the daytime (Table 5).

A1 is a Monodepth2 model trained on *day-clear* and *night* (i.e., everything excluding *day-rain*). A1 performed similarly to A0 at *night*, but significantly better with *rain*. Additionally, it can be seen how training on *day-rain* samples negatively affected the *day-clear* performance (A0) while excluding such rainy samples improved in standard conditions (A1). Then, A2 is another Monodepth2 model, trained on *day-clear* and the translated *night* samples we generated with the GAN. A2 was fed a mix of *day-clear* and generated *night* ones with $x = 15\%$, to resemble the day-night distribution of the training set (used by A1). The comparison of A1 with A2 shows mainly two aspects about the translated images (Figure 17): they are not entirely realistic, and, unlike the real ones, they do not prevent establishing the pixel correspondences. If the generated samples were completely realistic (i.e., like the real *night* ones from nuScenes), there would have been a much smaller difference between A1 and A2. In particular, the generated images do not fully resemble the real *night* ones (Figure 17), especially for the noise, which is more consistent throughout the generated frames compared to the real ones, and the darkness levels, with images that are not as black as the real *night* ones of nuScenes. This lack of realism in the generated images is the reason for the performance improvement of A2 at *night* compared to A1.

Similarly, WSGD [12] showed the importance of denoising *night* images, with noise detrimental to the models.

Method	tr.data	<i>day-clear</i> – nuScenes			<i>night</i> – nuScenes			<i>day-rain</i> – nuScenes		
		absRel	RMSE	δ_1	absRel	RMSE	δ_1	absRel	RMSE	δ_1
AdaBins [1]	<i>a: dnr</i>	0.1384	5.582	81.31	0.2296	7.344	63.95	0.1726	6.267	76.01
AdaBins [1]	<i>d</i>	0.1138	4.805	87.98	0.3336	14.002	45.77	0.1540	6.119	81.20
[ours] md4all-AD, <i>rain</i>	<i>dT(r)</i>	0.1052	4.621	89.58	0.2644	10.749	55.51	0.1380	6.030	83.32
[ours] md4all-AD, all	<i>dnT(r)</i>	0.1206	4.806	88.03	0.1821	6.372	75.33	0.1562	5.903	82.82

Table 8. Additional evaluation of LiDAR-supervised methods on the nuScenes [4] validation set. This table adds the second and third lines compared to Table 1.

Method	<i>avg/all</i> – nuScenes				<i>night-rain</i> – nuScenes				<i>d-clear</i>	<i>night</i>	<i>d-rain</i>
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1	sqRel	sqRel	sqRel
md2 [7], <i>d</i>	0.1576	2.002	7.164	80.49	0.3148	3.001	9.523	46.72	1.820	2.879	2.296
R4Dyn [5], <i>d</i> (radar)	0.1365	1.830	6.957	84.01	0.2431	2.945	10.055	56.95	1.661	2.889	1.938
RNW [13], <i>dn</i>	0.2931	3.557	9.304	55.13	0.3400	4.783	10.189	44.68	3.433	4.066	3.796
baseline, <i>d</i>	0.1480	2.032	7.065	82.08	0.2684	3.368	10.664	53.54	1.738	2.776	2.273
md4all-AD, <i>dT(nr)</i>	0.1602	2.245	7.226	81.02	0.2470	3.442	9.153	65.17	2.141	2.991	2.259
md4all-DD, <i>dT(nr)</i>	0.1429	1.828	6.782	82.67	0.2143	2.628	8.376	68.03	1.752	2.386	1.829
AdaBins [1], <i>a</i>	0.1604	1.103	5.868	78.72	0.2343	1.704	7.088	61.62	0.980	1.773	1.249
md4all-AD, <i>dnT(r)</i>	0.1328	0.952	5.139	85.92	0.1967	1.632	6.423	71.67	0.821	1.525	1.199

Table 9. Evaluation of fully-supervised (based on AdaBins [1]) and self-supervised methods (based on md2: Monodepth2 [7]) on the nuScenes [4] validation set. The models are the same as in Table 1. This table complements Table 1 with an evaluation on *all* conditions combined, as well as the most challenging *night-rain*.

Since the translated images do not exhibit the same kind of noise and reflections as the real ones and are particularly unrealistic when translating a sunny sample (Figure 17), A2 was able to establish pixel correspondences across the translated samples to a certain extent. Additionally, as x randomizes the condition of each input independently, with A2, the translations also introduce a regularization effect as data augmentation. A3 was also a Monodepth2 model but trained only on the *day-clear* samples. As shown in Table 1, this improves significantly compared to training on all conditions (A0) due to the impossibility of establishing correspondences at *night* for A0.

The weak velocity supervision [8] (A4) improved significantly over A3, thanks to better pose estimates. Compared to A0-A3, which need ground truth median scaling at test time, A4 is scale-aware and does not use it. With A5, we added heavy noise (consistently throughout the triplets) but computed the losses on the clean samples (i.e., without noise). This made it worse for *day-clear* and *day-rain*, but improved for *night* compared to A4. The motivation for A5 develops from the intense noise present in the *night* samples of nuScenes (Figure 2), which may confuse the models. We did not apply this under supervised settings (i.e., our method on AdaBins). The improvement seen with adding noise while computing the loss on the inputs without noise paved the way for the concept of our AD model. With A6, we added the translated images generated with the GAN from day to night ($x = 50\%$) to the training data.

This made it worse than A0 for *day-clear*, but similarly to A2, it improved for *night* due to the lack of realism of the generated samples, which allowed to establish pixel correspondences. For A6 (and A2), perfectly realistic generated samples would have been detrimental to learning.

With A7, we did not feed the translated images to the pose model but only to the depth one. This guarantees reasonable pose estimates, which improve the task at hand under all three conditions. Then, with ADn, we computed the losses only on the *day-clear* e_i samples, corresponding to the translated ones given as input. This significantly improved the model performance on *day-clear*, reaching a level similar to A3 (i.e., only a marginal degradation on the standard conditions). It should be noted that if the translated images perfectly mimicked the real *night* ones, A2, A6, and A7 would have performed relatively poorly, i.e., similarly to A0 and A1 at *night*. In the case of perfect day-to-night translations, always computing the loss only on the *day-clear* e_i samples (as in ADn, instead of calculating it on the translated ones, as in A2, A6, and A7) would have had a significantly positive impact at *night*.

With A9, we removed the time-dependent normalization from ADn, which was used from A6 to ADn. This shows that this technique benefits both *day-clear* and *day-rain*, as it helps construct a unified representation for all conditions. With ADa, we incorporated day-to-rain translations to ADn alongside the day-to-night images ($x = 66\%$, i.e., one-third for each condition). This delivered a similar performance

Method	40m – <i>day-clear</i> – nuScenes				40m – <i>night</i> – nuScenes				40m – <i>day-rain</i> – nuScenes			
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
md2	0.1095	0.796	3.535	88.89	0.2401	1.640	5.842	60.40	0.1405	1.083	4.259	82.64
b.line	0.1131	0.932	3.624	89.46	0.2118	1.816	6.476	63.47	0.1333	1.200	4.397	83.66
AD	0.1306	1.074	3.866	86.95	0.1907	1.670	5.414	73.24	0.1329	1.083	4.332	83.54
DD	0.1173	0.877	3.592	88.22	0.1672	1.322	5.025	75.50	0.1190	0.927	4.036	85.37

Table 10. Evaluation up to 40 meters of self-supervised approaches on the validation set of nuScenes [4]. md2: Monodepth2 [7] trained on *d*. b.line: baseline trained on *d*. AD: md4all-AD trained on $dT(nr)$. DD: md4all-DD trained on $dT(nr)$. The models are the same as in Table 1.

Method	40m – <i>avg/all</i> – nuScenes				40m – <i>night-rain</i> – nuScenes			
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
md2	0.1276	0.926	3.882	85.04	0.2768	2.020	6.604	53.26
b.line	0.1262	1.063	4.034	85.93	0.2466	2.301	7.440	57.13
AD	0.1369	1.135	4.096	85.03	0.2184	2.064	6.066	68.96
DD	0.1225	0.930	3.807	86.49	0.1925	1.653	5.661	71.38

Table 11. Evaluation up to 40 meters of self-supervised approaches on the validation set of nuScenes [4]. md2: Monodepth2 [7] trained on *d*. b.line: baseline trained on *d*. AD: md4all-AD trained on $dT(nr)$. DD: md4all-DD trained on $dT(nr)$. The models are the same as in Table 1. This table complements Table 10.

to ADn (e.g., improved for *night*, and improved the RMSE, with a worse absRel for *day-clear*). As for Table 1, the LiDAR ground truth is not fully reliable for *day-rain* and also significantly sparser than for *day-clear* (Figure 2). With A11, we added the day distillation loss (Equation 1) on the translated inputs while keeping the standard losses for the *day-clear* inputs. This combination improved across the board. Then, with DD, we simplified the training process by using only the day distillation loss for all inputs (including *day-clear*). Thus, DDn does this for *day-clear* and *night*, DD_r does it for *day-clear* and *day-rain*. Our day distillation provides a dense and reliable signal (from A4 inferring only on *day-clear* samples), improving the errors and metrics across the board.

Finally, with A15, we show the impact of avoiding the time-dependent normalization at test time. Compared to DD_a, A15 does not apply such time-dependent normalization at inference time but only at training time. A15 obtains comparable results throughout the various settings. Instead, as shown with A9, the time-dependent normalization is helpful at training time. After training, our model has learned robust features agnostic to the condition, allowing it to perform similarly regardless of the image normalization applied at test time. This demonstrates how our method does not need any condition-specific setups at inference time to deliver robust predictions, clearly separating our md4all from previous works requiring custom branches for each condition. In the supervised settings (i.e., our method applied on AdaBins), we did not perform any time-dependent normalization since the strong LiDAR supervision is enough to learn depth estimation at *night*.

In Table 5, we report various configurations of our method on the RobotCar [10] dataset. As for nuScenes (A15 in Table 4), we show that not applying the time-dependent normalization at test-time (i.e., executing the same operations with the same setup across the different conditions) does not negatively affect the predictions, achieving comparable results. Furthermore, we show how the results change when applying the median scaling via LiDAR data at test time. This technique is used by Monodepth2 [7], WSGD [12], and most other methods compared in this work. Our model does not need such scaling via ground truth data, thanks to its scale awareness learned via the weak velocity supervision introduced by PackNet-SfM [8].

Nevertheless, accurate scaling can further improve the results, especially at *night*. Compared to nuScenes, the RobotCar dataset provides less precise odometry information, causing difficulties for the baseline and our models to learn the correct scaling. This can be seen by the improved scores at *night* when applying the median scaling via LiDAR data. With reliable scaling via the ground truth data, md4all-DD outperforms md4all-AD.

AD and DD While the benefit of our DD configuration over AD is evident for nuScenes, the gap is not as significant for RobotCar, with the two delivering comparable results (Table 5). This difference can be attributed to various reasons. First of all, nuScenes is more challenging, as demonstrated by the lower scores obtained by the models across all conditions, especially at *night*. Thus, the improvements of DD over AD might be reduced for RobotCar since AD already achieves solid results. The higher amount of images

Method	60m – <i>day-clear</i> – nuScenes				60m – <i>night</i> – nuScenes				60m – <i>day-rain</i> – nuScenes			
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
md2	0.1283	1.387	5.447	86.15	0.2739	2.469	8.444	53.40	0.1623	1.779	6.312	79.23
b.line	0.1279	1.522	5.422	86.91	0.2348	2.779	9.502	59.28	0.1506	1.833	6.267	80.73
AD	0.1461	1.745	5.744	84.23	0.2113	2.526	7.789	69.92	0.1519	1.774	6.421	80.28
DD	0.1310	1.419	5.364	85.65	0.1859	2.029	7.377	72.10	0.1347	1.463	5.938	82.24

Table 12. Evaluation up to 60 meters of self-supervised approaches on the validation set of nuScenes [4]. md2: Monodepth2 [7] trained on *d*. b.line: baseline trained on *d*. AD: md4all-AD trained on $dT(nr)$. DD: md4all-DD trained on $dT(nr)$. The models are the same as in Table 1.

Method	60m – <i>avg/all</i> – nuScenes				60m – <i>night-rain</i> – nuScenes			
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
md2	0.1483	1.558	5.886	81.76	0.3086	2.782	8.790	47.76
b.line	0.1423	1.698	5.966	83.15	0.2648	3.130	9.858	54.09
AD	0.1536	1.827	6.057	82.16	0.2402	2.983	8.233	65.79
DD	0.1371	1.487	5.658	83.74	0.2101	2.368	7.661	68.58

Table 13. Evaluation up to 60 meters of self-supervised approaches on the validation set of nuScenes [4]. md2: Monodepth2 [7] trained on *d*. b.line: baseline trained on *d*. AD: md4all-AD trained on $dT(nr)$. DD: md4all-DD trained on $dT(nr)$. The models are the same as in Table 1. This table complements Table 12.

available on RobotCar to learn the translation task led to more realistic translations than nuScenes (Section A.4.5). Then, the less precise odometry information of RobotCar impacted the performance of the baseline through weak velocity supervision. Therefore, the baseline possibly learned wrong poses. This is not the case on nuScenes (Table 4), where the baseline (A4) improved significantly over Monodepth2 (A3). This did not happen for RobotCar. Since our md4all-DD learns to mimic the baseline via knowledge distillation, our model is directly affected by the weaker baseline in RobotCar, delivering similar results to AD. Instead, in nuScenes the gap between AD and DD is substantial throughout the conditions.

As shown with Monodepth2 [7] and AdaBins [1], our method is widely flexible and applicable to different architectures and types of supervision. While being out of the scope of this work, our approach can be seamlessly applied to other self-supervised or supervised frameworks, such as PackNet-SfM [8], since we do not alter the model architecture, but only its training scheme. In particular, to apply the proposed md4all to an existing depth estimation method, no structural changes are needed, as it is sufficient to feed to the model the translated images $x\%$ of the time during training.

A.3.2 DENSE Dataset: *Snow* and *Fog*

Disclaimer: First, please consider that these are only preliminary experiments and that we have not yet explored these conditions and models to the same extent as *night* and *rain* in the rest of this work. Nevertheless, we report them here as they provide interesting insights.

In Table 6, we show a first attempt to tackle the prob-

lem of monocular depth estimation in the presence of snow or fog with the DENSE dataset [2]. While our md4all performed better than the standard Monodepth2 [7] across the board, the improvement is relatively small compared to the other datasets and conditions explored (e.g., Tables 1 and 2). There are multiple reasons for this, explained below.

An impactful aspect to be considered is related to the available data. The condition boundaries are somewhat blurry. Overcast *day-clear* samples can be similar to light *fog* or light *snow*. This is problematic for the GAN used for image translation, which cannot distinguish the distributions and adequately learn the translation task.

Furthermore, the term snow is generic and includes various scenarios, such as light snow, heavy snow, blizzard, partly covered ground, fully covered ground, piles of snow, or wet ground with light snow falling. These settings differ substantially, but all belong to the same *snow* condition. This high variability is problematic for the translation task. While this issue can occur similarly with *night* and *rain* too, it is not as severe, and the diversity is more limited.

Another significant issue is the amount of usable image data for these conditions, which is insufficient to properly learn the translation task with ForkGAN [16]. As we did for *night* for nuScenes (Section 4.1), also for DENSE, we had to supplement with extra snow images taken from another dataset: Boreas [3]. We trained the snow ForkGAN with 17591 *day-clear* and 8443 *snow* samples from DENSE, plus 25036 *day-clear* and 26437 *snow* samples from Boreas for the pre-training. While supplementing with data from Boreas helped, the number of images from DENSE was relatively low compared to nuScenes and RobotCar, prevent-

Method	test – 40m – nuScenes				test – 60m – nuScenes				test – 80m – nuScenes			
	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
md2	0.1162	0.811	3.701	87.59	0.1376	1.364	5.650	84.08	0.1465	1.755	6.941	82.69
RNW	0.2500	2.237	6.114	62.63	0.2781	3.420	9.222	57.50	0.2900	4.169	11.289	55.56
b.line	0.1126	0.840	3.827	87.17	0.1275	1.364	5.747	84.33	0.1332	1.679	6.938	83.22
AD	0.1214	0.881	3.851	86.99	0.1353	1.387	5.731	84.14	0.1409	1.691	6.915	83.00
DD	0.1090	0.757	3.606	88.29	0.1221	1.204	5.418	85.57	0.1277	1.503	6.607	84.46
AdaBins	0.1434	0.617	3.233	83.09	0.1494	0.852	4.689	80.67	0.1532	1.055	5.849	79.62
AD sup.	0.1182	0.641	3.279	89.40	0.1221	0.785	4.293	87.98	0.1240	0.887	5.021	87.33

Table 14. Evaluation up to 40, up to 60, and up to 80 meters of self-supervised approaches on the test set of nuScenes [4]. All conditions are evaluated here (*avg/all*). AD sup.: our md4all-AD trained on $dnT(r)$ applied on AdaBins [1]. md2: Monodepth2 [7] trained on d . b.line: baseline trained on d . AD: md4all-AD trained on $dT(nr)$. DD: md4all-DD trained on $dT(nr)$. Time-dependent normalization was not applied to obtain these results as nuScenes provides no condition annotations for the test set. The models are the same as in Table 1.

ing effective translations.

Training data for the depth models was 7947 *day-clear* keyframes for DENSE. These keyframes were relatively few (15129 were used for nuScenes and 17790 for RobotCar), and they were extracted from short sequences, so they did not exhibit high variability. This reduced the depth estimation performance of the models. The validation set was also small with only 289 for *day-clear*, 1281 for *snow*, and 543 for *fog*. DENSE contains more images, but those were not usable due to various reasons, e.g., they were captured by different sensors.

Furthermore, as with *rain* (Figure 2), the LiDAR is not reliable in the presence of *snow* or *fog*, as it often captures snowflakes, fog particles, or is even obstructed by the snow accumulated on the sensor itself. We mitigated this problem by filtering the erroneous LiDAR points via clustering, but we could not eliminate all problematic measurements. As seen for *rain* on nuScenes, in adverse conditions the LiDAR sensor is unable to collect measurements at further distances (e.g., Figure 5 *rain* vs. *night* ground truth depth). Therefore, we could only evaluate a limited set of points at a closer distance. We used a single LiDAR scan as ground truth. Additionally, among the snow data, many samples were recorded in remote areas with relatively flat surroundings. Considering the limited distance and the flat surroundings, a model overfitting on flat ground may seem erroneously adequate by obtaining good quantitative results.

Additionally, for the weak velocity supervision of our baseline, we exploited the information from the CAN bus, as provided by the authors of DENSE. We used the vehicle speed and the frame rate to compute the camera translation between the frames. However, since the vehicle speed is provided as single value for each short sequence, the camera poses could only be coarsely approximated. This likely affected the performance of the baseline, hence that of our md4all-DD too. Furthermore, we used the CAN data to discard static inputs (i.e., stationary ego vehicle) and those where the ego vehicle is turning. We filtered the latter when

the steering wheel angle exceeded 20° . This filtering led to the numbers indicated above.

All these points should be considered when evaluating these preliminary results on DENSE.

First of all, regarding Table 6, it can be seen how the *day-clear* results are not as good as those seen for nuScenes (Table 1) or RobotCar (Table 2). This could be attributed to DENSE containing more challenging data. More likely, it is due to the inability of the models to properly generalize on DENSE due to the relatively low diversity in the training data and the limited amount of training samples. Therefore, our md4all-DD learned from a weak baseline which could not correctly estimate depth in standard conditions. Nevertheless, our md4all-DD outperformed Monodepth2 in standard settings, thanks to the regularization effect of our translations.

In the table, we report light-fog for *fog* and full-coverage or currently snowing for *snow*. We opted for light-fog since dense-fog exhibited too few LiDAR points for the evaluation, all at relatively close distances (easier). Instead, light-fog allowed for a more thorough assessment at further distances. For reference, all results were better with dense-fog than light-fog. For *snow*, we selected those with full-coverage or weather metadata snow. This is because, among the annotated conditions, they had the most precise boundaries with other conditions.

Moreover, both models perform similarly with *fog* as in ideal settings (i.e., *day-clear*). While this hints that fog is not as challenging as rain or night (Tables 1 and 2), the values are also affected by the limited distance of the ground truth used for the evaluation. Therefore, the performance may degrade significantly at further distances due to the fog preventing seeing the details, but that cannot be evaluated. Nevertheless, already at the available ground truth distances, our model outperformed Monodepth2, on which ours is based.

Despite the limited distance of the ground truth, *snow* appears more challenging than *fog*, causing a significant

drop in performance compared to the ideal settings (i.e., *day-clear*). With snow, the limited data available to learn proper translations substantially impacted our method’s performance, which obtained only slightly better scores than Monodepth2.

Stronger condition boundaries (e.g., more precise annotations) and more training data would significantly improve the translations and our method’s outcomes. Furthermore, depth ground truth reaching further distances without any artifacts would allow us to assess the actual performance of the models. While these factors would contribute to a more considerable gap between the proposed md4all and Monodepth2, the issues with the translations also highlight the limitations of our approach: the difficulty in collecting adverse data that would lead to solid results (e.g., Table 2 with RobotCar [10]).

Due to the substantial limitations encountered with this data, the DENSE dataset is unsuitable for depth estimation. However, we used it to provide these preliminary results with snow and fog. New real data with artifact-free long-distance ground truth is needed to properly explore monocular depth estimation in these conditions.

A.3.3 Different Distributions of Conditions

In Table 7, we explore the effect of different data distributions among the conditions during training. We vary this via the parameter x . In the rest of this work, x was selected to equally distribute the inputs among the conditions. So for RobotCar 50% for half for *day* and half for *night* (i.e., 50d - 50n in the table); for nuScenes 66% corresponding to one third for each of *day-clear*, *night*, and *rain*; one third each also for the DENSE dataset.

While intuitively increasing the amount of *day* images could improve the performance on *day*, this is not the case by randomizing via x at each training sample independently. This is because with enough epochs, our model sees all images in all conditions, so the training data remains unchanged, causing only minor differences as the model might be fed more or fewer translations (Table 7). For *day*, beyond the observed regularization effect (e.g., Table 2), there is little room for gains as long as the baseline model B to distill from remains the same. Instead, if x affected which portion of the training data is translated, it would have a more significant impact than shown in Table 7. In that case, seeing too many or too few translated images may impair the performance as the model does not experience enough of the ideal settings or not enough adverse conditions to tackle them properly.

A.3.4 Supervised Configuration Comparisons

Table 8 reports a comparison of different supervised configurations of AdaBins [1] and our md4all-AD applied on

AdaBins. Specifically, AdaBins trained only on *day-clear* resulted in a significant improvement on *day-clear* and *day-rain* compared to the AdaBins trained in all conditions (i.e., a). Analogously, our model trained on *day-clear* and translated *day-rain* samples performed better than ours trained on all but substantially worse at *night*. As seen in the self-supervised case, our model trained in all conditions outperformed the baseline across the board (i.e., AdaBins trained on all), thereby not introducing any trade-off while improving in adverse conditions over the model it is based on.

A.3.5 nuScenes Night-rain and Average

In Table 9, we report results on more conditions of nuScenes [4], such as the most difficult *night-rain* and an average over *all*, alongside the sqRel errors not fitting in Table 1 (due to the limited space available). *All* is not computed as an average on the various conditions but rather as an average of the performance on each sample (i.e., *night* counts marginally, accounting for only 10% of the images). Our model outperforms the baseline AdaBins across the board for the supervised case. Similarly, in the self-supervised setting, our md4all improved significantly over the baseline and Monodepth2 [7], second only in ideal conditions (*day-clear*) to the radar-based R4Dyn [5].

A.3.6 nuScenes Test Set

Table 14 reports errors and metrics on the official test set of nuScenes [4]. Since no metadata about the weather and illumination conditions is available for the test set, the values are to be considered for *all* conditions combined. As for the validation set, results were computed with the available ground truth data (i.e., LiDAR). The test set of nuScenes is more accessible than its validation set, with the models achieving better performance in the former across all three distance ranges. For these reasons, we focused on the validation set, and we report the test set results here for completeness. Nevertheless, our supervised and self-supervised models performed better than the models they are based on (i.e., AdaBins and Monodepth2, respectively) across the various conditions and depths.

A.3.7 Evaluation over Different Distances

For completeness, we also include the results for self-supervised models computed up to different distance ranges, namely up to 40, 60, and 80 meters. This complements the values reported in the main paper (up to 80 meters for nuScenes, and up to 50 for RobotCar) and should ease comparisons with future works. These results are reported across the following tables:

- Tables 10, 11, and 14 for nuScenes up to 40 meters.

Method	tr.data	40m – <i>day</i> – RobotCar				40m – <i>night</i> – RobotCar			
		absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
Monodepth2 [7]	<i>d</i>	0.1181	0.614	3.034	86.51	0.3022	1.702	4.984	45.97
[ours] baseline	<i>d</i>	0.1198	0.678	3.229	86.69	0.3908	3.541	8.206	22.52
[ours] md4all-AD	<i>dT(n)</i>	0.1099	0.650	3.130	88.10	0.1203	0.762	3.531	85.87
[ours] md4all-DD	<i>dT(n)</i>	0.1120	0.618	3.125	87.18	0.1206	0.723	3.479	84.92

Table 15. Evaluation of self-supervised works on the RobotCar [10] test set up to 40 meters. The models are the same as in Table 2.

Method	tr.data	60m – <i>day</i> – RobotCar				60m – <i>night</i> – RobotCar			
		absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
Monodepth2 [7]	<i>d</i>	0.1201	0.698	3.215	86.38	0.3029	1.728	5.045	45.88
[ours] baseline	<i>d</i>	0.1213	0.746	3.382	86.61	0.3909	3.548	8.228	22.51
[ours] md4all-AD	<i>dT(n)</i>	0.1116	0.731	3.291	88.02	0.1231	0.903	3.812	85.76
[ours] md4all-DD	<i>dT(n)</i>	0.1130	0.661	3.234	87.13	0.1225	0.824	3.664	84.86

Table 16. Evaluation of self-supervised works on the RobotCar [10] test set up to 60 meters. The models are the same as in Table 2.

Method	tr.data	80m – <i>day</i> – RobotCar				80m – <i>night</i> – RobotCar			
		absRel	sqRel	RMSE	δ_1	absRel	sqRel	RMSE	δ_1
Monodepth2 [7]	<i>d</i>	0.1203	0.718	3.245	86.38	0.3030	1.729	5.046	45.88
[ours] baseline	<i>d</i>	0.1214	0.759	3.404	86.61	0.3909	3.548	8.228	22.51
[ours] md4all-AD	<i>dT(n)</i>	0.1118	0.742	3.308	88.02	0.1236	0.952	3.880	85.76
[ours] md4all-DD	<i>dT(n)</i>	0.1131	0.666	3.243	87.13	0.1229	0.865	3.713	84.86

Table 17. Evaluation of self-supervised works on the RobotCar [10] test set up to 80 meters. The models are the same as in Table 2.

- Tables 12, 13, and 14 for nuScenes up to 60 meters.
- Tables 1, 8, 9, and 14 for nuScenes up to 80 meters.
- Table 15 for RobotCar up to 40 meters.
- Table 2 for RobotCar up to 50 meters.
- Table 16 for RobotCar up to 60 meters.
- Table 17 for RobotCar up to 80 meters.

The scores improve at lower distances as the problem becomes more effortless. Changes are limited due to the sparsity of the LiDAR at further distances, especially for RobotCar. While evaluating using more LiDAR frames would have allowed us to evaluate at greater distances, we avoided it as it would have introduced artifacts in the ground truth for dynamic objects. Nevertheless, our models consistently improved across the various distances and conditions tested. As described in Section 4.1, the results we reported are to be considered up to 80 meters for nuScenes, and up to 50 meters for RobotCar, unless otherwise noted.

A.4. Additional Qualitative Results

For all qualitative results reported in this work (also Figures 1, 5, and 6 in the main paper), the predictions of our

self-supervised model were all performed by our md4all-DD trained on all conditions (i.e., *dT(nr)* for nuScenes and *dT(n)* for RobotCar) and based on Monodepth2, while the supervised ones were all performed by our md4all-AD trained on *dnT(r)* and based on AdaBins. All qualitative images of the standard Monodepth2 were produced by the model trained only on *day-clear* (*day* for RobotCar). In contrast, the ones for the standard AdaBins were made by training it on all conditions, meaning that the best scoring models of each type produced them.

The following sections introduce new qualitative results on the various settings and conditions, such as nuScenes with self-supervised models (Section A.4.1), nuScenes with supervised methods (Section A.4.2), RobotCar with self-supervised approaches (Section A.4.3), failure cases (Section A.4.4), and translated images (Section A.4.5).

A.4.1 nuScenes – Self-Supervised

Night – nuScenes In Figure 8, we compare our md4all-DD to Monodepth2 [7] on particularly challenging *night* samples due to the extreme darkness levels, as well as the high amounts of reflections (also due to the wet ground, in *night-rain* conditions). Instead, in Figure 9, we compare the same models on brighter, i.e., easier samples. It can be seen that the standard Monodepth2 trained on *day-clear* had marked

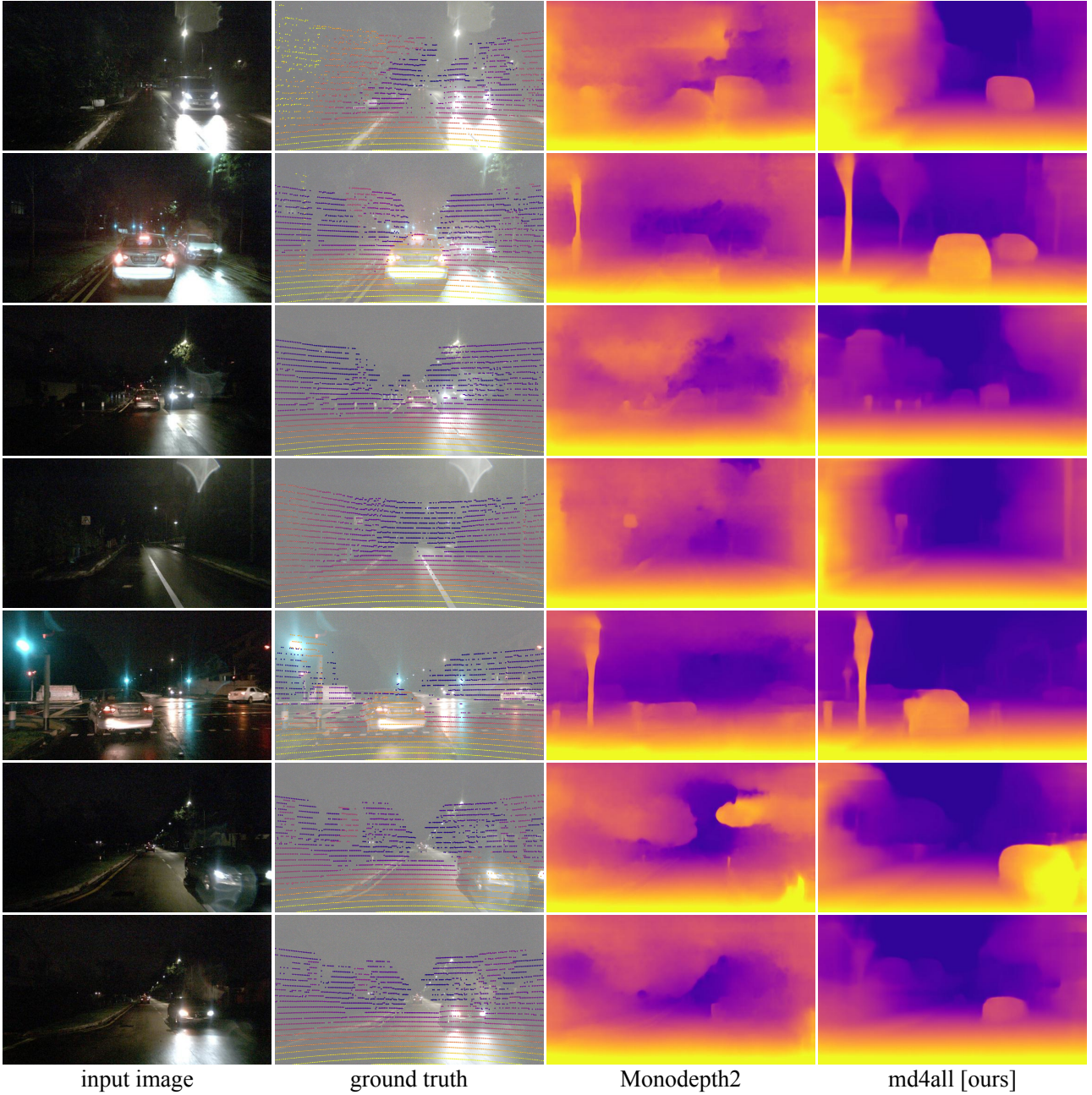


Figure 8. Comparison of self-supervised models on nuScenes [4] *night* samples. The standard Monodepth2 [7] is compared to our md4all-DD applied to Monodepth2. This set of samples is particularly challenging for the standard Monodepth2 due to the overall darkness and reflections.

difficulties with darker scenes (Figure 8), while it delivered satisfactory results in brighter settings (Figure 9). This significant difference can be attributed to the fact that Monodepth2 was trained only on *day-clear* (the model trained on *all* conditions performed significantly worse, especially at *night*, as shown in Table 1) and that the details are easier to grasp in brighter *night* inputs compared to very dark

ones. In brighter images, the road, vehicles, curb, grass, and trees on the sides can be seen relatively well thanks to the light emitted by the streetlights, allowing the standard Monodepth2 to estimate reasonable depth maps. Conversely, with darker scenes, the depth cues are highly challenging to extract due to the noise and general blackness. Nevertheless, Monodepth2 tended to estimate the ground well also

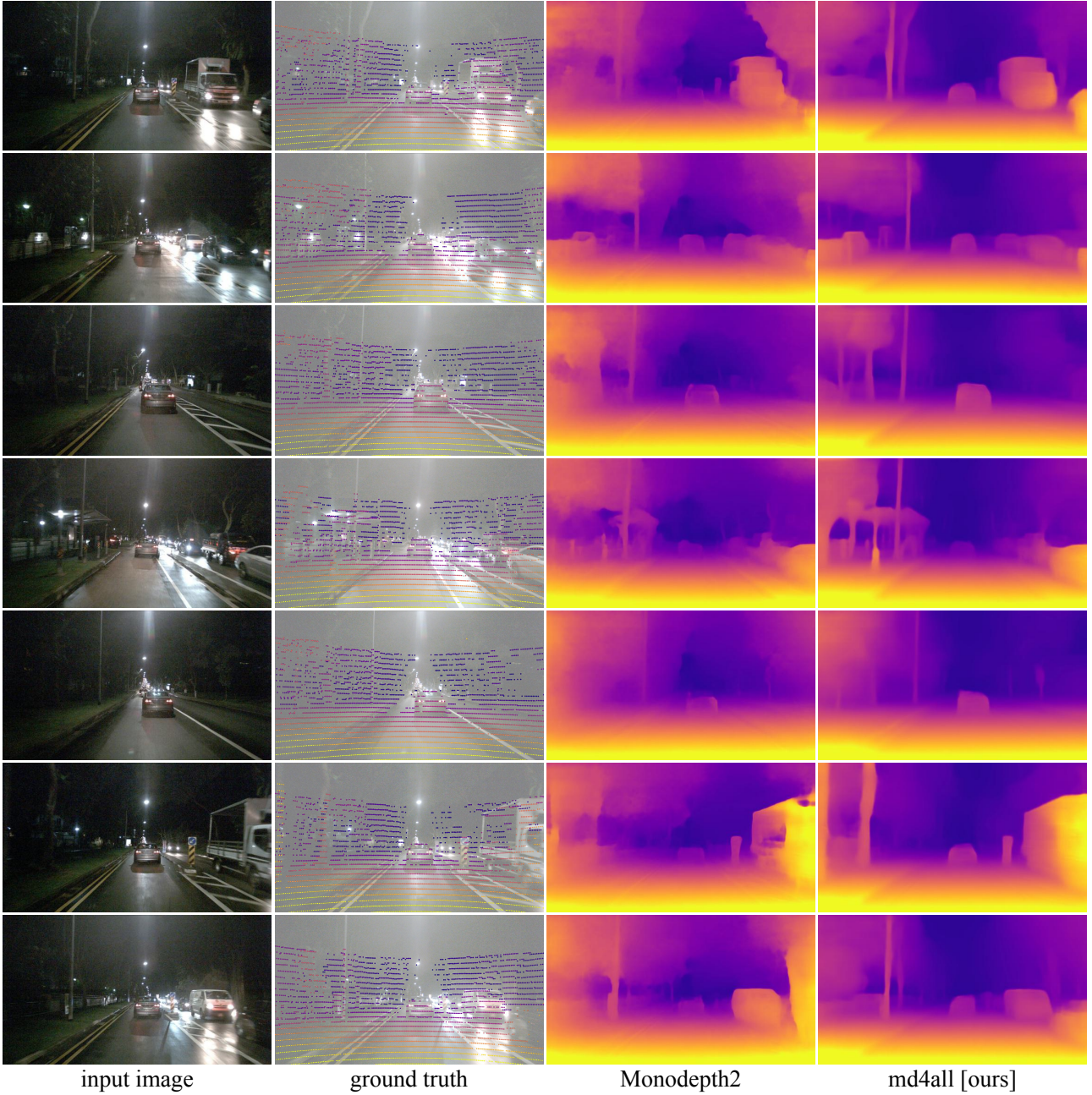


Figure 9. Comparison of self-supervised models on relatively bright nuScenes [4] *night* samples. The standard Monodepth2 [7] is compared to our md4all-DD applied to Monodepth2. This set of samples could be handled reasonably by the standard Monodepth2, thanks to the high brightness of the scenes.

in darker settings. This can be due to the model being biased toward scenes with flat and regular roads in front of the ego vehicle and the road being the most visible part of the images. Instead, Monodepth2 had severe issues with the sky, which is not evaluated since the ground truth is unavailable. Nevertheless, the proposed md4all delivered good and sharp estimates in dark and bright environments.

Rain – nuScenes In Figure 10, we compare our md4all-DD to Monodepth2 [7] on *rainy* samples of nuScenes. While the standard Monodepth2 trained on *day-clear* delivered reasonable estimates for the most part, it was severely affected by the reflections on the wet ground (e.g., first row). Also, the blurriness due to water drops affected Monodepth2, triggering overly smooth estimates. The figure

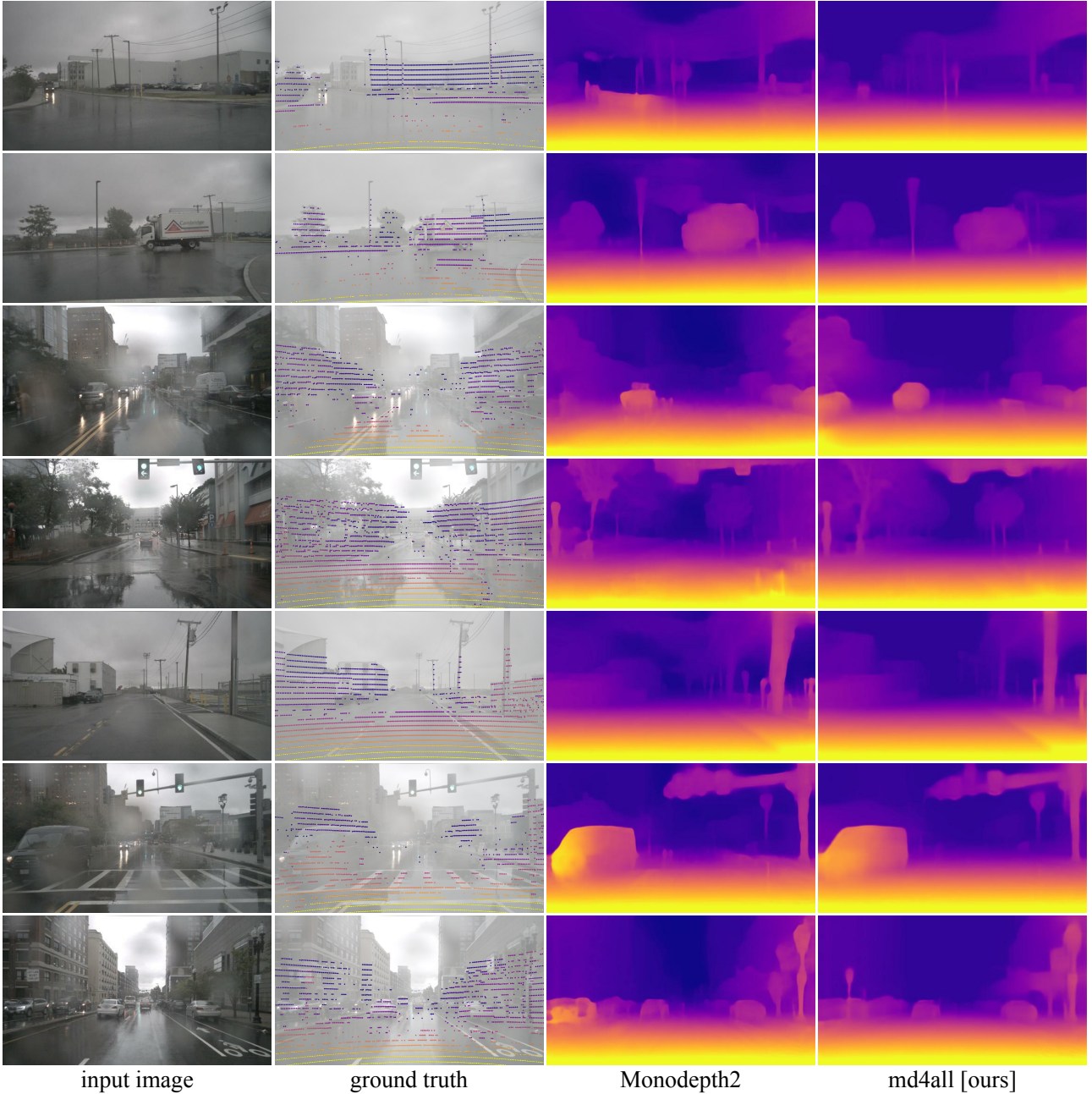


Figure 10. Comparison of self-supervised models on nuScenes [4] *rain* samples. The standard Monodepth2 [7] is compared to our md4all-DD applied to Monodepth2.

shows also Monodepth2 having issues estimating the depth of the sky (second and fifth row). Since none of these problems occurs without rain (i.e., *day-clear*, shown in Figure 11), they can be attributed to the challenging weather conditions. Instead, our md4all delivered reasonable depth estimates regardless of the adverse settings, because its supervision signal was always from ideal settings.

Day-clear – nuScenes In the lower half of Figure 11,

we compare the performance at daytime. The *day-clear* scores of Monodepth2 (d) and our md4all-DD (a) are relatively close to one another (Table 1). Based on the same model (i.e., Monodepth2), the depth maps of the two are similar, with ours delivering smoother surfaces (e.g., vehicles), with fewer artifacts around the objects, such as the black car on the left of the bottom input. Our extra smoothness is due to the day distillation scheme, propagating dense

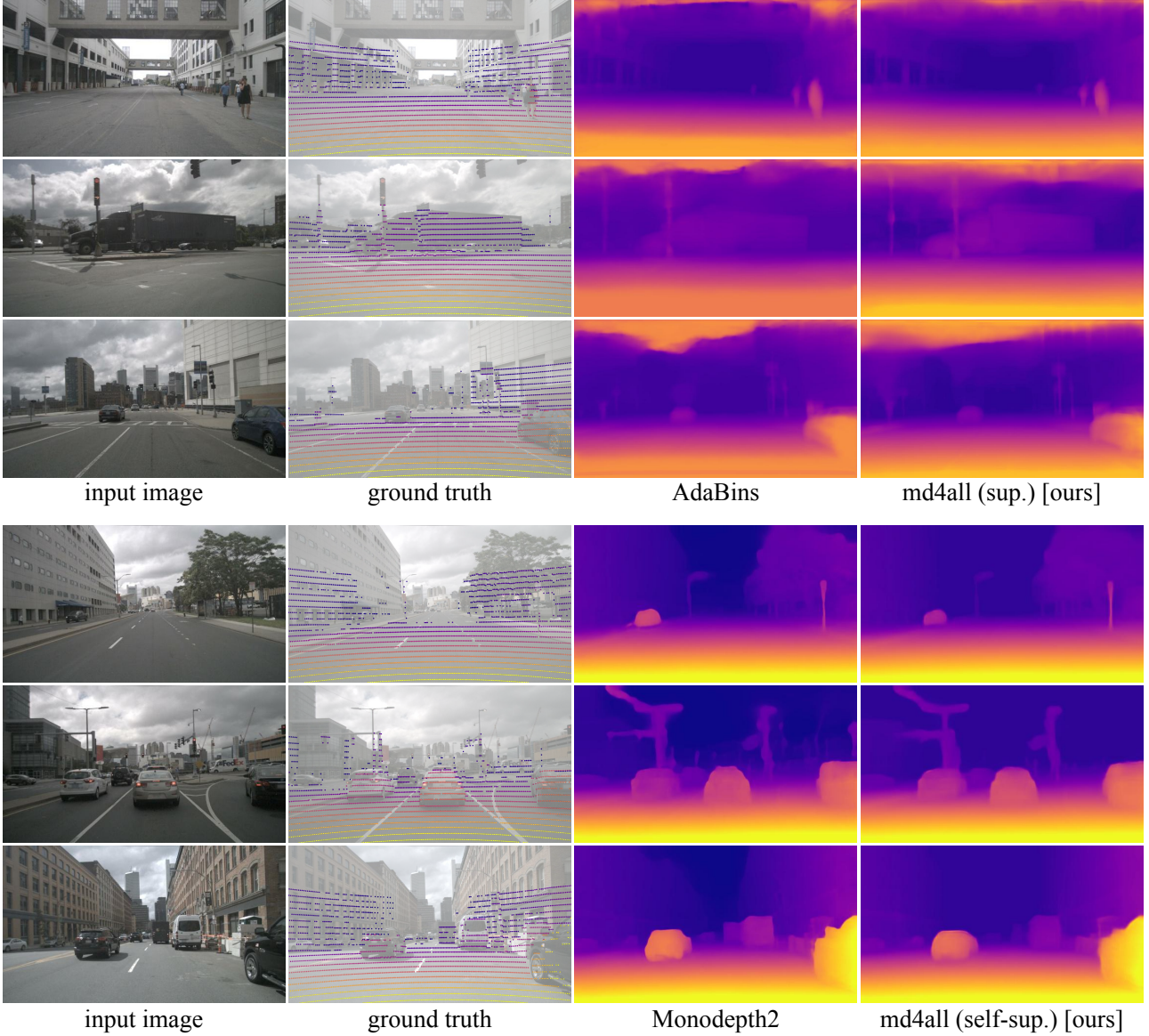


Figure 11. Comparison of models on nuScenes [4] *day-clear* samples. In the upper half, the standard AdaBins [1] is compared to our md4all-AD applied to AdaBins. In the lower half, the standard Monodepth2 [7] is compared to our md4all-DD applied to Monodepth2.

estimates from the baseline. Since our same model predicts significantly better depth maps in adverse conditions than Monodepth2, the figure confirms that our md4all does not degrade in ideal conditions. This proves the effectiveness of our techniques, as they do not introduce any performance trade-off in standard settings while enabling significantly better outcomes in challenging conditions.

A.4.2 nuScenes – Supervised

Night – nuScenes In Figure 12, we compare our md4all-AD to AdaBins [1] on particularly challenging *night* sam-

ples due to the extreme darkness levels and the high amount of reflections (same samples as in Figure 8). Our md4all-AD delivered overall sharper and more accurate estimates, as can be seen for the thin structures (e.g., poles and tree trunks), as well as the boundaries of the vehicles in the bottom two rows. Although all depth maps are displayed with the same colormap, the standard AdaBins never estimated depths that triggered the use of the bright yellow color (i.e., closest distance). This problem is less pronounced during daytime (Figure 11). This can be attributed to the standard AdaBins suffering from overfitting on the sparse ground truth data of nuScenes. Towards this end, we had to ap-

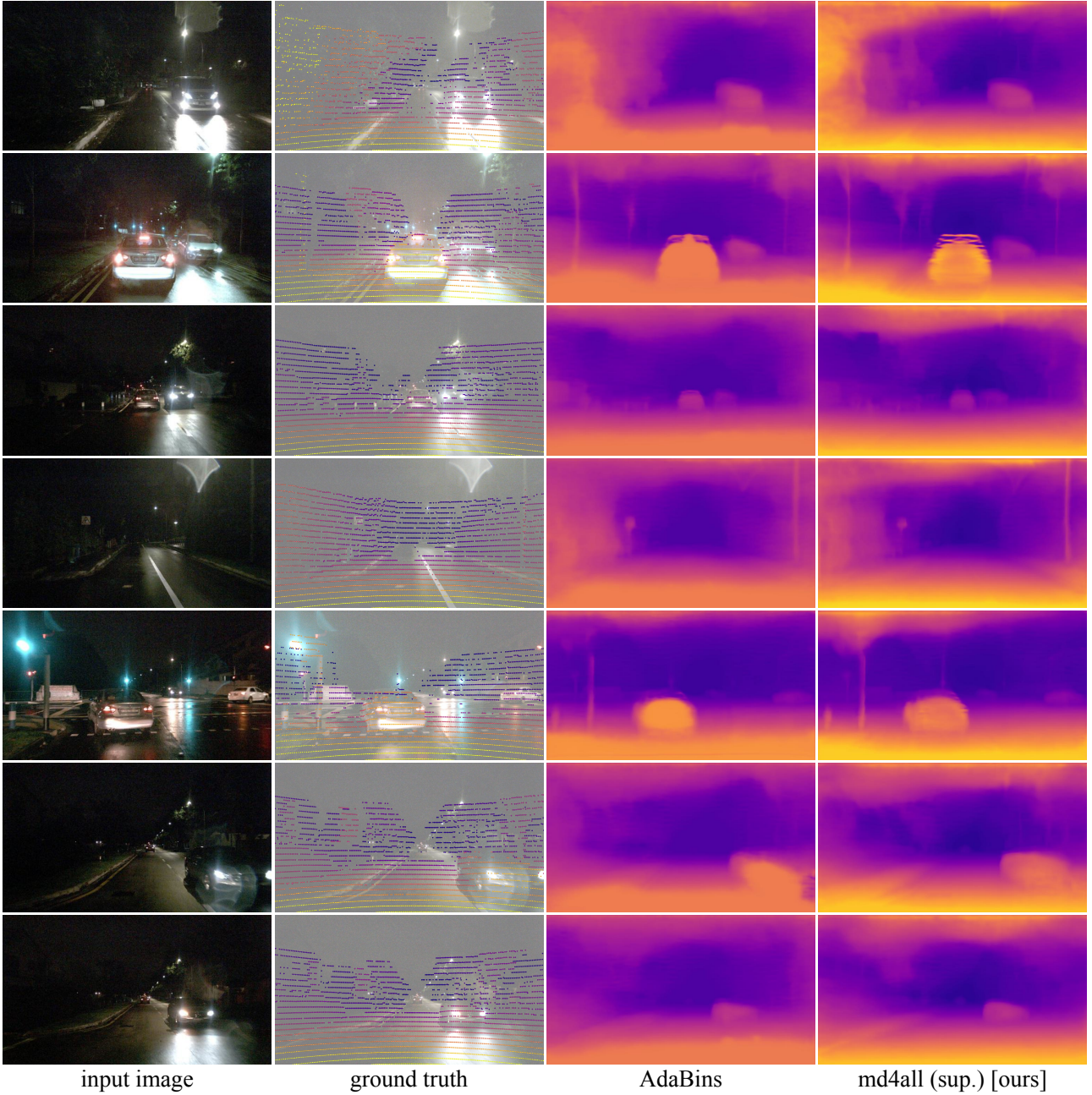


Figure 12. Comparison of supervised models on nuScenes [4] *night* samples. The standard AdaBins [1] is compared to our md4all-AD applied to AdaBins. This set of samples is particularly challenging due to the overall darkness and reflections.

ply early stopping to prevent severe horizontal artifacts from appearing on the outputs following the LiDAR detections. Applying our md4all on AdaBins introduced challenging augmentations during training, reducing overfitting and allowing it to train longer. This led to a model that estimated depth throughout a more extensive range without the horizontal artifacts. The qualitative results align with the scores gap in Table 1. Moreover, some artifacts are noticeable for

both models around the roof of the vehicles. These are due to the ground truth LiDAR data reporting further estimates in those areas (e.g., fifth row). This is caused by the relative position of the LiDAR sensor to the camera, with the former being at a higher location, thereby seeing beyond objects compared to the camera’s perspective.

Rain – nuScenes In Figure 13, we compare our md4all-AD to AdaBins [1] on *rain* samples of nuScenes. As

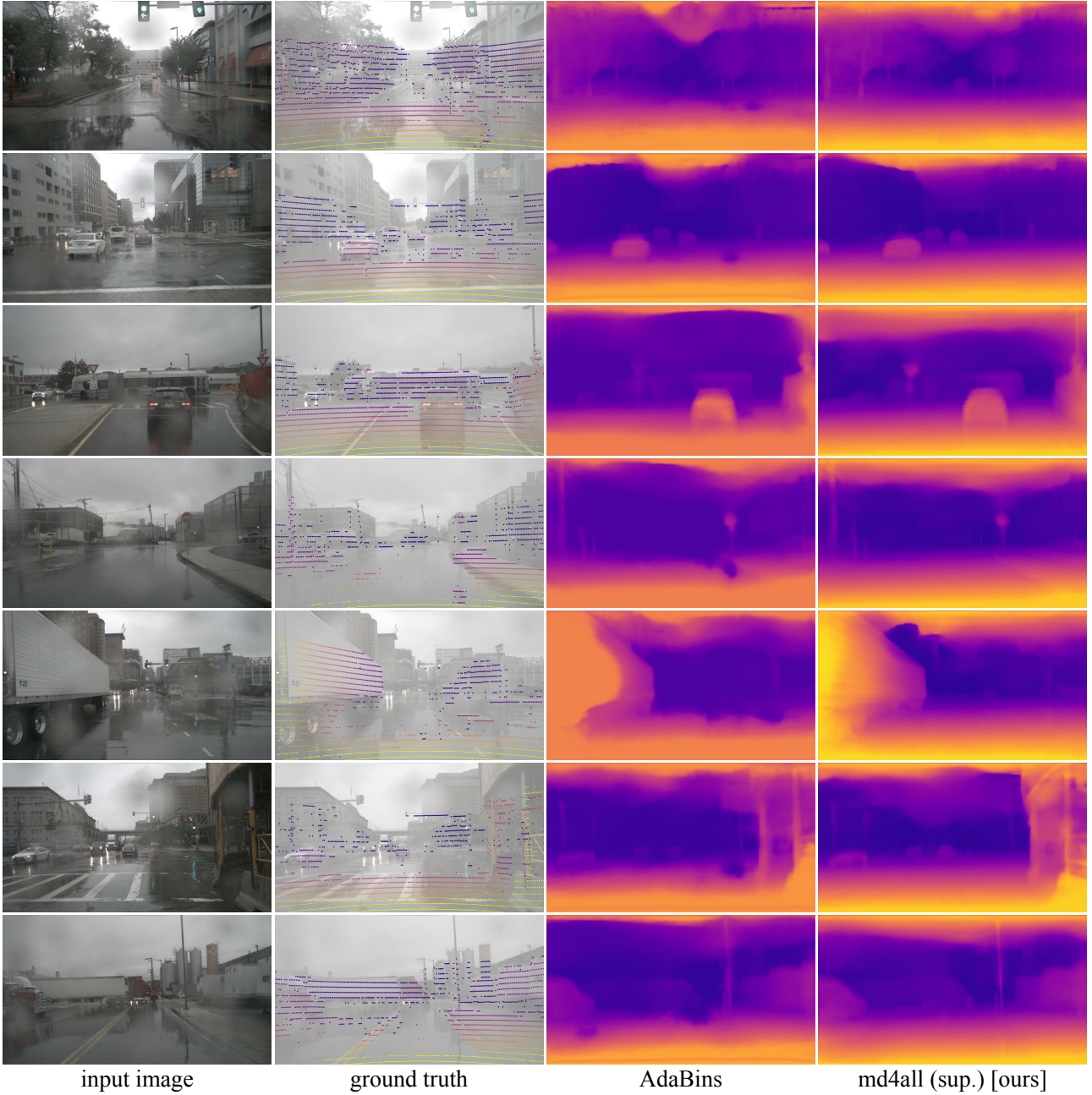


Figure 13. Comparison of supervised models on nuScenes [4] *rain* samples. The standard AdaBins [1] is compared to our md4all-AD applied to AdaBins.

the standard AdaBins learned from the erroneous measurements of the LiDAR ground truth (e.g., Figure 2), it systematically estimated artifacts on the ground, resulting in holes in its depth maps. The erroneous LiDAR measurements can also be seen from the ground truth reported for each image. Such wrong measurements are caused systematically by each highly reflective object (e.g., traffic sign) reflected on the wet ground. Therefore, training on such a

wrong signal causes the standard AdaBins to replicate the artifacts in its output (e.g., in front of the stop sign in the fourth row). Nevertheless, thanks to the reliable training signal from ideal conditions, our md4all-AD delivered good estimates without such artifacts.

Day-clear – nuScenes In the upper half of Figure 11, we compare the performance at daytime. The scores between AdaBins and our md4all-AD trained on all conditions favor

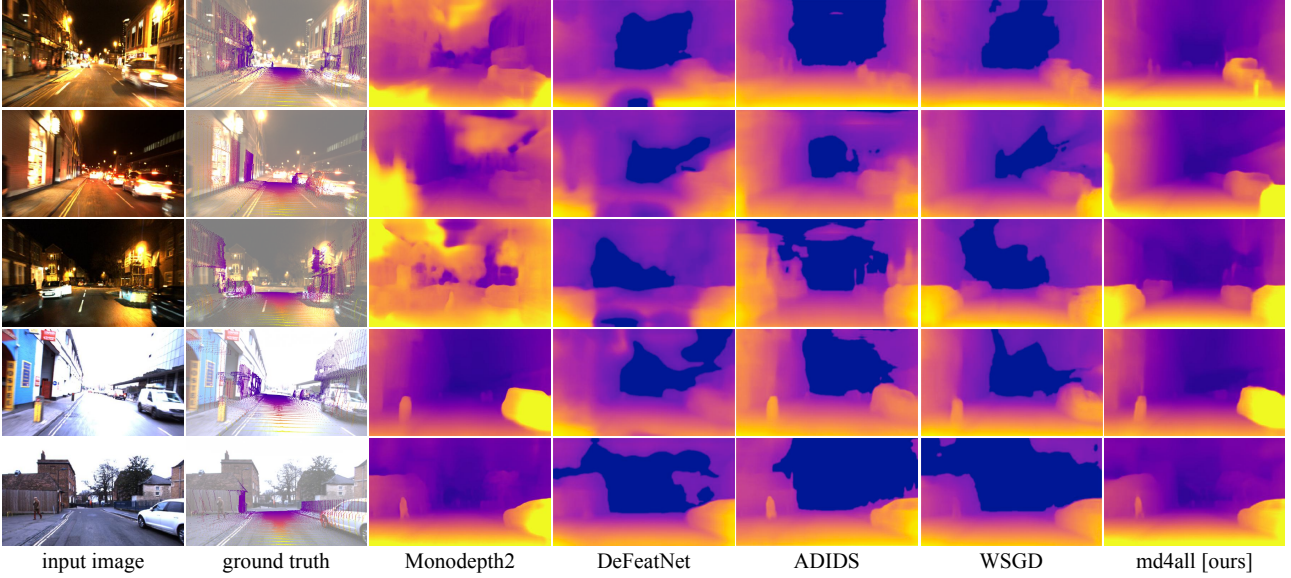


Figure 14. Comparison of self-supervised models on RobotCar [10] *night* and *day* samples. The samples are exactly the ones reported by Vankadari et al. in WSGD [12], from which we took directly the predictions of DeFeatNet [11], ADIDS [9], and WSGD [12]. We compare these with Monodepth2 [7] and our md4all-DD.

ours (Table 1). This can be seen from the depth maps, with ours delivering smoother estimates for the road and sharper details throughout. Ours separated the truck in the second frame from the background. As discussed above, this can be attributed to the strong tendency of the standard AdaBins to overfit. Instead, by feeding a mix of original and translated images, our method acts as data augmentation, mitigating the problem. Nevertheless, the limitations of such a sparse ground truth signal are evident, with smooth edges due to its sparsity and see-through effects caused by the displacement between the LiDAR and the camera. For these reasons, self-supervised outputs look better overall, as the models are also less prone to overfitting.

A.4.3 RobotCar – Self-Supervised

Comparison with other *night* methods In Figure 14, we compare our md4all-DD with various other methods targeting depth estimation in *night* conditions, namely DeFeatNet [11], ADIDS [9], and WSGD [12], as well as the standard Monodepth2 [7] designed for ideal conditions. The samples shown are exactly the ones displayed by WSGD [12] in their paper, from which we took the outputs of [11, 9, 12]. However, unlike theirs [12], for ours and Monodepth2, we do not manually threshold the maximum depth, showing the entire depth estimation, including further distances. This is possibly the reason for theirs being artificially dark in the background. Remarkably, as seen already in Table 2 and Figure 6, our md4all-DD delivered more accurate and sharper estimates in both conditions, and especially at *night*, thanks to its robust feature extraction

suitable for both *day* and *night* inputs.

Night – RobotCar Figure 15 shows additional outputs of the standard Monodepth2 [7] compared to our md4all-DD applied on Monodepth2. Although the *night* scenes are not as dark and not as noisy as those from nuScenes [4] (e.g., Figure 8), Monodepth2 had major issues estimating the depth of the image. Darkness and streetlights were detrimental for the standard Monodepth2 (e.g., in the second row). Moreover, compared to nuScenes, the images from RobotCar are often blurry at night, especially those from turns (first row). Nevertheless, the proposed md4all estimated reasonable depth maps regardless of these issues in the input. Due to the textureless pure white sky in the *day* samples of RobotCar (last two rows), the models had issues capturing its depth during training. The same occurred for the top of buildings, which are often too bright and indistinguishable from the sky in the images (e.g., fourth input in Figure 14). This problem is inherent to the data itself and causes the sky to always be predicted to be relatively close (brighter color). This did not happen for nuScenes.

Day – RobotCar Figure 15 also shows additional predictions during daytime (bottom rows). As seen in Table 2, while performing significantly better at *night*, the performance of the proposed method does not degrade during the *day* compared to the standard Monodepth2 [7]. Due to the knowledge distillation from the baseline to our md4all-DD model, surfaces result smoother, but edges remain sharp. The bottom input is particularly challenging as it features a turn. Both models correctly estimated the scene’s depth, with Monodepth2 delivering more details on the tree in the



Figure 15. Comparison of self-supervised models on RobotCar [10] *night* and *day* samples. The standard Monodepth2 [7] is compared to our md4all-DD applied to Monodepth2.

foreground but less on the tree in the background (left). As discussed for the *night* scenes, the textureless white sky of RobotCar prevents the models from learning its depth correctly, which causes erroneous estimations at test time. Instead, nuScenes [4] includes different sky conditions (e.g., sunny and cloudy), allowing the models to learn its depth.

A.4.4 Failure Cases

While our techniques bring significant improvements across various conditions, there is still room for improvement. Figure 16 reports failure cases of our models. Inherited from Monodepth2 [7], our self-supervised model has issues with

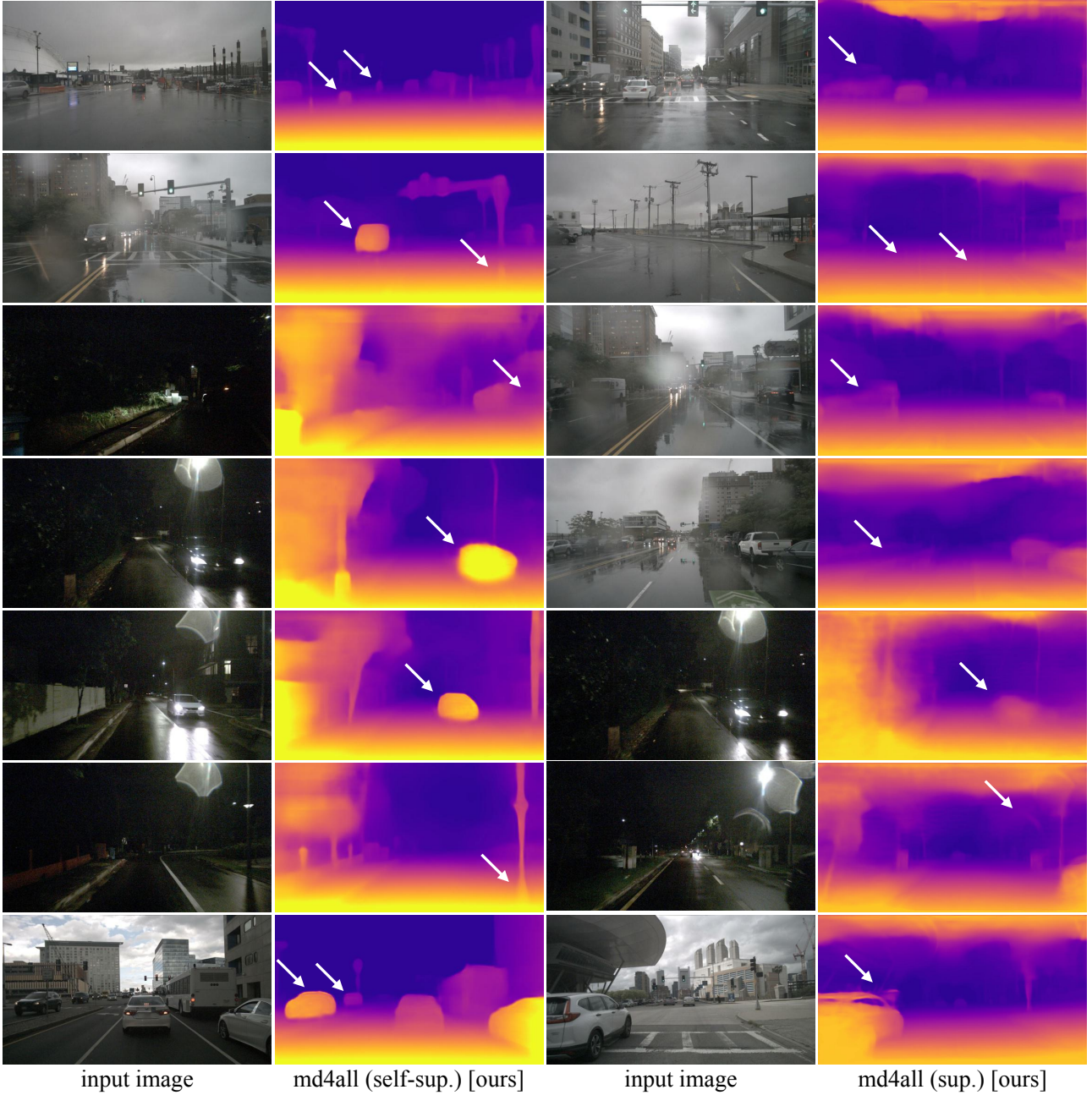


Figure 16. Failure cases of our self-supervised (md4all-DD, Monodepth2-based) and supervised (md4all-AD, AdaBins-based) models on samples from nuScenes [4]. White arrows mark issues in the predictions.

dynamic objects, especially oncoming traffic, whose distance is wrongly estimated due to the violations of the moving camera in a static world assumption [5]. This occurs in all three conditions. Furthermore, the model is occasionally fooled by reflections on the ground or misleading shadows which look like objects (first and sixth row). Additionally, highly dark scenes are also challenging (third row) due to the lack of information across nearly black pix-

els. These issues could be mitigated by integrating the data from the cost-effective radar, as in R4Dyn [5], which is robust against adverse weather and challenging illumination conditions. Our method is not bound to a specific architecture or pipeline, so it could be applied to R4Dyn directly. Another drawback our md4all inherited from Monodepth2, which we did not address, is the lack of temporal consistency in the depth predictions. Instead, for the supervised



Figure 17. Example of day-to-adverse image translations on nuScenes [4]. Training samples are translated from *day-clear* to both *rain* and *night*. For reference, in the first and last columns, we included real *rain* and *night* samples from the validation set. We share publicly the translated *night* and *rain* images for the entire *day-clear* training set.

setting, we focused on eliminating the artifacts due to the erroneous ground truth measurements. While such issues are appropriately addressed, other problems persist, such as with particular reflections and the blur caused by water drops (third row). Radar may help here too.

A.4.5 Day-to-Adverse Translation

This section shows samples of the translated images we generated with ForkGAN [16]. We publicly share all translated images corresponding to the ideal settings from the training sets of nuScenes [4] and RobotCar [10].

nuScenes In Figure 17, we show examples of the translated images used to train our models on nuScenes. The GAN [16] added plausible reflections and lights simulating wet ground or streetlights. While these additions are not al-

ways realistic, they bring challenging variations to the training set, resembling the adverse conditions in the first and last columns. The standard *day-clear* training set includes a mix of sunny and cloudy scenes. For cloudy ones, shadows are limited, with less contrast overall, making it easier for the translation model. Instead, the inputs with sunny conditions are particularly challenging for the GAN, resulting in less convincing outputs (e.g., in the last row). The GAN also learned to add water drops blurring certain areas for *rain*, and intense noise for *night*. The results of our models would directly benefit from improvements in the translations towards greater realism. With perfect translations, our md4all would further reduce the gap between ideal and challenging conditions.

RobotCar In Figure 18, we report examples of the translated images used on RobotCar. In this case, the GAN was



Figure 18. Example of day-to-adverse image translations on RobotCar [10]. Training samples are translated from *day* to *night*. For reference, we included real *night* samples from the test set in the central column. We share publicly the translated *night* images for the entire *day* training set.

trained entirely on RobotCar, thanks to the high amount of *night* samples available. Since the weather in the *day* samples tends to be always cloudy, with a consistently white sky, the GAN did not exhibit the issues seen for nuScenes with sunny inputs (Figure 17). Instead, the GAN delivered highly plausible samples with a higher degree of realism than for nuScenes, enabling our model to narrow the margin between daytime and nighttime performances. As seen for nuScenes, the GAN added streetlights resembling the ones seen throughout the dataset. Furthermore, it can be seen clearly that the GAN learned the headlights of the ego vehicle, which it added to illuminate the road ahead.

For both datasets, while being somewhat plausible, the translated samples are not perfect. For nuScenes, for example, noise patterns in the *night* translations are repeated similarly over the lower left corner of the images. The noise is more unstructured for the real *night* images. Additionally, the GAN did not learn to turn on the lights of other cars, which are often a source of issues for the models.

A.5. Attempted Approaches That Did Not Work

We explored several alternative solutions for this challenging problem, and we intend to mention them in this section to help future researchers who target the same issues. As written in Section 4.1, we experimented with diffusion models to learn our day-to-adverse image translation. However, the lack of paired images for *day* and *night* made it unfeasible. Therefore, we opted for GANs, which do not need

paired inputs. Furthermore, we experimented with the following methods on the *night* samples of nuScenes, seeking improvements over the standard Monodepth2. Spatial and temporal attention could have allowed the model to focus on valuable information. Still, it did not bring an improvement, possibly due to the large amount of noise varying across different *night* images, which could have prevented to attend on the helpful information. Additionally, we experimented with incorporating geometrical priors (e.g., from edge detection) into the losses, but they were similarly not beneficial. The simple solution presented in this work was the most effective to tackle this complex problem.

References

- [1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 3, 6, 8, 9, 10, 15, 16, 17
- [2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 5, 8
- [3] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, Angela P Schoellig, and Timothy D Barfoot. Boreas: A multi-season au-

- tonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2):33–42, 2023. [8](#)
- [4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#)
- [5] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021. [4](#), [5](#), [6](#), [10](#), [20](#)
- [6] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. [2](#)
- [7] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [18](#), [19](#)
- [8] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. [2](#), [5](#), [6](#), [7](#), [8](#)
- [9] Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised monocular depth estimation for all day images using domain separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12737–12746, 2021. [5](#), [18](#)
- [10] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. [3](#), [4](#), [5](#), [7](#), [10](#), [11](#), [18](#), [19](#), [21](#), [22](#)
- [11] Jaime Spencer, Richard Bowden, and Simon Hadfield. DeFeat-Net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020. [5](#), [18](#)
- [12] Madhu Vankadari, Stuart Golodetz, Sourav Garg, Sangyun Shin, Andrew Markham, and Niki Trigoni. When the Sun Goes Down: Repairing photometric losses for all-day depth estimation. In *Proceedings of the Conference on Robot Learning*, pages 1992–2003. PMLR, 2023. [4](#), [5](#), [7](#), [18](#)
- [13] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing Nighttime Weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16055–16064, 2021. [4](#), [5](#), [6](#)
- [14] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. [2](#)
- [15] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. [4](#)
- [16] Ziqiang Zheng, Yang Wu, Xinran Han, and Jianbo Shi. ForkGAN: Seeing into the rainy night. In *Proceedings of the European Conference on Computer Vision*, pages 155–170. Springer, 2020. [3](#), [8](#), [21](#)

Licenses of the Publications

The following are the licenses from the publisher (i.e., IEEE) of each publication included in this dissertation [82, 83, 84, 85, 86, 162].



Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds

Author: Stefano Gasperini

Publication: IEEE Robotics and Automation Letters

Publisher: IEEE

Date: April 2021

Copyright © 2021, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW



CertainNet: Sampling-Free Uncertainty Estimation for Object Detection

Author: Stefano Gasperini

Publication: IEEE Robotics and Automation Letters

Publisher: IEEE

Date: April 2022

Copyright © 2022, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW



3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection



Conference Proceedings:

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Author:

Alexander Lehner; Stefano Gasperini; Alvaro Marcos-Ramiro; Michael Schmidt; Mohammad-Ali Nikouei Mahani; Nassir Navab; Benjamin Busam; Federico Tombari

Publisher: IEEE

Date: 18-24 June 2022

Copyright © 2022, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW



RightsLink

[Sign in/Register](#)

Segmenting Known Objects and Unseen Unknowns without Prior Knowledge

Conference Proceedings: 2023 IEEE/CVF International Conference on Computer Vision (ICCV)

Author: Stefano Gasperini

Publisher: IEEE

Date: 01 October 2023

Copyright © 2023, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[BACK](#)[CLOSE WINDOW](#)



R4Dyn: Exploring Radar for Self-Supervised Monocular Depth Estimation of Dynamic Scenes

Conference Proceedings: 2021 International Conference on 3D Vision (3DV)

Author:

Stefano Gasperini; Patrick Koch; Vinzenz Dallabetta; Nassir Navab; Benjamin Busam; Federico Tombari

Publisher: IEEE

Date: 1-3 Dec. 2021

Copyright © 2021, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW



RightsLink

[Sign in/Register](#)

Robust Monocular Depth Estimation under Challenging Conditions

Conference Proceedings: 2023 IEEE/CVF International Conference on Computer Vision (ICCV)

Author: Stefano Gasperini

Publisher: IEEE

Date: 01 October 2023

Copyright © 2023, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[BACK](#)[CLOSE WINDOW](#)

Bibliography

- [1] H. Abualsaud, S. Liu, D. B. Lu, K. Situ, A. Rangesh, and M. M. Trivedi. “LaneAF: Robust multi-lane detection with affinity fields”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7477–7484 (see p. 24).
- [2] M. Alawadhi, J. Almazrouie, M. Kamil, and K. A. Khalil. “A systematic literature review of the factors influencing the adoption of autonomous driving”. In: *International Journal of System Assurance Engineering and Management* 11 (2020), pp. 1065–1082 (see pp. 11, 12, 14).
- [3] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. “Safe reinforcement learning via shielding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018 (see p. 30).
- [4] M. Althoff and J. M. Dolan. “Online verification of automated road vehicles using reachability analysis”. In: *IEEE Transactions on Robotics* 30.4 (2014), pp. 903–918 (see pp. 12, 30).
- [5] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus. “Learning robust control policies for end-to-end autonomous driving from data-driven simulation”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1143–1150 (see p. 139).
- [6] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola. *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014 (see pp. 10, 15).
- [7] A. Antoniou, A. Storkey, and H. Edwards. “Data augmentation Generative Adversarial Networks”. In: *arXiv preprint arXiv:1711.04340* (2017) (see pp. 41, 69, 137, 140).
- [8] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019) (see p. 40).
- [9] A. Atapour-Abarghouei and T. P. Breckon. “Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer”. In: *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition*. 2018, pp. 2800–2810 (see p. 141).
- [10] E. Awad, S. Dsouza, R. Kim, J. Schulz, J. Henrich, A. Shariff, J.-F. Bonnefon, and I. Rahwan. “The moral machine experiment”. In: *Nature* 563.7729 (2018), pp. 59–64 (see pp. 14, 31).
- [11] K. Backman, D. Kulić, and H. Chung. “Learning to assist drone landings”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3192–3199 (see p. 21).
- [12] A. Badki, O. Gallo, J. Kautz, and P. Sen. “Binary TTC: A temporal geofence for autonomous navigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12946–12955 (see p. 12).
- [13] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner. “The Oxford Radar RobotCar Dataset: A radar extension to the Oxford RobotCar Dataset”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2020, pp. 6433–6438 (see p. 23).
- [14] L. Bastian, T. Czempel, C. Heiliger, K. Karcz, U. Eck, B. Busam, and N. Navab. “Know your sensORs – A modality study for surgical action classification”. In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 11.4 (2023), pp. 1113–1121 (see p. 21).

- [15] L. Bastian, D. Derkacz-Bogner, T. D. Wang, B. Busam, and N. Navab. “SegmentOR: Obtaining Efficient Operating Room Semantics Through Temporal Propagation”. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2023, pp. 57–67 (see pp. 20, 21, 25).
- [16] L. Bastian, T. D. Wang, T. Czempel, B. Busam, and N. Navab. “DisguisOR: Holistic face anonymization for the operating room”. In: *International Journal of Computer Assisted Radiology and Surgery* (2023), pp. 1–7 (see p. 21).
- [17] L. Bastian, Y. Xie, N. Navab, and Z. Löhner. “Hybrid functional maps for crease-aware non-isometric shape matching”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 3313–3323 (see p. 21).
- [18] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9297–9307 (see pp. 25, 26, 44, 139).
- [19] J. Behley, A. Milioto, and C. Stachniss. “A benchmark for LiDAR-based panoptic segmentation based on KITTI”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2021, pp. 13596–13603 (see pp. 44, 46).
- [20] C. I. Bercea, M. Neumayr, D. Rueckert, and J. A. Schnabel. “Mask, Stitch, and Re-Sample: Enhancing Robustness and Generalizability in Anomaly Detection through Automatic Diffusion Models”. In: *arXiv preprint arXiv:2305.19643* (2023) (see p. 38).
- [21] C. I. Bercea, B. Wiestler, D. Rueckert, and J. A. Schnabel. “Generalizing Unsupervised Anomaly Detection: Towards Unbiased Pathology Screening”. In: *Medical Imaging with Deep Learning*. 2023 (see p. 38).
- [22] L. Berscheid, P. Meißner, and T. Kröger. “Self-supervised learning for precise pick-and-place without object model”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4828–4835 (see p. 21).
- [23] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, et al. “Improving image generation with better captions”. In: *OpenAI Papers* (2023) (see pp. 5, 10).
- [24] S. F. Bhat, I. Alhashim, and P. Wonka. “AdaBins: Depth estimation using adaptive bins”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4009–4018 (see p. 26).
- [25] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. “Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets”. In: *arXiv preprint arXiv:2311.15127* (2023) (see p. 141).
- [26] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. “End to end learning for self-driving cars”. In: *arXiv preprint arXiv:1604.07316* (2016) (see p. 16).
- [27] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021) (see p. 137).
- [28] J.-F. Bonnefon, A. Shariff, and I. Rahwan. “The social dilemma of autonomous vehicles”. In: *Science* 352.6293 (2016), pp. 1573–1576 (see p. 16).
- [29] F. Bork, A. Lehner, U. Eck, N. Navab, J. Waschke, and D. Kugelmann. “The effectiveness of collaborative augmented reality in gross anatomy teaching: a quantitative and qualitative pilot study”. In: *Anatomical Sciences Education* 14.5 (2021), pp. 590–604 (see p. 22).
- [30] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari. “Semantic monocular SLAM for highly dynamic environments”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018, pp. 393–400 (see p. 24).

- [31] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. “Language models are few-shot learners”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901 (see pp. 15, 29, 33).
- [32] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez. “Zero-shot semantic segmentation”. In: *Advances in Neural Information Processing Systems* 32 (2019) (see pp. 37, 83).
- [33] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA Urban Challenge: Autonomous vehicles in city traffic*. Vol. 56. Springer, 2009 (see p. 11).
- [34] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung, et al. “Boreas: A multi-season autonomous driving dataset”. In: *The International Journal of Robotics Research* 42.1-2 (2023), pp. 33–42 (see p. 35).
- [35] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. “nuScenes: A multimodal dataset for autonomous driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631 (see pp. 6, 20, 22, 23, 26, 31, 35, 101).
- [36] A. Caliskan, J. J. Bryson, and A. Narayanan. “Semantics derived automatically from language corpora contain human-like biases”. In: *Science* 356.6334 (2017), pp. 183–186 (see p. 31).
- [37] M. Cannici, C. Plizzari, M. Planamente, M. Ciccone, A. Bottino, B. Caputo, and M. Matteucci. “N-ROD: A neuromorphic dataset for synthetic-to-real domain adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2021, pp. 1342–1347 (see p. 33).
- [38] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li. “Invisible for both camera and LiDAR: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks”. In: *Proceedings of the IEEE Symposium on Security and Privacy*. 2021, pp. 176–194 (see p. 41).
- [39] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660 (see pp. 33, 137, 138, 141).
- [40] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos. “A real-time approach for chance-constrained motion planning with dynamic obstacles”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3620–3625 (see p. 21).
- [41] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu. “Deep metric learning for open world semantic segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15333–15342 (see p. 37).
- [42] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. “An empirical study and analysis of generalized zero-shot learning for object recognition in the wild”. In: *Proceedings of the European conference on computer vision*. Springer, 2016, pp. 52–68 (see p. 37).
- [43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A simple framework for contrastive learning of visual representations”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1597–1607 (see p. 136).
- [44] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun. “GeoSim: Realistic video simulation via geometry-aware composition for self-driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7230–7240 (see p. 140).
- [45] Z. Chen, J. Yang, J. Huang, R. de Lutio, J. M. Esturo, B. Ivanovic, O. Litany, Z. Gojcic, S. Fidler, M. Pavone, et al. “OmniRe: Omni urban scene reconstruction”. In: *arXiv preprint arXiv:2408.16760* (2024) (see p. 140).
- [46] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. “Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12475–12485 (see pp. 26, 36, 44, 45).

- [47] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger. “TransFuser: Imitation with transformer-based sensor fusion for autonomous driving”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2022) (see p. 23).
- [48] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon. “ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14367–14376 (see p. 141).
- [49] L. Collingwood. “Privacy implications and liability issues of autonomous vehicles”. In: *Information & Communications Technology Law* 26.1 (2017), pp. 32–45 (see p. 14).
- [50] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3213–3223 (see pp. 17, 20, 84).
- [51] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. “Joint distribution optimal transportation for domain adaptation”. In: *Advances in Neural Information Processing Systems* 30 (2017) (see p. 39).
- [52] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric. “Multimodal trajectory predictions for autonomous driving using deep convolutional networks”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2019, pp. 2090–2096 (see p. 135).
- [53] R. Cyganski. “Automated Vehicles and Automated Driving from a Demand Modeling Perspective”. In: *Autonomous driving: Technical, legal and social aspects* (2016), pp. 233–253 (see p. 11).
- [54] T. Czempel, M. Paschali, M. Keicher, W. Simson, H. Feussner, S. T. Kim, and N. Navab. “TeCNO: Surgical phase recognition with multi-stage temporal convolutional networks”. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer. 2020, pp. 343–352 (see p. 21).
- [55] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. “Scaling egocentric vision: The Epic-Kitchens dataset”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 720–736 (see p. 22).
- [56] R. De Charette and F. Nashashibi. “Traffic light recognition using image processing compared to learning processes”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 333–338 (see p. 24).
- [57] A. De la Escalera, J. M. Armingol, and M. Mata. “Traffic sign recognition and analysis for intelligent vehicles”. In: *Image and Vision Computing* 21.3 (2003), pp. 247–258 (see p. 24).
- [58] V. R. Desaraju and J. P. How. “Decentralized path planning for multi-agent teams with complex constraints”. In: *Autonomous Robots* 32 (2012), pp. 385–403 (see p. 17).
- [59] A. Devo, J. Mao, G. Costante, and G. Loianno. “Autonomous single-image drone exploration with deep reinforcement learning and mixed reality”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5031–5038 (see p. 21).
- [60] H. Dhama, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht. “Semantic image manipulation using scene graphs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision And Pattern Recognition*. 2020, pp. 5213–5222 (see p. 141).
- [61] H. Dhama, F. Manhardt, N. Navab, and F. Tombari. “Graph-to-3D: End-to-end generation and manipulation of 3D scenes using scene graphs”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16352–16361 (see p. 140).
- [62] F. Di Felice, A. Remus, S. Gasperini, B. Busam, L. Ott, F. Tombari, R. Siegwart, and C. A. Avizzano. “Zero123-6D: Zero-shot novel view synthesis for RGB category-level 6D pose estimation”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2024, pp. 14204–14211 (see p. 21).

- [63] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “CARLA: An open urban driving simulator”. In: *Conference on Robot Learning*. PMLR. 2017, pp. 1–16 (see pp. 33, 141).
- [64] U. DoT. “Preparing for the future of transportation: Automated vehicles 3.0”. In: *U.S. Department of Transportation* (2018) (see p. 15).
- [65] K. Drukker, W. Chen, J. Gichoya, N. Grusauskas, J. Kalpathy-Cramer, S. Koyejo, K. Myers, R. C. Sá, B. Sahiner, H. Whitney, et al. “Toward fairness in artificial intelligence for medical image analysis: Identification and mitigation of potential biases in the roadmap from data collection to model deployment”. In: *Journal of Medical Imaging* 10.6 (2023), pp. 061104–1–061104–26 (see p. 34).
- [66] V. Ehm, N. E. Amrani, Y. Xie, L. Bastian, M. Gao, W. Wang, L. Sang, D. Cao, Z. Löhner, D. Cremers, et al. “Beyond Complete Shapes: A quantitative Evaluation of 3D Shape Matching Algorithms”. In: *arXiv preprint arXiv:2411.03511* (2024) (see p. 21).
- [67] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. “Robust physical-world attacks on deep learning visual classification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1625–1634 (see pp. 40, 41).
- [68] D. J. Fagnant and K. Kockelman. “Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations”. In: *Transportation Research Part A: Policy and Practice* 77 (2015), pp. 167–181 (see pp. 10, 15).
- [69] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. “Predictive active steering control for autonomous vehicle systems”. In: *Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580 (see p. 17).
- [70] R. Fan, H. Wang, P. Cai, and M. Liu. “SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 340–356 (see p. 24).
- [71] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360 (see p. 17).
- [72] C. Finn, P. Abbeel, and S. Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135 (see p. 40).
- [73] R. Fong, M. Patrick, and A. Vedaldi. “Understanding deep networks via extremal perturbations and smooth masks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2950–2958 (see p. 138).
- [74] E. Fraedrich and B. Lenz. “Societal and individual acceptance of autonomous driving”. In: *Autonomous driving: Technical, legal and social aspects* (2016), pp. 621–640 (see p. 14).
- [75] U. Franke, D. Pfeiffer, C. Rabe, C. Knoepfel, M. Enzweiler, F. Stein, and R. Herrtwich. “Making Bertha see”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2013, pp. 214–221 (see p. 12).
- [76] G. French, M. Mackiewicz, and M. Fisher. “Self-ensembling for visual domain adaptation”. In: *International Conference on Learning Representations*. 2018 (see p. 39).
- [77] C. B. Frey and M. A. Osborne. “The future of employment: How susceptible are jobs to computerisation?”. In: *Technological Forecasting and Social Change* 114 (2017), pp. 254–280 (see p. 14).
- [78] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1050–1059 (see p. 38).
- [79] Y. Gal, R. Islam, and Z. Ghahramani. “Deep Bayesian active learning with image data”. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2017, pp. 1183–1192 (see pp. 26, 33).

- [80] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030 (see pp. 32, 39, 41).
- [81] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler. “GET3D: A generative model of high quality 3D textured shapes learned from images”. In: *Advances In Neural Information Processing Systems* 35 (2022), pp. 31841–31854 (see p. 141).
- [82] S. Gasperini, J. Haug, M.-A. N. Mahani, A. Marcos-Ramiro, N. Navab, B. Busam, and F. Tombari. “CertainNet: Sampling-Free Uncertainty Estimation for Object Detection”. In: *IEEE Robotics and Automation Letters* 7.2 (2021), pp. 698–705 (see pp. 1, 23, 26, 35, 36, 38, 57, 69, 71, 83, 87, 136, 138, 139, 211).
- [83] S. Gasperini, P. Koch, V. Dallabetta, N. Navab, B. Busam, and F. Tombari. “R4Dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2021, pp. 751–760 (see pp. 2, 6, 22, 23, 26, 30, 36, 102, 115, 116, 119, 137, 139, 177, 211).
- [84] S. Gasperini, M.-A. N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. “Panoster: End-to-end panoptic segmentation of LiDAR point clouds”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3216–3223 (see pp. 1, 6, 23, 24, 30, 32, 45, 87, 135, 139, 211).
- [85] S. Gasperini, A. Marcos-Ramiro, M. Schmidt, N. Navab, B. Busam, and F. Tombari. “Segmenting Known Objects and Unseen Unknowns without Prior Knowledge”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19321–19332 (see pp. 1, 6, 23, 25, 33–35, 37, 38, 136–139, 161, 211).
- [86] S. Gasperini, N. Morbitzer, H. Jung, N. Navab, and F. Tombari. “Robust Monocular Depth Estimation under Challenging Conditions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 8177–8186 (see pp. 1, 6, 20, 23, 25, 26, 35, 36, 115, 116, 137, 139, 185, 211).
- [87] S. Gasperini, M. Paschali, C. Hopke, D. Wittmann, and N. Navab. “Signal clustering with class-independent segmentation”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 3982–3986 (see pp. 45–47, 135).
- [88] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. “A survey of uncertainty in deep neural networks”. In: *Artificial Intelligence Review* 56.Suppl 1 (2023), pp. 1513–1589 (see p. 38).
- [89] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets robotics: The KITTI dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237 (see pp. 33–35, 101, 139).
- [90] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for autonomous driving? the KITTI vision benchmark suite”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361 (see pp. 6, 12, 17, 21, 22, 25, 26, 29, 31, 33–35, 139).
- [91] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *International Conference on Learning Representations*. 2018 (see p. 33).
- [92] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. “Scaling open-vocabulary image segmentation with image-level labels”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 540–557 (see pp. 137, 138).
- [93] D. J. Glancy. “Autonomous and automated and connected cars-Oh my! First generation autonomous cars in the legal ecosystem”. In: *Minnesota Journal of Law Science & Technology* 16 (2015), p. 619 (see p. 16).
- [94] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. “Digging into self-supervised monocular depth estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3828–3838 (see pp. 24, 26, 29, 101, 115, 117).
- [95] K. Goel, P. Srinivasan, S. Tariq, and J. Philbin. “QuadroNet: Multi-task learning for real-time semantic depth aware instance segmentation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 315–324 (see p. 12).

- [96] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi. “Bayes’ Rays: Uncertainty quantification for neural radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20061–20070 (see p. 140).
- [97] L. Goli, D. Rebain, S. Sabour, A. Garg, and A. Tagliasacchi. “nerf2nerf: Pairwise registration of neural radiance fields”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2023, pp. 9354–9361 (see p. 140).
- [98] N. J. Goodall. “Machine ethics and automated vehicles”. In: *Road Vehicle Automation* (2014), pp. 93–102 (see p. 14).
- [99] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems* 27 (2014) (see pp. 39, 41).
- [100] I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *International Conference on Learning Representations*. 2015 (see pp. 30, 32, 40).
- [101] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. “3D packing for self-supervised monocular depth estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2485–2494 (see pp. 12, 102).
- [102] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. “Semantically-Guided Representation Learning for Self-Supervised Monocular Depth”. In: *International Conference on Learning Representations*. 2020 (see p. 101).
- [103] I. Gulrajani and D. Lopez-Paz. “In Search of Lost Domain Generalization”. In: *International Conference on Learning Representations*. 2020 (see p. 39).
- [104] D. Guo and K. K. Leang. “Image-based estimation, planning, and control of a cable-suspended payload for package delivery”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2698–2705 (see p. 21).
- [105] L. Gustafson, C. Rolland, N. Ravi, Q. Duval, A. Adcock, C.-Y. Fu, M. Hall, and C. Ross. “FACET: Fairness in Computer Vision Evaluation Benchmark”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 20370–20382 (see pp. 34, 35).
- [106] R. C. Hampshire and D. Shoup. “What share of traffic is cruising for parking?” In: *Journal of Transport Economics and Policy* 52.3 (2018), pp. 184–201 (see p. 10).
- [107] A. Harakeh, M. Smart, and S. L. Waslander. “BayesOD: A Bayesian approach for uncertainty estimation in deep object detectors”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2020, pp. 87–93 (see p. 58).
- [108] K. He, G. Gkioxari, P. Dollár, and R. Girshick. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969 (see p. 44).
- [109] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778 (see p. 29).
- [110] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. “Natural adversarial examples”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15262–15271 (see p. 32).
- [111] M. Herb, M. Lemberger, M. M. Schmitt, A. Kurz, T. Weiherer, N. Navab, and F. Tombari. “Semantic image alignment for vehicle localization”. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. 2021, pp. 1124–1131 (see pp. 17, 24).
- [112] M. Herb, T. Weiherer, N. Navab, and F. Tombari. “Crowd-sourced semantic edge mapping for autonomous vehicles”. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. 2019, pp. 7047–7053 (see pp. 17, 24).
- [113] A. Hermans, G. Floros, and B. Leibe. “Dense 3D semantic mapping of indoor scenes from RGB-D images”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2014, pp. 2631–2638 (see p. 21).

- [114] A. Hevelke and J. Nida-Rümelin. “Responsibility for crashes of autonomous vehicles: An ethical analysis”. In: *Science and Engineering Ethics* 21 (2015), pp. 619–630 (see p. 14).
- [115] D. Hidalgo-Carvajal, H. Chen, G. C. Bettelani, J. Jung, M. Zavaglia, L. Busse, A. Naceri, S. Leutenegger, and S. Haddadin. “Anthropomorphic Grasping with Neural Object Shape Completion”. In: *IEEE Robotics and Automation Letters* (2023) (see pp. 19, 21).
- [116] F. Holm, G. Ghazaei, T. Czempel, E. Özsoy, S. Saur, and N. Navab. “Dynamic Scene Graph Representation for Surgical Video”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2023, pp. 81–87 (see p. 21).
- [117] J. Hornauer and V. Belagiannis. “Gradient-based uncertainty for monocular depth estimation”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 613–630 (see pp. 38, 138).
- [118] J. Hornauer and V. Belagiannis. “Heatmap-based Out-of-Distribution Detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 2603–2612 (see p. 37).
- [119] J. Hornauer, A. Holzbock, and V. Belagiannis. “Out-of-Distribution Detection for Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 1911–1921 (see pp. 38, 135).
- [120] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska. “One thousand and one hours: Self-driving motion prediction dataset”. In: *Conference on Robot Learning*. PMLR. 2021, pp. 409–418 (see p. 12).
- [121] L. Hoyer, D. Dai, and L. Van Gool. “DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9924–9935 (see pp. 25, 33, 39).
- [122] L. Hoyer, D. Dai, and L. Van Gool. “Domain Adaptive and Generalizable Network Architectures and Training Strategies for Semantic Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) (see pp. 33, 39, 135, 141).
- [123] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira. “Generalized ODIN: Detecting out-of-distribution image without learning from out-of-distribution data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10951–10960 (see p. 37).
- [124] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado. “GAIA-1: A Generative World Model for Autonomous Driving”. In: *arXiv preprint arXiv:2309.17080* (2023) (see p. 141).
- [125] X. Hu, L. Chen, B. Tang, D. Cao, and H. He. “Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles”. In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 482–500 (see p. 135).
- [126] Y. Hu, X. Jia, M. Tomizuka, and W. Zhan. “Causal-based time series domain generalization for vehicle intention prediction”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2022, pp. 7806–7813 (see p. 24).
- [127] J. Hwang, S. W. Oh, J.-Y. Lee, and B. Han. “Exemplar-Based Open-Set Panoptic Segmentation Network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1175–1184 (see pp. 37, 83).
- [128] P. A. Ioannou and C.-C. Chien. “Autonomous intelligent cruise control”. In: *Transactions on Vehicular technology* 42.4 (1993), pp. 657–672 (see p. 11).
- [129] A. Javornik, B. Marder, J. B. Barhorst, G. McLean, Y. Rogers, P. Marshall, and L. Warlop. “‘What lies behind the filter?’ Uncovering the motivations for using augmented reality (AR) face filters on social media and their effect on well-being”. In: *Computers in Human Behavior* 128 (2022), pp. 107–126 (see p. 22).

- [130] X. Jiang, S. Li, Y. Liu, S. Wang, F. Jia, T. Wang, L. Han, and X. Zhang. “Far3D: Expanding the Horizon for Surround-view 3D object detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 3. 2024, pp. 2561–2569 (see p. 33).
- [131] T. Jochem, D. Pomerleau, B. Kumar, and J. Armstrong. “PANS: A portable navigation platform”. In: *Proceedings of the Intelligent Vehicles Symposium*. IEEE. 1995, pp. 107–112 (see p. 11).
- [132] S. Jung, J. Lee, D. Gwak, S. Choi, and J. Choo. “Standardized Max Logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15425–15434 (see pp. 34, 37, 38).
- [133] J. Kabalar, S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. “Towards long-term retrieval-based visual localization in indoor environments with changes”. In: *IEEE Robotics and Automation Letters* 8.4 (2023), pp. 1975–1982 (see p. 24).
- [134] N. Kalra and S. M. Paddock. “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 182–193 (see p. 10).
- [135] C. Kamann and C. Rother. “Benchmarking the robustness of semantic segmentation models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8828–8838 (see p. 30).
- [136] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. “Traffic monitoring and accident detection at intersections”. In: *Transactions on Intelligent Transportation Systems* 1.2 (2000), pp. 108–118 (see p. 20).
- [137] K. Kania, K. M. Yi, M. Kowalski, T. Trzciński, and A. Tagliasacchi. “CoNeRF: Controllable neural radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18623–18632 (see p. 140).
- [138] M. A. Karaoglu, N. Brasch, M. Stollenga, W. Wein, N. Navab, F. Tombari, and A. Ladikos. “Adversarial domain feature adaptation for bronchoscopic depth estimation”. In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Part IV 24. Springer. 2021, pp. 300–310 (see p. 39).
- [139] A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe. “Keyframe-based visual-inertial online SLAM with relocalization”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2017, pp. 6662–6669 (see p. 24).
- [140] J. Katrolia, L. Krämer, J. R. Rambach, B. Mirbach, and D. Stricker. “An Adversarial Training based Framework for Depth Domain Adaptation”. In: *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Vol. 4: VISAPP. SCITEPRESS, 2021, pp. 353–361 (see p. 141).
- [141] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza. “Champion-level drone racing using deep reinforcement learning”. In: *Nature* 620.7976 (2023), pp. 982–987 (see p. 21).
- [142] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. “SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1521–1529 (see p. 21).
- [143] A. Kendall and Y. Gal. “What uncertainties do we need in Bayesian deep learning for computer vision?” In: *Advances in Neural Information Processing Systems* 30 (2017) (see p. 38).
- [144] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah. “Learning to drive in a day”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2019, pp. 8248–8254 (see pp. 12, 16, 139).
- [145] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. “3D Gaussian Splatting for real-time radiance field rendering”. In: *ACM Transactions on Graphics* 42.4 (2023), pp. 139–1 (see p. 140).

- [146] A. Khakzar, S. Baselizadeh, S. Khanduja, C. Rupprecht, S. T. Kim, and N. Navab. “Neural response interpretation through the lens of critical pathways”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13528–13538 (see p. 138).
- [147] M. A. Khan, H. E. Sayed, S. Malik, T. Zia, J. Khan, N. Alkaabi, and H. Ignatious. “Level-5 autonomous driving—Are we there yet? A review of research literature”. In: *Computing Surveys* 55.2 (2022), pp. 1–38 (see p. 6).
- [148] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim. “Learning not to learn: Training deep neural networks with biased data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9012–9020 (see p. 34).
- [149] A. Kirillov, R. Girshick, K. He, and P. Dollár. “Panoptic feature pyramid networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6399–6408 (see p. 44).
- [150] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. “Panoptic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413 (see pp. 24, 32, 43, 44, 83).
- [151] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick. “Segment Anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026 (see pp. 33, 83, 84, 137, 138, 141).
- [152] R. Kohavi. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *International Joint Conference on Artificial Intelligence*. Vol. 14. 2. 1995, pp. 1137–1145 (see p. 32).
- [153] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. J. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser. “Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12871–12881 (see p. 140).
- [154] Y. Kuznetsov, J. Stuckler, and B. Leibe. “Semi-supervised deep learning for monocular depth map prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6647–6655 (see p. 24).
- [155] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia. “Spherical transformer for lidar-based 3d recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17545–17555 (see p. 135).
- [156] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. “Deeper depth prediction with fully convolutional residual networks”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2016, pp. 239–248 (see p. 24).
- [157] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 6405–6416 (see p. 38).
- [158] E. Lamon, A. De Franco, L. Peternel, and A. Ajoudani. “A capability-aware role allocation approach to industrial assembly tasks”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3378–3385 (see p. 21).
- [159] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. “PointPillars: Fast encoders for object detection from point clouds”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12697–12705 (see pp. 26, 136).
- [160] M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl. “A cross-season correspondence dataset for robust semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9532–9542 (see p. 25).
- [161] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (see p. 32).

- [162] A. Lehner, S. Gasperini, A. Marcos-Ramiro, M. Schmidt, M.-A. N. Mahani, N. Navab, B. Busam, and F. Tombari. “3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17295–17304 (see pp. 1, 20, 23, 25, 26, 30, 33, 35, 36, 40, 83, 136–140, 145, 149, 211).
- [163] A. Lehner, S. Gasperini, A. Marcos-Ramiro, M. Schmidt, N. Navab, B. Busam, and F. Tombari. “3D adversarial augmentations for robust out-of-domain predictions”. In: *International Journal of Computer Vision* 132.3 (2024), pp. 931–963 (see pp. 1, 20, 26, 33, 35, 37, 40, 41, 69, 70, 136–140, 145).
- [164] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. “Coupled object detection and tracking from static cameras and moving vehicles”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1683–1698 (see p. 24).
- [165] Z. Leng, S.-C. Wu, M. Saleh, A. Montanaro, H. Yu, Y. Wang, N. Navab, X. Liang, and F. Tombari. “Dynamic Hyperbolic Attention Network for Fine Hand-object Reconstruction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 14894–14904 (see p. 22).
- [166] L. Li, V. A. Zimmer, W. Ding, F. Wu, L. Huang, J. A. Schnabel, and X. Zhuang. “Random style transfer based domain generalization networks integrating shape and spatial information”. In: *International Workshop on Statistical Atlases and Computational Models of the Heart, MICCAI 2020*. Springer. 2021, pp. 208–218 (see p. 39).
- [167] L. Li, V. A. Zimmer, J. A. Schnabel, and X. Zhuang. “AtrialGeneral: domain generalization for left atrial segmentation of multi-center LGE MRIs”. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Part VI 24. Springer. 2021, pp. 557–566 (see p. 39).
- [168] N. Li, C. P. Ho, J. Xue, L. W. Lim, G. Chen, Y. H. Fu, and L. Y. T. Lee. “A progress review on solid-state LiDAR and nanophotonics-based LiDAR sensors”. In: *Laser & Photonics Reviews* 16.11 (2022), p. 2100511 (see p. 135).
- [169] S. Li, M. M. Ozo, C. De Wagter, and G. C. de Croon. “Autonomous drone race: A computationally efficient vision-based navigation and control strategy”. In: *Robotics and Autonomous Systems* 133 (2020), p. 103621 (see p. 21).
- [170] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J. A. DeCastro, S. Karaman, and D. Rus. “Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3459–3466 (see p. 24).
- [171] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari. “Structure-SLAM: Low-drift monocular SLAM in indoor environments”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6583–6590 (see p. 21).
- [172] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari. “RGB-D SLAM with structural regularities”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2021, pp. 11581–11587 (see p. 24).
- [173] S. Liang, S. Wang, K. Li, M. Niemeyer, S. Gasperini, N. Navab, and F. Tombari. “SuperGSeg: Open-Vocabulary 3D Segmentation with Structured Super-Gaussians”. In: *arXiv preprint arXiv:2412.10231* (2024) (see p. 140).
- [174] J.-T. Lin, D. Dai, and L. Van Gool. “Depth estimation from monocular images and sparse radar data”. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. 2020, pp. 10233–10240 (see p. 103).
- [175] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common objects in context”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014 (see pp. 33, 84).
- [176] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles. “Spatiotemporal relationship reasoning for pedestrian intent prediction”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3485–3492 (see p. 24).

- [177] D. Liu, R. Yu, and H. Su. “Adversarial shape perturbations on 3D point clouds”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 88–104 (see pp. 69, 70).
- [178] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang. “An end-to-end network for panoptic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6172–6181 (see p. 46).
- [179] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan. “Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness”. In: *Advances in Neural Information Processing Systems*. 2020 (see pp. 37, 38, 83, 84).
- [180] L. Liu, X. Song, M. Wang, Y. Liu, and L. Zhang. “Self-supervised monocular depth estimation for all day images using domain separation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12737–12746 (see p. 115).
- [181] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys. “Towards robust visual odometry with a multi-camera system”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018, pp. 1154–1161 (see p. 24).
- [182] R. Liu, C. Yang, W. Sun, X. Wang, and H. Li. “StereoGAN: Bridging synthetic-to-real domain gap by joint optimization of domain translation and stereo matching”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12757–12766 (see p. 141).
- [183] S. Liu, B. Yu, J. Tang, Y. Zhu, and X. Liu. “Communication challenges in infrastructure-vehicle cooperative autonomous driving: A field deployment perspective”. In: *Wireless Communications* 29.4 (2022), pp. 126–131 (see p. 135).
- [184] W. Liu, X. Wang, J. Owens, and Y. Li. “Energy-based out-of-distribution detection”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21464–21475 (see p. 37).
- [185] Y. Liu, X. Chen, C. Liu, and D. Song. “Delving into transferable adversarial examples and black-box attacks”. In: *International Conference on Learning Representations*. 2017 (see p. 41).
- [186] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015 (see p. 44).
- [187] J. Luiten, T. Fischer, and B. Leibe. “Track to reconstruct and reconstruct to track”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1803–1810 (see p. 24).
- [188] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. “TrafficPredict: Trajectory prediction for heterogeneous traffic-agents”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6120–6127 (see p. 17).
- [189] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. “1 year, 1000 km: The Oxford RobotCar dataset”. In: *The International Journal of Robotics Research* 36.1 (2017), pp. 3–15 (see p. 35).
- [190] R. Madigan, T. Louw, M. Dziennus, T. Graindorge, E. Ortega, M. Graindorge, and N. Merat. “Acceptance of automated road transport systems (ARTS): an adaptation of the UTAUT model”. In: *Transportation Research Procedia* 14 (2016), pp. 2217–2226 (see p. 16).
- [191] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *International Conference on Learning Representations*. 2018 (see pp. 32, 40, 138, 140).
- [192] F. Magistri, E. Marks, S. Nagulavancha, I. Vizzo, T. L  be, J. Behley, M. Halstead, C. McCool, and C. Stachniss. “Contrastive 3D shape completion and reconstruction for agricultural robots using RGB-D frames”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10120–10127 (see p. 20).
- [193] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. “Deep model-based 6D pose refinement in RGB”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 800–815 (see p. 21).
- [194] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garc  a, and D. Scaramuzza. “Event-based vision meets deep learning on steering prediction for self-driving cars”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5419–5427 (see p. 23).

- [195] G. E. Marchant and R. A. Lindor. “The coming collision between autonomous vehicles and the liability system”. In: *Santa Clara L. Rev.* 52 (2012), p. 1321 (see pp. 14, 16).
- [196] K. Martin. “Ethical implications and accountability of algorithms”. In: *Journal of Business Ethics* 160 (2019), pp. 835–850 (see p. 14).
- [197] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016 (see p. 5).
- [198] M. Maximov, K. Galim, and L. Leal-Taixé. “Focus on defocus: Bridging the synthetic to real domain gap for depth estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1071–1080 (see p. 141).
- [199] K. McGuire, G. De Croon, C. De Wagter, K. Tuyls, and H. Kappen. “Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1070–1076 (see p. 21).
- [200] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. “A survey on bias and fairness in machine learning”. In: *ACM Computing Surveys* 54.6 (2021), pp. 1–35 (see p. 34).
- [201] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. “NeRF: Representing scenes as neural radiance fields for view synthesis”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 405–421 (see p. 140).
- [202] A. Milioto, J. Behley, C. McCool, and C. Stachniss. “LiDAR Panoptic Segmentation for Autonomous Driving”. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. 2020 (see p. 44).
- [203] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf. “Evaluating merging strategies for sampling-based uncertainty techniques in object detection”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2019, pp. 2348–2354 (see pp. 58, 138).
- [204] F. Milletari, N. Navab, and S.-A. Ahmadi. “V-Net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2016, pp. 565–571 (see p. 29).
- [205] A. A. Mohammed, K. Ambak, A. M. Mosa, and D. Syamsunur. “A review of traffic accidents and related practices worldwide”. In: *The Open Transportation Journal* 13.1 (2019) (see pp. 5, 10).
- [206] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. “Universal adversarial perturbations”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1765–1773 (see p. 41).
- [207] H. P. Moravec. “The Stanford cart and the CMU rover”. In: *Proceedings of the IEEE* 71.7 (1983), pp. 872–884 (see p. 11).
- [208] I. Mouawad, N. Brasch, F. Manhardt, F. Tombari, and F. Odone. “Time-to-Label: Temporal Consistency for Self-Supervised Monocular 3D Object Detection”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 8988–8995 (see pp. 23, 139).
- [209] F. Munir, S. Azam, and M. Jeon. “SSTN: Self-supervised domain adaptation thermal object detection for autonomous driving”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2021, pp. 206–213 (see p. 23).
- [210] N. Navab, A. Martin-Gomez, M. Seibold, M. Sommersperger, T. Song, A. Winkler, K. Yu, and U. Eck. “Medical augmented reality: definition, principle components, domain modeling, and design-development-validation process”. In: *Journal of Imaging* 9.1 (2022), p. 4 (see p. 22).
- [211] A. Nekrasov, A. Hermans, L. Kuhnert, and B. Leibe. “UGainS: Uncertainty guided anomaly instance segmentation”. In: *DAGM German Conference on Pattern Recognition*. Springer. 2023, pp. 50–66 (see p. 38).

- [212] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann. “Mix3D: Out-of-context data augmentation for 3D scenes”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2021, pp. 116–125 (see p. 32).
- [213] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kotschieder. “The Mapillary Vistas dataset for semantic understanding of street scenes”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2017 (see p. 35).
- [214] S. J. Oh, M. Fritz, and B. Schiele. “Adversarial image perturbation for privacy protection a game theory perspective”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1491–1500 (see p. 40).
- [215] OpenAI (2023). “GPT-4 Technical Report”. In: *arXiv preprint arXiv:2303.08774* (2023) (see pp. 15, 137, 138).
- [216] E. Özsoy, T. Czempel, F. Holm, C. Pellegrini, and N. Navab. “LABRAD-OR: Lightweight Memory Scene Graphs for Accurate Bimodal Reasoning in Dynamic Operating Rooms”. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer. 2012 (see p. 21).
- [217] N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, and N. Navab. “Statistical modeling and recognition of surgical workflow”. In: *Medical Image Analysis* 16.3 (2012), pp. 632–641 (see p. 21).
- [218] D. Paez-Granados, Y. He, D. Gonon, D. Jia, B. Leibe, K. Suzuki, and A. Billard. “Pedestrian-robot interactions on autonomous crowd navigation: Reactive control methods and evaluation metrics”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2022, pp. 149–156 (see p. 20).
- [219] C. Pakusch, G. Stevens, A. Boden, and P. Bossauer. “Unintended effects of autonomous driving: A study on mobility preferences in the future”. In: *Sustainability* 10.7 (2018), p. 2404 (see p. 11).
- [220] J. Palacin, J. A. Salse, I. Valgañón, and X. Clua. “Building a mobile robot for a floor-cleaning operation in domestic environments”. In: *Transactions on Instrumentation and Measurement* 53.5 (2004), pp. 1418–1424 (see p. 21).
- [221] F. Paredes-Vallés and G. C. de Croon. “Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3446–3455 (see p. 21).
- [222] P. Parekh, S. Patel, N. Patel, and M. Shah. “Systematic review and meta-analysis of augmented reality in medicine, retail, and games”. In: *Visual Computing for Industry, Biomedicine, and Art* 3 (2020), pp. 1–20 (see p. 22).
- [223] J. Park, B. G. Jeong, S. I. Kim, D. Lee, J. Kim, C. Shin, C. B. Lee, T. Otsuka, J. Kyoung, S. Kim, et al. “All-solid-state spatial light modulator with independent phase and amplitude control for three-dimensional LiDAR applications”. In: *Nature Nanotechnology* 16.1 (2021), pp. 69–76 (see p. 23).
- [224] W. Park, S. Seo, and J. Bae. “A hybrid gripper with soft material and rigid structures”. In: *IEEE Robotics and Automation Letters* 4.1 (2018), pp. 65–72 (see p. 21).
- [225] M. Paschali, S. Gasperini, A. G. Roy, M. Y.-S. Fang, and N. Navab. “3DQ: Compact quantized neural networks for volumetric whole brain segmentation”. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer. 2019, pp. 438–446 (see p. 25).
- [226] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang Jr. “Perception, planning, control, and coordination for autonomous vehicles”. In: *Machines* 5.1 (2017), p. 6 (see p. 11).
- [227] A. Piccoli, M.-O. Pahl, and L. Wüstrich. “Group key management in constrained IoT settings”. In: *IEEE Symposium on Computers and Communications*. 2020, pp. 1–6 (see p. 135).

- [228] B. J. Pijnacker Hordijk, K. Y. Scheper, and G. C. De Croon. “Vertical landing for micro air vehicles using event-based optical flow”. In: *Journal of Field Robotics* 35.1 (2018), pp. 69–90 (see p. 21).
- [229] S. Pillath. “Automated vehicles in the EU”. In: (2016) (see p. 15).
- [230] M. Planamente, C. Plizzari, M. Cannici, M. Ciccone, F. Strada, A. Bottino, M. Matteucci, and B. Caputo. “DA4Event: Towards bridging the sim-to-real gap for event cameras using domain adaptation”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6616–6623 (see p. 33).
- [231] C. Plizzari, S. Goel, T. Perrett, J. Chalk, A. Kanazawa, and D. Damen. “Spatial cognition from egocentric video: Out of sight, not out of mind”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2025 (see p. 22).
- [232] C. Plizzari, G. Goletto, A. Furnari, S. Bansal, F. Ragusa, G. M. Farinella, D. Damen, and T. Tommasi. “An outlook into the future of egocentric vision”. In: *International Journal of Computer Vision* 132.11 (2024), pp. 4880–4936 (see p. 22).
- [233] C. Plizzari, T. Perrett, B. Caputo, and D. Damen. “What Can a Cook in Italy Teach a Mechanic in India? Action Recognition Generalisation Over Scenarios and Locations”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 13656–13666 (see pp. 20, 22).
- [234] C. Plizzari, M. Planamente, G. Goletto, M. Cannici, E. Gusso, M. Matteucci, and B. Caputo. “E2 (GO) MOTION: Motion augmented event stream for egocentric action recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19935–19947 (see p. 22).
- [235] C. Plizzari, A. Tonioni, Y. Xian, A. Kulshrestha, and F. Tombari. “Omnia de EgoTempo: Benchmarking Temporal Understanding of Multi-Modal LLMs in Egocentric Videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025 (see p. 22).
- [236] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. “Full-resolution residual networks for semantic segmentation in street scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4151–4160 (see p. 24).
- [237] D. Pomerleau. “RALPH: Rapidly adapting lateral position handler”. In: *Proceedings of the Intelligent Vehicles Symposium*. IEEE. 1995, pp. 506–511 (see p. 11).
- [238] D. A. Pomerleau. “ALVINN: An autonomous land vehicle in a neural network”. In: *Advances in Neural Information Processing Systems*. 1989, pp. 305–313 (see p. 11).
- [239] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari. “Sampling-free epistemic uncertainty estimation using approximated variance propagation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2931–2940 (see p. 38).
- [240] J. Postels, Y. Strümpfer, K. Reichard, L. Van Gool, and F. Tombari. “3D Compression Using Neural Fields”. In: *arXiv preprint arXiv:2311.13009* (2023) (see p. 140).
- [241] C. V. Poulton, A. Yaacobi, D. B. Cole, M. J. Byrd, M. Raval, D. Vermeulen, and M. R. Watts. “Coherent solid-state LIDAR with silicon photonic optical phased arrays”. In: *Optics letters* 42.20 (2017), pp. 4091–4094 (see p. 23).
- [242] L. Prechelt. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69 (see p. 32).
- [243] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. “Learning transferable visual models from natural language supervision”. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763 (see pp. 33, 37, 83, 84, 137, 138).
- [244] U. Rafi, J. Gall, and B. Leibe. “A semantic occlusion model for human pose estimation from a single depth image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2015, pp. 67–74 (see p. 24).

- [245] M. Raghu, K. Blumer, R. Sayres, Z. Obermeyer, B. Kleinberg, S. Mullainathan, and J. Kleinberg. “Direct uncertainty prediction for medical second opinions”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5281–5290 (see p. 38).
- [246] M.-J. Rakotosaona, F. Manhardt, D. M. Arroyo, M. Niemeyer, A. Kundu, and F. Tombari. “NeRFMeshing: Distilling Neural Radiance Fields into Geometrically-Accurate 3D Meshes”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2023 (see p. 140).
- [247] A. Ramazzina, M. Bijelic, S. Walz, A. Sanvito, D. Scheuble, and F. Heide. “ScatterNeRF: Seeing Through Fog with Physically-Based Inverse Neural Rendering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 17957–17968 (see p. 35).
- [248] A. Ramazzina, S. Walz, P. Dahal, M. Bijelic, and F. Heide. “Gated Fields: Learning Scene Reconstruction from Gated Videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 10530–10541 (see p. 23).
- [249] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos. “PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6262–6271 (see p. 24).
- [250] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari. “Urban radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12932–12942 (see p. 140).
- [251] A. Remus, S. D’Avella, F. Di Felice, P. Tripicchio, and C. A. Avizzano. “i2c-net: Using Instance-Level Neural Networks for Monocular Category-Level 6D Pose Estimation”. In: *IEEE Robotics and Automation Letters* 8.3 (2023), pp. 1515–1522 (see p. 21).
- [252] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 91–99 (see p. 44).
- [253] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695 (see pp. 140, 141).
- [254] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. “Learning in an uncertain world: Representing ambiguity through multiple hypotheses”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3591–3600 (see p. 38).
- [255] D. von Rütte, E. Fedele, J. Thomm, and L. Wolf. “FABRIC: Personalizing Diffusion Models with Iterative Feedback”. In: *arXiv preprint arXiv:2307.10159* (2023) (see p. 141).
- [256] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. “Adapting visual category models to new domains”. In: *Proceedings of the European Conference on Computer Vision*. Part IV 11. Springer. 2010, pp. 213–226 (see p. 39).
- [257] S. Saha, L. Hoyer, A. Obukhov, D. Dai, and L. Van Gool. “EDAPS: Enhanced Domain-Adaptive Panoptic Segmentation”. In: 2023 (see p. 141).
- [258] C. Sakaridis, D. Dai, and L. Van Gool. “ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10765–10775 (see p. 12).
- [259] M. Saleh, S.-C. Wu, L. Cosmo, N. Navab, B. Busam, and F. Tombari. “Bending Graphs: Hierarchical shape matching using gated optimal transport”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11757–11767 (see p. 21).
- [260] J. Sanchez, J.-E. Deschaud, and F. Goulette. “Domain generalization of 3D semantic segmentation in autonomous driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 18077–18087 (see pp. 39, 135).

- [261] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, et al. “Back to the Feature: Learning robust camera localization from pixels to pose”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3247–3257 (see p. 24).
- [262] T. Sattler, B. Leibe, and L. Kobbelt. “Efficient & effective prioritized matching for large-scale image-based localization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.9 (2016), pp. 1744–1756 (see p. 24).
- [263] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. “Benchmarking 6DoF outdoor visual localization in changing conditions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8601–8610 (see p. 24).
- [264] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet. “Image-to-LiDAR self-supervised distillation for autonomous driving data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9891–9901 (see p. 12).
- [265] A. Savkin, R. Ellouze, N. Navab, and F. Tombari. “Unsupervised traffic scene generation with synthetic 3D scene graphs”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2021, pp. 1229–1235 (see pp. 33, 39, 140).
- [266] O. Scheel, N. S. Nagaraja, L. Schwarz, N. Navab, and F. Tombari. “Attention-based lane change prediction”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2019, pp. 8655–8661 (see p. 17).
- [267] C. Schlarman and M. Hein. “On the adversarial robustness of multi-modal foundation models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2023, pp. 3677–3685 (see p. 137).
- [268] S. Schmidt and S. Günnemann. “Stream-based active learning by exploiting temporal properties in perception with temporal predicted loss”. In: *British Machine Vision Conference*. 2023 (see p. 26).
- [269] S. Schmidt, J. Körner, D. Fuchsgruber, S. Gasperini, F. Tombari, and S. Günnemann. “Prior2Former—Evidential Modeling of Mask Transformers for Assumption-Free Open-World Panoptic Segmentation”. In: *arXiv preprint arXiv:2504.04841* (2025) (see pp. 33–35, 37, 38, 136).
- [270] S. Schmidt, L. Schenk, L. Schwinn, and S. Günnemann. “Joint Out-of-Distribution Filtering and Data Discovery Active Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025 (see p. 26).
- [271] S. Schmidt, L. Stappen, L. Schwinn, and S. Günnemann. “Generalized Synchronized Active Learning for Multi-Agent-Based Data Selection on Mobile Robotic Systems”. In: *IEEE Robotics and Automation Letters* (2024) (see p. 26).
- [272] S. Schmidt, L. Stumpp, D. Valverde, and S. Günnemann. “Deep Sensor Fusion with Constraint Safety Bounds for High Precision Localization”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 12256–12262 (see p. 17).
- [273] M. Schneider, J. Rambach, and D. Stricker. “Augmented reality based on edge computing using the example of remote live support”. In: *Proceedings of the IEEE International Conference on Industrial Technology*. 2017, pp. 1277–1282 (see p. 21).
- [274] M. Schwall, T. Daniel, T. Victor, F. Favaro, and H. Hohnhold. “Waymo public road safety performance data”. In: *arXiv preprint arXiv:2011.00038* (2020) (see p. 12).
- [275] M. Sensoy, L. M. Kaplan, and M. Kandemir. “Evidential Deep Learning to Quantify Classification Uncertainty”. In: *Advances in Neural Information Processing*. 2018, pp. 3183–3193 (see pp. 38, 136).
- [276] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. “Adversarial training for free!” In: *Advances in Neural Information Processing Systems* 32 (2019) (see p. 41).

- [277] R. R. Shetty, M. Fritz, and B. Schiele. “Adversarial scene editing: Automatic object removal from weak supervision”. In: *Advances in Neural Information Processing Systems* 31 (2018) (see p. 141).
- [278] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang. “MVDream: Multi-view diffusion for 3D generation”. In: *arXiv preprint arXiv:2308.16512* (2023) (see p. 141).
- [279] C. Shorten and T. M. Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of Big Data* 6.1 (2019), pp. 1–48 (see p. 32).
- [280] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. “Implicit neural representations with periodic activation functions”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7462–7473 (see p. 140).
- [281] B. W. Smith. “Automated vehicles are probably legal in the United States”. In: *Tex. A&M L. Rev.* 1 (2013), p. 411 (see p. 15).
- [282] F. E. Sotiropoulos and H. H. Asada. “A model-free extremum-seeking approach to autonomous excavator control based on output power maximization”. In: *Robotics and Automation Letters* 4.2 (2019), pp. 1005–1012 (see p. 135).
- [283] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (see p. 32).
- [284] S. Steyer, G. Tanzmeister, and D. Wollherr. “Grid-based environment estimation using evidential mapping and particle tracking”. In: *IEEE Transactions on Intelligent Vehicles* 3.3 (2018), pp. 384–396 (see p. 12).
- [285] Y. Strümpfer, J. Postels, R. Yang, L. V. Gool, and F. Tombari. “Implicit neural representations for image compression”. In: *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 74–91 (see p. 140).
- [286] D. Stutz, M. Hein, and B. Schiele. “Disentangling adversarial robustness and generalization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6976–6987 (see p. 32).
- [287] D. Stutz, M. Hein, and B. Schiele. “Relating adversarially robust generalization to flat minima”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7807–7817 (see p. 32).
- [288] Y. Su, J. Rambach, N. Minaskan, P. Lesur, A. Pagani, and D. Stricker. “Deep multi-state object pose estimation for augmented reality assembly”. In: *IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 2019, pp. 222–227 (see p. 21).
- [289] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari. “ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6738–6748 (see p. 21).
- [290] P. Sun, H. Kretschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. “Scalability in perception for autonomous driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2446–2454 (see p. 12, 22, 26).
- [291] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu. “SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 21371–21382 (see p. 141).
- [292] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. “Intriguing properties of neural networks”. In: *International Conference on Learning Representations*. 2014 (see pp. 40, 41).
- [293] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann. “OpenMask3D: Open-Vocabulary 3D Instance Segmentation”. In: *Advances in Neural Information Processing Systems*. 2023 (see p. 38).

- [294] K. Tateno, F. Tombari, I. Laina, and N. Navab. “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6243–6252 (see p. 24).
- [295] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Tech. rep. J3016_201806. SAE International, 2018 (see pp. 9, 10).
- [296] R. R. Teetor. *Speed control device for resisting operation of the accelerator*. Aug. 1950 (see p. 11).
- [297] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer. “Vision and navigation for the Carnegie-Mellon Navlab”. In: *Transactions on Pattern Analysis and Machine Intelligence* 10.3 (1988), pp. 362–373 (see p. 11).
- [298] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. “Stanley: The robot that won the DARPA Grand Challenge”. In: *Journal of Field Robotics* 23.9 (2006), pp. 661–692 (see pp. 11, 12).
- [299] C. Tomasi and R. Manduchi. “Bilateral filtering for gray and color images”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 1998, pp. 839–846 (see p. 136).
- [300] A. Torralba and A. Efros. “Unbiased look at dataset bias”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 1521–1528 (see p. 33).
- [301] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. “Ensemble adversarial training: Attacks and defenses”. In: *International Conference on Learning Representations*. 2017 (see p. 41).
- [302] F. Tristram, S. Gasperini, N. Navab, and F. Tombari. “Re-Nerfing: Improving Novel View Synthesis through Novel View Synthesis”. In: *arXiv preprint arXiv:2312.02255* (2023) (see p. 140).
- [303] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari. “SPARF: Neural radiance fields from sparse and noisy poses”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4190–4200 (see p. 140).
- [304] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. “Robustness May Be at Odds with Accuracy”. In: *International Conference on Learning Representations*. 2019 (see p. 32).
- [305] S. Tsugawa, S. Jeschke, and S. E. Shladover. “A review of truck platooning projects for energy savings”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 68–77 (see p. 10).
- [306] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun. “Physically realizable adversarial examples for LiDAR object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13713–13722 (see pp. 12, 138).
- [307] S. Ullman. “The interpretation of structure from motion”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426 (see pp. 101, 116).
- [308] D. Ungureanu, F. Bogo, S. Galliani, P. Sama, X. Duan, C. Meekhof, J. Stühmer, T. J. Cashman, B. Tekin, J. L. Schönberger, et al. “HoloLens 2 research mode as a tool for computer vision research”. In: *arXiv preprint arXiv:2008.11239* (2020) (see p. 22).
- [309] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. “Autonomous driving in urban environments: Boss and the Urban Challenge”. In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466 (see p. 12).
- [310] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. “Uncertainty estimation using a single deep deterministic neural network”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9690–9700 (see pp. 38, 57, 136).
- [311] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. “IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments”. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 2019, pp. 1743–1751 (see pp. 31, 35).

- [312] Y. Velikova, M. F. Azampour, W. Simson, V. Gonzalez Duque, and N. Navab. “LOTUS: Learning to Optimize Task-based US representations”. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2023, pp. 435–445 (see p. 25).
- [313] Y. Velikova, W. Simson, M. Salehi, M. F. Azampour, P. Paprottka, and N. Navab. “CACTUSS: Common anatomical CT-US space for US examinations”. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2022, pp. 492–501 (see pp. 25, 33, 39).
- [314] N. E. Vellinga. “Automated driving and its challenges to international traffic law: which way to go?” In: *Law, Innovation and Technology* 11.2 (2019), pp. 257–278 (see p. 15).
- [315] T. Victor, K. Kusano, T. Gode, R. Chen, and M. Schwall. *Safety performance of the Waymo rider-only automated driving system at one million miles*. Tech. rep. Technical report, Waymo LLC, 2023 (see pp. 6, 10, 14).
- [316] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese. “Generalizing to unseen domains via adversarial data augmentation”. In: *Advances in Neural Information Processing Systems* 31 (2018) (see pp. 30, 139).
- [317] Z. Wadud, D. MacKenzie, and P. Leiby. “Help or hindrance? The travel, energy and carbon impacts of highly automated vehicles”. In: *Transportation Research Part A: Policy and Practice* 86 (2016), pp. 1–18 (see p. 10).
- [318] J. Wald, N. Navab, and F. Tombari. “Learning 3D semantic scene graphs with instance embeddings”. In: *International Journal of Computer Vision* 130.3 (2022), pp. 630–651 (see p. 140).
- [319] J. Wald, T. Sattler, S. Golodetz, T. Cavallari, and F. Tombari. “Beyond controlled environments: 3D camera re-localization in changing indoor scenes”. In: *Proceedings of the European Conference on Computer Vision*. Vol. 16.VII. Springer. 2020, pp. 467–487 (see p. 21).
- [320] S. Walz, M. Bijelic, A. Ramazzina, A. Walia, F. Mannan, and F. Heide. “Gated Stereo: Joint Depth Estimation from Gated and Wide-Baseline Active Stereo Cues”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13252–13262 (see p. 23).
- [321] A. Wang, A. Jasour, and B. C. Williams. “Non-Gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6041–6048 (see p. 21).
- [322] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. “YOLOv7 Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7464–7475 (see p. 33).
- [323] G. Wang, F. Manhardt, F. Tombari, and X. Ji. “GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16611–16621 (see p. 21).
- [324] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren. “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks”. In: *Neurocomputing* 338 (2019), pp. 34–45 (see p. 38).
- [325] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. “MaX-DeepLab: End-to-end panoptic segmentation with mask transformers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5463–5474 (see p. 46).
- [326] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu. “Generalizing to unseen domains: A survey on domain generalization”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022) (see p. 39).
- [327] K. Wang, Z. Zhang, Z. Yan, X. Li, B. Xu, J. Li, and J. Yang. “Regularizing Nighttime Weirdness: Efficient self-supervised monocular depth estimation in the dark”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16055–16064 (see p. 115).

- [328] S. Wang, Q. Cheng, S. Gasperini, W. Zhang, S.-C. Wu, N. Zeller, D. Cremers, and N. Navab. “VoxNeRF: Bridging Voxel Representation and Neural Radiance Fields for Enhanced Indoor View Synthesis”. In: *IEEE Robotics and Automation Letters* (2025) (see p. 140).
- [329] T. D. Wang, L. Bastian, T. Czempel, C. Heiliger, and N. Navab. “Beyond Role-Based Surgical Domain Modeling: Generalizable Re-Identification in the Operating Room”. In: *Medical Image Analysis* (2025) (see p. 21).
- [330] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al. “InternImage: Exploring large-scale vision foundation models with deformable convolutions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 14408–14419 (see p. 33).
- [331] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao. “Train in Germany, test in the USA: Making 3D object detectors generalize”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11713–11723 (see pp. 36, 39).
- [332] O. Wasenmüller, M. Meyer, and D. Stricker. “Augmented reality 3d discrepancy check in industrial applications”. In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2016, pp. 125–134 (see p. 22).
- [333] T. Watanabe, N. Kishimoto, K. Hayafune, K. Yamada, and N. Maede. “Development of an intelligent cruise control system”. In: *Steps Forward. Intelligent Transport Systems World Congress*. 1995 (see p. 11).
- [334] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. “Information-theoretic model predictive control: Theory and applications to autonomous driving”. In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1603–1622 (see pp. 17, 135).
- [335] T. Winkle. “Safety benefits of automated vehicles: Extended findings from accident research for development, validation and testing”. In: *Autonomous driving: Technical, legal and social aspects* (2016), pp. 335–364 (see p. 5).
- [336] E. Wong, L. Rice, and J. Z. Kolter. “Fast is better than free: Revisiting adversarial training”. In: *International Conference on Learning Representations*. 2019 (see pp. 41, 69).
- [337] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari. “Incremental 3D Semantic Scene Graph Prediction from RGB Sequences”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5064–5074 (see p. 140).
- [338] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari. “SCFusion: Real-time incremental scene reconstruction with semantic completion”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2020, pp. 801–810 (see p. 21).
- [339] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. “SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7515–7525 (see pp. 20, 140).
- [340] M. Wysocki, M. F. Azampour, C. Eilers, B. Busam, M. Salehi, and N. Navab. “Ultra-NeRF: Neural radiance fields for ultrasound imaging”. In: *Medical Imaging with Deep Learning*. PMLR. 2024, pp. 382–401 (see p. 140).
- [341] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. “Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.9 (2018), pp. 2251–2265 (see p. 37).
- [342] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. “Feature generating networks for zero-shot learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5542–5551 (see p. 37).
- [343] C. Xiang, C. R. Qi, and B. Li. “Generating 3D adversarial point clouds”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9136–9144 (see pp. 69, 70, 140).

- [344] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. “Neural fields in visual computing and beyond”. In: *Computer Graphics Forum*. Vol. 41. 2. Wiley. 2022, pp. 641–676 (see p. 140).
- [345] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. “UPSNet: A unified panoptic segmentation network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8818–8826 (see pp. 44–46).
- [346] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai. “A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 736–753 (see pp. 83, 137, 138).
- [347] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li. “Depth completion from sparse LiDAR data with depth-normal constraints”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2811–2820 (see p. 24).
- [348] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. “Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2272–2281 (see p. 39).
- [349] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li. “2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 677–695 (see p. 33).
- [350] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo. “Towards photo-realistic virtual try-on by adaptively generating-preserving image content”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7850–7859 (see p. 22).
- [351] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen. “DeeperLab: Single-shot image parser”. In: *arXiv preprint arXiv:1902.05093* (2019) (see p. 44).
- [352] Z. Yang, Y. Zhou, Z. Chen, and J. Ngiam. “3D-MAN: 3D multi-frame attention network for object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1863–1872 (see p. 139).
- [353] S. Yenamandra, P. Ramesh, V. Prabhu, and J. Hoffman. “FACTS: First Amplify Correlations and Then Slice to Discover Bias”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4794–4804 (see pp. 34, 35).
- [354] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. “BDD100K: A diverse driving dataset for heterogeneous multitask learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2636–2645 (see pp. 31, 35).
- [355] H. Yu, G. C. E. de Croon, and C. De Wagter. “AvoidBench: A high-fidelity vision-based obstacle avoidance benchmarking suite for multi-rotors”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2023, pp. 9183–9189 (see p. 21).
- [356] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang. “Counterfactual zero-shot and open-set visual recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15404–15414 (see p. 37).
- [357] F. Zenke, B. Poole, and S. Ganguli. “Continual learning through synaptic intelligence”. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2017, pp. 3987–3995 (see pp. 26, 33).
- [358] G. Zhai, D. Huang, S.-C. Wu, H. Jung, Y. Di, F. Manhardt, F. Tombari, N. Navab, and B. Busam. “MonoGraspNet: 6-DoF grasping with a single RGB image”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2023, pp. 1708–1714 (see p. 21).
- [359] G. Zhai, E. P. Örnek, S.-C. Wu, Y. Di, F. Tombari, N. Navab, and B. Busam. “CommonScenes: Generating Commonsense 3D Indoor Scenes with Scene Graph Diffusion”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 30026–30038 (see p. 140).

- [360] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer. “LiT: Zero-shot transfer with locked-image text tuning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18123–18133 (see p. 37).
- [361] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115 (see p. 32).
- [362] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. “Efficient neural network robustness certification with general activation functions”. In: *Advances in Neural Information Processing Systems* 31 (2018) (see p. 32).
- [363] M. Zhang and C. Ré. “Contrastive adapters for foundation model group robustness”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21682–21697 (see pp. 137, 138).
- [364] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong. “Adversarial AutoAugment”. In: *International Conference on Learning Representations*. 2019 (see p. 41).
- [365] Y. Zhang, A. Khakzar, Y. Li, A. Farshad, S. T. Kim, and N. Navab. “Fine-grained neural network explanation by identifying input features with predictive information”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20040–20051 (see p. 138).
- [366] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia. “MonoViT: Self-supervised monocular depth estimation with a vision transformer”. In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. 2022, pp. 668–678 (see p. 137).
- [367] X. Zhou, D. Wang, and P. Krähenbühl. “Objects as points”. In: *arXiv preprint arXiv:1904.07850* (2019) (see pp. 26, 57, 58).
- [368] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. “Cylindrical and asymmetrical 3D convolution networks for lidar segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9939–9948 (see p. 26).
- [369] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. “Object detection in 20 years: A survey”. In: *Proceedings of the IEEE* (2023) (see p. 17).

List of Figures

1.1	Representations of autonomous vehicles created with DALL·E 3 [23]. The left image shows a fictional autonomous car in the streets of Munich, Germany, equipped with perception sensors such as a LiDAR on the roof. As depicted on the right side, the high autonomy enables all passengers to focus on activities such as working or reading as the vehicle drives itself.	5
2.1	Representation of the 6 levels of autonomy [295]. At Level 0, no automation is in place. Level 1 includes lateral or longitudinal support. Level 2 features both kinds of support, i.e., adaptive cruise control and lane following. At Level 3, the responsibility shifts from the driver to the vehicle. With Level 4, no driver is required under certain conditions, e.g., geofenced. Level 5 indicates full and unconstrained automation.	10
2.2	Examples of successful earlier efforts towards autonomous driving. The left picture shows the autonomous desert racing vehicle Stanley [298] by Stanford University. The Stanford team won the 2005 DARPA Grand Challenge as Stanley autonomously drove through the course in the Mojave Desert. The vehicle is a Volkswagen Tuareg with various sensors, including LiDARs on the roof. The right image shows the first autonomous vehicle built by Google around 2015. The car is a custom-built two-seater without driving controls fitted with perception sensors such as the roof-mounted LiDAR. Earlier, starting in 2009, Google was testing on retrofitted Toyota Prius and Lexus RX. Then, in 2016, Google's Self-Driving Car Project became Waymo. The right image is taken from Wikipedia under the license https://creativecommons.org/licenses/by-sa/4.0/	12
2.3	Examples of assisted driving features already available in production cars. Both vehicles shown offer Level 2 functionalities, including adaptive cruise control and lane following. BMW equips its vehicles with radars, cameras, and ultrasonic sensors (soon also LiDARs), while Tesla advocates a camera-only approach (although it may soon add back radars). The BMW 8 Series was captured in February 2020 on a rainy day in the German Autobahn. The controls for the driving assistance features are shown on the left spoke of the steering wheel, including the dial for adjusting the speed, the buttons for the distance to keep from the vehicle in front, and the button to activate lane following. BMW features lights on the steering wheel (above the buttons on the spokes) to signal the engagement of the assistance features and call for the driver's attention with yellow or red flashes alongside audible warnings. The picture of the Tesla Model 3 was taken in October 2023 at a pedestrian crossing in the center of Bologna, Italy. Both cars provide live feedback of what the vehicle is perceiving. BMW indicates this in a simplified view in the instrument cluster and head-up displays, while Tesla provides more information in the central display, including pedestrians, traffic signs, and traffic lights. The images have been anonymized, hiding the faces of the pedestrians.	13

2.4	Vehicles by Waymo and Zoox equipped with autonomous driving sensors and features. The pictures were captured in the streets of San Francisco, California, in June 2023. The Waymo car on the left has no driver and can offer autonomous rides to selected users, operating as a robot taxi. Vehicles like the retrofitted white Jaguar I-Pace shown are equipped with many sensors in various housings, especially around the roof area. Zoox is developing a mini-bus without driving controls on board and is testing on public roads with retrofitted Toyota Highlander SUVs equipped with multiple sensor rigs featuring numerous sensors, especially around the roof and front of the vehicle. While the sensor configuration between the two is different, both companies opted for multiple LiDARs, cameras, and radars. The images have been anonymized, hiding license plates and vehicle identifiers.	14
2.5	Representation of the standard autonomous driving software stack. Sensory inputs and an HD map are processed by the perception module, which performs scene understanding and localization. This feeds the planning component responsible for estimating the ego trajectory given that of others. The control module finally executes the computed trajectory. The focus of this dissertation is on scene understanding, highlighted in the diagram by the blue box.	16
3.1	Examples of scene understanding tasks and applications, from outdoor to indoor. Autonomous driving is a prominent application for scene understanding, both in 2D and 3D. Robotics, augmented reality, and healthcare are also popular applications in 2D and 3D as well. Image sources: 3D object detection (top) from nuScenes [35], panoptic segmentation from Cityscapes [50], depth estimation from our md4all [86], 3D object detection (middle) from our 3D-VField [162], 3D semantic segmentation from our [163], 4D segmentation across space and time from SegmentOR [15], activity recognition of videos from ARGO1M [233], 3D shape completion and panoptic segmentation through incremental 3D scene graphs from SceneGraphFusion [339].	20
3.2	Example of LiDAR and radar scans projected to the corresponding image. The data was captured on a clear day in Singapore and is part of the nuScenes dataset [35]. Radars offer sparse measurements with significant levels of noise due to multi-path effects. The points are colored depending on their distance. The arrows indicate two radar points projecting on the same car but measured to be over 5 meters apart [83] [©2021 IEEE]. This is due to the radar projecting along a horizontal plane, so higher points correspond to further measurements, and the further point is due to the radar waves reaching beyond the minivan passing underneath it. LiDARs are less noisy and denser but suffer from other issues, such as artifacts with adverse weather (Section 5.3.2). The LiDAR scan shown is not particularly dense as it was captured with a 32-beam sensor. Other sensors reach 64 [90, 290] or 128 beams, offering higher resolutions.	22
3.3	Representation of various 2D scene understanding tasks for autonomous driving. Object detection aims to find the presence and location of objects of interest (e.g., cars). Depth estimation is about predicting a distance measurement for each pixel in input. Semantic segmentation assigns a semantic class to each pixel. Instance segmentation identifies the individual pixels of an object. Panoptic segmentation combines semantic and instance segmentation. Object detection can also be performed in 3D from 2D or 3D data. Segmentation tasks can also be performed on 3D data, e.g., assigning semantic estimates to each point of a 3D point cloud instead of a pixel.	24

4.1	Examples of data from KITTI [90], BDD100K [354], nuScenes [35], and the Indian Driving Dataset [311]. The pioneering KITTI dataset includes mainly average cases around German residential areas. In contrast, the other, more recent datasets feature more challenging scenarios, such as difficult illumination conditions, large construction sites, and unstructured scenes. The differences highlight the domain gaps between the datasets and the diversity of traffic scenarios.	31
4.2	Example of possible data bias due to the vehicles present in a dataset. If the training data is not diverse enough for the model to abstract the characteristics of the objects and scenes of interest (e.g., contains only one car model), it may develop a bias towards the training data distribution, failing with other scenarios that are not part of it (e.g., with unseen colors and models).	34
4.3	Example of challenging weather and illumination conditions from nuScenes [35], BDD100K [354], Boreas [34], and RobotCar [189]. Compared to standard illumination settings in clear or cloudy scenarios, the conditions shown pose a substantial challenge due to noise, blurriness, reflections, lack of texture, and hidden information in the dark or under snow. Additionally, with adverse weather, LiDAR sensors introduce artifacts that need to be accounted for (Section 5.3.2).	35
4.4	Adversarial examples on images and point clouds. At the top, an adversarial pattern is added to the image of a panda, fooling the classifier into predicting a gibbon with high confidence despite the images looking identical to the human eye (adapted from [100]). At the bottom left, physical adversarial patches and paint are applied on stop signs [67], disguising the stop signs as other signs, such as speed limits, according to a model trained to classify signs. At the bottom right, the points of the LiDAR point cloud of a car are altered adversarially while preserving plausibility [163], hiding the car as a different object for a 3D semantic segmentation model.	40
5.1	Schematic comparison of existing panoptic segmentation approaches with the proposed Panoster [84]. Both top-down [345] and bottom-up [46] approaches break down the panoptic task into sub-tasks, such as object detection or center regression. In the diagrams, red boxes highlight critical modules needed to aggregate such sub-tasks. Panoptic fusion aims to resolve conflicts between the intermediate predictions for top-down methods. For bottom-up approaches, the grouping step is responsible for the instance segmentation output, turning intermediate estimates into instance masks. Instead, Panoster simplifies the bottom-up paradigm in an end-to-end fashion. Panoster eliminates the need to resolve conflicts or take critical grouping steps since the instance IDs are delivered directly from the network.	45

5.2	Representation of data distributions. In this simplified example, training data is particularly limited, including only one car model and a few colors (original). Augmented samples add more colors, flip and rotate the inputs (augmented), but rare colors and unseen models are out-of-distribution (OOD). Unconstrained adversarial examples (e.g., those of prior works in the point cloud domain [177, 343]) introduce highly unrealistic samples that do not represent existing vehicles. The proposed technique (highlighted with a rectangle) generates plausible adversarial examples that, when used as augmentation, expand the training data, thus increasing its diversity. By constraining the perturbations with domain knowledge, the generated samples may resemble those that can be found in the wild. Thanks to adversarial augmentation, the system can detect more instances and models, including those that could not be previously detected (detectable w/ adv.aug.). Nevertheless, this technique does not allow the system to detect all unseen models, so some may remain out-of-distribution (still OOD).	70
5.3	Examples of unknown objects (right) given a standard set of known categories learned during training (left). Depending on the training data and the evaluation settings (e.g., the real world), out-of-distribution data can comprise a wide range of objects and classes. With segmentation tasks, when not properly taken into account, unknown objects are systematically and wrongly associated with a class from the pool of known ones. . . .	84
5.4	Representation of the issues of self-supervised monocular depth models with dynamic objects. The distance of leading traffic tends to be overestimated since it does not change position across training sequences, such as the far background when driving at the same speed. Instead, the depth of incoming traffic is often underestimated because such vehicles move significantly more than their surroundings across training sequences. These problems are due to the assumption of the moving camera in a static world, which is violated systematically by any moving object in the scene, e.g., traffic.	102
5.5	Issues of self-supervised monocular depth estimation models at nighttime. General darkness, sensor noise, and light flares caused by vehicle lights and street lamps reduce visibility and prevent establishing correct pixel correspondences across the training sequences. These issues are systematic and detrimental to the learning process, causing severe artifacts in output.	115
5.6	Representation of LiDAR measurements in different weather conditions. Problems occur with adverse weather since the LiDAR rays are disturbed (i.e., absorbed or scattered) by raindrops, wet surfaces, water drops inside fog, or snowflakes. Learning depth from such a signal causes the propagation of artifacts into depth estimates, such as holes in the ground in the presence of rain [86]. The absorption of the rays causes sparser point clouds that reduce the efficacy of LiDAR in these scenarios. Absorbed rays result in erroneous measurements, which are detrimental to downstream tasks. These issues do not occur at night with clear or cloudy weather. Instead, radar is not affected by adverse weather thanks to the longer wavelength of its signal, yet inherently too sparse and noisy to learn depth solely from it [83].	116
7.1	Representation of the end-to-end autonomous driving software stack. Compared to the standard stack shown in Figure 2.5, in this end-to-end setup, the perception, localization, and planning components are merged without needing an HD map. So, sensory inputs are processed by a unified perception and planning module, which implicitly performs scene understanding and directly outputs trajectories for the ego vehicle to follow, simplifying the stack.	138

