

Introspective Methods for Learning-enabled Robotic Perception and Planning

Jianxiang Feng

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: apl. Prof. Dr. Georg Groh

Prüfende der Dissertation:

1. Prof. Dr. Rudolph Triebel
2. Prof. Dr. Angela Schöllig
3. Prof. Dr. Fabio Ramos

Die Dissertation wurde am 21.12.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 28.11.2024 angenommen.

List of Acronyms

| | |
|--------|--|
| AI | Artificial Intelligence |
| AL | Active Learning |
| BNNs | Bayesian Neural Networks |
| CRFs | Conditional Random Field |
| CRSB | Conditional Resampled Base Distribution |
| DL | Deep Learning |
| DNNs | Deep Neural Networks |
| ELBO | Evidence Lower Bound |
| GNNs | Graph Neural Networks |
| GRACE | Graph Assembly Processing Networks |
| IB | Information Bottleneck |
| ID | In-Distribution |
| KLD | Kull-Leibler Divergence |
| LA | Laplace Approximation |
| LARS | Learned Accept/Reject Sampling |
| LBP | Loopy Belief Propagation |
| LL | Log-Likelihoods |
| MCD | Monte-Carlo Dropout |
| MCMC | Markov Chain Monte Carlo |
| MDP | Markov Decision Processes |
| ML | Machine Learning |
| MLE | Maximum Likelihood Estimation |
| MOG | Mixture of Gaussians |
| NFs | Normalizing Flows |
| OOD | Out-of-Distribution |
| POMDPs | Partially Observed Markov Decision Processes |
| RASP | Robotic Assemble Sequence Planning |
| RL | Reinforcement Learning |
| RSB | Resampled Base Distribution |
| SLAM | Simultaneous Localization And Mapping |
| VI | Variational Inference |

Abstract

Introspection refers to shaping the self-awareness of an agent's internal state. This capability can be essential for building trustworthy and cognitive open-world robotic autonomy. As core components within an autonomy stack, striving for enhanced generalizability and efficiency, learning-enabled perception and planning are gaining wider adoption to contend with the complex real world. However, problems of the data-driven learning paradigm, e.g. overconfident predictions and vulnerability against Out-of-Distribution (OOD) inputs, raise serious safety concerns for these approaches applied in robotics. These problems motivate specific research challenges to be addressed, spanning from reliable uncertainty estimation and effective OOD detection to learning actively. In this thesis, we attempt to address these challenges by developing learning-based methods with improved introspective capabilities for application in robotic perception and assembly sequence planning.

To this end, we first develop a method based on Bayesian Deep Learning and Probabilistic Graphical Models for reliable uncertainty estimation. This method can not only assist uncertainty-based adaptive object classification but also incorporate object co-occurrence in the scene, facilitating semantic reasoning capabilities. Secondly, towards open-world robot deployment, we introduce an efficient and flexible OOD detection method with flow-based deep generative models, where we propose to utilize an expressive base distribution in the flow to mitigate the fundamental topological constraint. This leads to a performant open-set object detector that is compatible with diverse existing architectures. We further study a similar idea for feasibility learning of an assembly in the task of Robotic Assemble Sequence Planning (RASP), for which we propose a holistic data-driven graphical approach based on Graph Neural Networks (GNNs). This work provides a promising direction to address the challenge of spatial embodiment. Lastly, to pave the way to an active and incremental learning-enabled robot, we devise an active learning pipeline for sim-to-real object detection based on uncertainty estimates from Bayesian Neural Networks (BNNs), in which we tackled the issue of label distribution shift under such conditions with a simple yet effective sampling strategy.

Besides comprehensive evaluation in simulation and on self-collected and benchmark data sets, we further conduct real-robot experiments with these methods on an assistance robot and an aerial manipulator, demonstrating their practical applicability. We aim to develop methods to equip a data-driven, learning-enabled robot with introspective capabilities for greater reliability, adaptivity, and autonomy.

Zusammenfassung

Unter Introspektion versteht man die Gestaltung der Selbstwahrnehmung des inneren Zustands eines Agenten. Diese Fähigkeit kann für den Aufbau einer vertrauenswürdigen und kognitiven Roboterautonomie in der offenen Welt von entscheidender Bedeutung sein. Als Kernkomponenten innerhalb eines Autonomiestapels streben sie nach verbesserter Generalisierbarkeit und Effizienz, lerngestützte Wahrnehmung und Planung gewinnen zunehmend an Bedeutung mit der komplexen realen Welt. Probleme des datengetriebenen Lernparadigmas, z.B. Übermäßige Vorhersagen und Anfälligkeit gegenüber Out-of-Distribution (OOD)-Eingaben. Es bestehen ernsthafte Sicherheitsbedenken hinsichtlich dieser in der Robotik angewandten Ansätze. Diese Probleme motivieren spezifische Forschungsherausforderungen, die angegangen werden müssen, angefangen bei der zuverlässigen Unsicherheitsschätzung, effektive OOD-Erkennung für aktives Lernen. In dieser Arbeit möchten wir uns mit diesen befassen Herausforderungen durch die Entwicklung lernbasierter Methoden mit verbesserten introspektiven Fähigkeiten für die Anwendung in der Roboterwahrnehmung und Montagesequenzplanung.

Zu diesem Zweck entwickeln wir zunächst eine Methode, die auf Bayesian Deep Learning und Probabilistische Grafische Modelle zur zuverlässigen Unsicherheitsschätzung. Diese Methode kann nicht nur helfen Unsicherheitsbasierte adaptive Objektklassifizierung, sondern auch die Einbeziehung des gleichzeitigen Vorkommens von Objekten in der Szene und erleichtert so die Fähigkeit zum semantischen Denken. Zweitens in Richtung Open-World-Roboter Bereitstellung führen wir eine effiziente und flexible OOD-Erkennungsmethode mit Flow-Based deep generative Modelle, bei denen wir vorschlagen, eine ausdrucksstarke Basisverteilung im Fluss zu nutzen um die grundlegende topologische Einschränkung abzuschwächen. Dies führt zu einem performanten Open-Set-Objekt Detektor, der mit verschiedenen bestehenden Architekturen kompatibel ist. Wir untersuchen weiter etwas Ähnliches Idee zum Machbarkeitslernen einer Baugruppe im Rahmen der Roboter-Montagesequenzplanung (RASP), für das wir einen ganzheitlichen datengesteuerten grafischen Ansatz basierend auf Graph Neural vorgeschlagen Netzwerke (GNNs). Diese Arbeit bietet eine vielversprechende Richtung zur Bewältigung der räumlichen Herausforderung Verkörperung. Um schließlich den Weg zu einem aktiven und inkrementell lernfähigen Roboter zu ebnen, wir entwickeln eine aktive Lernpipeline für die Sim-to-Real-Objekterkennung basierend auf Unsicherheit Schätzungen von Bayesian Neural Networks (BNNs), in denen wir uns mit dem Thema Label befasst haben Verteilungsunterschiede unter solchen Bedingungen mit einer einfachen, aber effektiven Probenahmestrategie.

Neben umfassender Auswertung in Simulation, an selbst erhobenen und Benchmark-Datensätzen, Darüber hinaus führen wir mit diesen Methoden Realroboter-Experimente an einem Assistenzroboter durch ein Luftmanipulator, der ihre praktische Anwendbarkeit demonstriert. Insgesamt wollen wir uns weiterentwickeln Methoden, um einen datengesteuerten,

lernfähigen Roboter mit introspektiven Fähigkeiten für mehr auszustatten Zuverlässigkeit, Anpassungsfähigkeit und Autonomie.

Acknowledgment

When my fingers hit the keyboard, close to concluding this thesis, my thoughts flash back to the kind support from many others for whom I am deeply grateful throughout this adventure.

My foremost gratitude goes to my advisor, Prof. Rudolph Triebel. His thinking was inspiring, his enthusiasm contagious, his trust encouraging, and his support invaluable. He established a phenomenal research environment with outstanding colleagues and incredible freedom at DLR, enabling me to pursue the research I like. When confronted with the unexpected dilemma on the cusp of finishing, I could keep moving forward, backed up by his invigorating words and deeds. Working with him over the past years was a pleasure and honor. I would like to extend my heartfelt gratitude to my colleague and office mate, Jongseok Lee. This thesis is hard to reach the current stage without his support. The charming dawn after paper submission and numerous intelligently challenging discussions with him shaped my first and fondest memory in research. More importantly, I have learned invaluable lessons from him, which not only re-crystallized my understanding of robotics and machine learning research but also helped develop my foundational research skills. The special thanks also flow to Maximilian Durner, one of my master thesis supervisors and our research group leader. He mentored me closely by enlightening me with his technical expertise in robotics, guiding me on scientific writing, and facilitating inter-department collaboration with other colleagues, which tremendously hones my research abilities. Furthermore, I am greatly indebted to Prof. Stephan Günemann and Simon Geisler within the MuDS program. I appreciate their contributions to this thesis and learned a great deal from the discussion with them. Besides, sincere thanks go to my committee members: Prof. Angela Schoellig, Prof. Fabio Ramos, and the chair, Prof. Georg Groh.

Over the past years, I have also worked on perception in the EDAN (an assistive robot) team at DLR, led by Jörn Vogel. I would like to express my warmest thanks to the whole team, including Annette Hagenruber, Gabriel Quere, Samuel Bustamante, and Maged Iskandar. The successful demo at AUTOMATICA 2022 and the 1st place in the Cyathlon Competition 2023 with our pilot Matthias and new members Sebastian Jung and Elle Miller are strikingly unforgettable to me. These experiences forged my understanding of the "bittersweet" recipe of how to make a complex robot system work smoothly in and outside the lab. This recipe might potentially explain why Jörn can make such a tasty cake.

I have been fortunate to join the SAM (an aerial manipulator robot) team at DLR, led by Dr. Konstantin Kondak. Along with Jongseok, Dr. Ribin Balachandran, Dr. Marco De Stefano, Hrishik Mishra, Manuel Schnaus, and Nari Song, we worked for the KUKA Innovation Award 2023. Though the time was relatively short, I really enjoyed the teamwork and gained valuable experience with another complex robot system. I owe a huge thank you to them!

I was so lucky to be surrounded by a group of brilliant people to whom I own my sincere thanks: Dr. Ulrich Hillenbrand, thank you for being my official mentor and the first supervisor at DLR. I enjoyed each conversation with you, both professionally and personally. Dr. Zoltan-Csaba Marton, thank you for the kind help at both the beginning and the end of this journey; Mohit Kumar and Matan Atad, thank you for being my students and supporting my research; Dr. Ismael Valentin Rodriguez Brena, thank you for the great collaboration; Finally, a big thank you to all the colleagues and friends inside and outside DLR who helped me directly and indirectly, including but not limited to Dr. Korbinian Nottensteiner, Matthias Humt, Yizhe Wu, Janis Postel, Martin Sundermeyer, Maximilian Denningen, Dominik Winkenbauer, Wout Boerdijk, Dr. Klaus H. Strobl, Dr. Riccardo Giubilato, Maximilian Ulmer, Xuming Meng, Robert Schuller, Ana Elvira Huevo Martin, Xuwei Wu, Harsimran Singh, Dr. Daniel Leidner, Marcus Gerhard Müller, Mallikarjuna Vayugundla, Ria Vijayan, Florian Schmidt and so on.

Last but not least, many thanks to all my friends and family outside of work! All this would not have been possible without their enduring support of them, both emotionally and physically.

Munich, Winter 2024

Jianxiang Feng

Contents

List of Acronyms

| | |
|--|-----------|
| 1. Introduction | 1 |
| 1.1. Motivation | 1 |
| 1.2. Contributions | 5 |
| 1.3. Thesis Structure | 8 |
| 1.4. Publication Note | 10 |
| 2. Fundamentals of Robotic Introspection | 12 |
| 2.1. Introduction | 12 |
| 2.2. Literature Study | 14 |
| 2.3. Preliminaries | 17 |
| 2.3.1. Bayesian Neural Networks | 17 |
| 2.3.2. Normalizing Flows | 19 |
| 2.3.3. Graph Neural Networks | 21 |
| 3. Introspective Methods for Robotic Perception | 23 |
| 3.1. Uncertainty-based Adaptive Classification with Scene Contexts | 23 |
| 3.1.1. Related Work | 24 |
| 3.1.2. Methodology | 24 |
| 3.1.3. Summary of Results | 27 |
| 3.2. Flow-based Open-Set Object Detection | 27 |
| 3.2.1. Related Work | 27 |
| 3.2.2. Methodology | 28 |
| 3.2.3. Summary of Results | 30 |
| 3.3. Active Learning for Sim-to-Real Object Detection | 31 |
| 3.3.1. Related Work | 31 |
| 3.3.2. Methodology | 33 |
| 3.3.3. Summary of Results | 36 |
| 4. Introspective Methods for Robotic Assembly Sequence Planning | 37 |
| 4.1. Assembly Sequences Prediction via Graph Representations | 38 |
| 4.1.1. Related Work | 38 |

| | | |
|-----------|---|-----------|
| 4.1.2. | Methodology | 39 |
| 4.1.3. | Summary of Results | 42 |
| 4.2. | Density-based Feasibility Learning | 43 |
| 4.2.1. | Related Work | 44 |
| 4.2.2. | Methodology | 44 |
| 4.2.3. | Summary of Results | 45 |
| 5. | Applications and Discussions | 46 |
| 5.1. | Applications | 46 |
| 5.1.1. | Introduction of Robotic Systems | 46 |
| 5.1.2. | Saving Annotation Efforts for Real Robot Perception | 51 |
| 5.1.3. | Detecting Out-of-Distribution Objects for Inspection and Maintenance | 54 |
| 5.2. | Conclusions | 55 |
| 5.3. | Limitations | 56 |
| 5.4. | Future Directions | 57 |
| 6. | Summary of Publications | 59 |
| 6.1. | Publication 1 : Uncertainty-based Adaptive Classification with Scene Contexts | 60 |
| 6.2. | Publication 2 : Flow-based Open-Set Object Detection | 61 |
| 6.3. | Publication 3 : Bayesian Active Learning for Sim-to-Real Object Detection . | 62 |
| 6.4. | Publication 4 : Predicting Assembly Sequences via Graph Representations . | 63 |
| 6.5. | Publication 5 (Pre-Print): Introspective Robotic Assembly via Feasibility Learning | 64 |
| 6.6. | Publication Licenses | 65 |
| | Bibliography | 74 |
| A. | Full Text of Publications | 94 |
| A.1. | Publication 1 | 94 |
| A.2. | Publication 2 | 111 |
| A.3. | Publication 3 | 124 |
| A.4. | Publication 4 | 133 |
| A.5. | Publication 5 | 142 |

1. Introduction

1.1. Motivation

“ If knowledge is power, knowing what you don't know is wisdom.

”

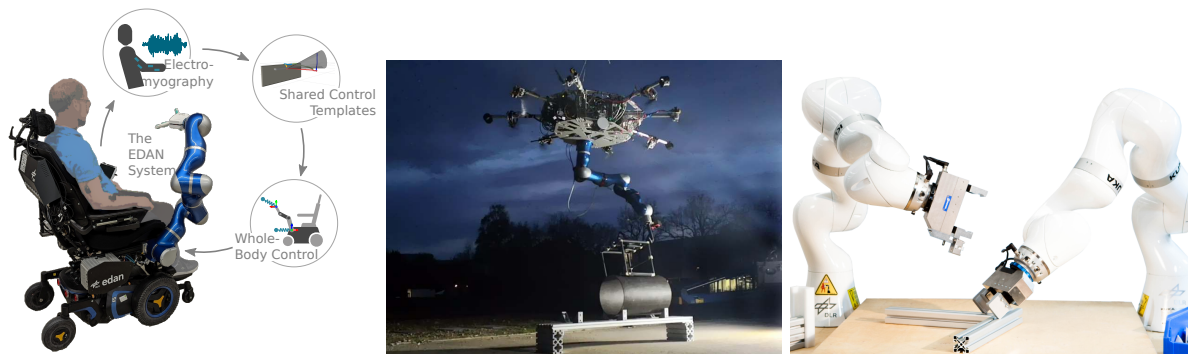
Adam Grant

Enabled by the impressive performance and generalization capability brought by the advances in Deep Learning (DL) over decades, robots nowadays are able to function more intelligently in diversified scenarios. Though, envisioned to perform complex tasks, a robot needs to *perceive*, *plan*, and execute *actions* based on incomplete and uncertain information from a continuously evolving and dynamic environment. Such imperfect conditions can easily cause mistakes, that might negate the success of the robot's mission or potentially jeopardize human lives in safety-critical applications. For example, in case of the assistance robot for elderly people caregiving in Figure 1a, a mis-behavior of the robot arm might cause hazardous injuries to the users or costly damages to the robot. In this respect, the current DL-based approaches primarily strive for prediction accuracy boosting with an unlikely achievable goal of being correct all the time while being lax about another important capability – being wisdom and know what the model doesn't know, to embrace and effectively handle the failure cases. This inattentive research gap renders DL into an exacerbator instead of an ameliorator for such critical issue of reliability and robustness in robotics.

It is challenging to address the safety concern by endowing learning-enabled robots with such capability due to numerous fundamental limitations of DL including the implicit *closed-set assumption* [Sin+22], *overconfidence* for false predictions [Gaw+23], the *lack of adaptive capability* [Sün+18] and so on. The closed-set assumption (also known as i.i.d. assumption, namely independent and identically distributed) requires the training and test data distribution to be identical, which is routinely violated in an open-set real-world environment where our robots are envisaged to be deployed. Moreover, being overconfident about false predictions is unfavorable or even detrimental from a system-level perspective because this kind of misleading output signals would adversely affect other sub-modules unpredictably within a large system. Last but not least, the inability to actively and continually learn from the data largely limits the application scenarios, confining the robot to work in a pre-defined and static environment.

To illustrate these with an example of an aerial manipulation robot in Figure 1b, by relying on a learning-based object detector in the dynamic and uncontrollable outdoor environment, the robot might confidently but falsely detect an unknown object as a known one. This erroneous detection can easily damage the manipulator or the object due to the lack of manipulation skills for the unknown object. Further, when requested to complete a mission in a starkly different factory, the robot fails to adapt to the new environment, which might lead to substantial costs of time, money and human engineering.

To further emphasize the salient impact of these challenges in robotics, we provide a more detailed explanation of the research challenges under the taxonomy introduced in [Sün+18].



(a) An assistive robot for helping people with disabilities, for EDAN [Vog+20b]. (b) An aerial manipulation robot for factory inspection and maintenance [Lee+23]. (c) A dual-armed robotic systems for assembly sequence planning [Rod+19].

Figure 1.: Exemplar robotic systems (more details in chapter 5) demand effective and reliable perception and planning functionalities, motivating for the development of introspective methods that can address such requirement and meanwhile is generic so that it can be used for a diverse set of robotic functional modules.

Research Challenges The foremost challenges we aim to address for learning-enabled robotics are in line with those of deep learning in robotic vision proposed by [Sün+18]. To note that we interpret these challenges in a broader context, i.e. learning-based components in a robot autonomy stack, which can be a learning-based perception, planning or even control module. The research challenges consisting of three aspects – *learning*, *reasoning* and *embodiment*, are briefly described and positioned along three conceptually orthogonal axes according to their increasing complexity and dependencies in Fig. 3.

- **Learning Challenges** In this thesis, we primarily attend to the learning challenges, DL-based components in robotics are expected to provide reliable uncertainty estimates for their predictions. This function is highly beneficial for data fusion with other sub-modules and safety-critical failure avoidance.

Further, the learning-based components should be capable of operating in an open-set environment that is full of unknowns, or OOD data, which is hard to avoid during real-world robot deployment. The current techniques of uncertainty estimation can alleviate

this problem but still fall short when compared with dedicated approaches such as density-based methods with Normalizing Flows (NFs). More seriously, the fundamental architectural limitation in NFs [Cor+20] raises significant concerns on its performance and computation overheads.

Moreover, aiming to facilitate a long-term robotic autonomy to alleviate intensive manual efforts when adapting in new environments, these modules need to learn and update incrementally and actively in an efficient and autonomous way. Though there are encouraging advances for this capability in DL research, many of them still fall short on the practicality and performance cross domains such as the gap from simulation to reality. Concretely speaking, training with simulation data helps resolve the data efficiency issue to certain extent while the remaining sim-to-real gap still impedes the generalizability. How to further eliminate the last mile of this gap with Active Learning (AL) remains an open question since there are distributions shifts in data selection process.

- **Embodiment Challenges** Embodiment is the cornerstone that sets robot learning apart from pure ML. The temporal and spatial embodiments leverage the characteristics of a robot as an active agent in the physical world. How to effectively incorporate these aspects separately and jointly into the learning methods is the research challenge yet to be addressed. To put this challenge into the case of Robotic Assemble Sequence Planning (RASP), the assembly robot such as the dual-armed system in Figure 1c needs to understand whether certain assemblies are feasible for it to reach and manipulate without collision. Due to the complex relations between various parts, e.g., the whole parts and its surfaces, and the sparsity of the features, it is challenging to discover the spatial structure in a learnable manner while also considering the feasibility of the robot actions.
- **Reasoning Challenges** Reasoning represents the human-like cognitive capability of a robot. When integrated on a cognitive robot, the learning-based components are anticipated to understand and reason about the semantics or geometry as humans do. Semantics denote semantic regularities appearing around us, e.g. object co-occurrence or situation awareness in specific scenarios and so on. Geometry is a ubiquitous cue for in diverse tasks as robot is an active agent that needs to act in the physical world with various structures. The main barrier to actualize such capabilities resides in the used design paradigm and architecture. For example, to incorporate semantics reasoning into an existing end-to-end trained neural nets is hard as the knowledge of the data has been abstracted and stored in a "black-box". Classical methods such as CRFs excel at specifying such patterns but how to merge them into the 'black-box' is challenging. On the other hand, commonly-used convolutional neural nets can handle geometry in the image data quite well but fail to model structured data such as graph. Graph Neural

Networks (GNNs) also provide a good alternative but receive little attention for the RASP task to the best of our knowledge.

Introspective knowledge [McC99] and the corresponding concepts grounded in robotics like robotic introspection [Fox+06]; [Mor07] and introspective capabilities [Gri+13]; [Tri+16] are promising to provide a remedy but surprisingly inattentive in the algorithmic approach development in robotics. It defines the ability to self-observe, model and intelligently alter internal state without external assistance such as humans. To put this into context, we expect to endow a robot with **introspective capabilities** – being able to provide calibrated confidence, identify unknown data, and consequentially utilize this capability for long-term autonomous adaptation cross different scenarios. In a word, the robot should know what it doesn't know and grow by learning from this process.

Therefore in this thesis, our goal is to investigate introspective methods for learning-enabled robotics. To ground our methods into a robot system and demonstrate the effectiveness, we focus on the perception and planning modules of a robot due to their importance in the system and the urgent safety demand raised by its commonly encountered complex interactions with the surrounding. Most specifically, to comply with customized tasks of the robot systems used in the work visualized in figure 1, we narrow down our application scope to certain tasks of these two modules listed in figure 2. For perception, we concentrate on object classification, object detection and their efficient adaptation from simulation data to the reality. For planning, we attend to a variant of task planning, namely the robotic assembly sequence planning and the feasibility learning in this task. Through a long stretch of development from methodology to application, we hope to pave the way toward a reliable and high-performing robotic system in the near future.

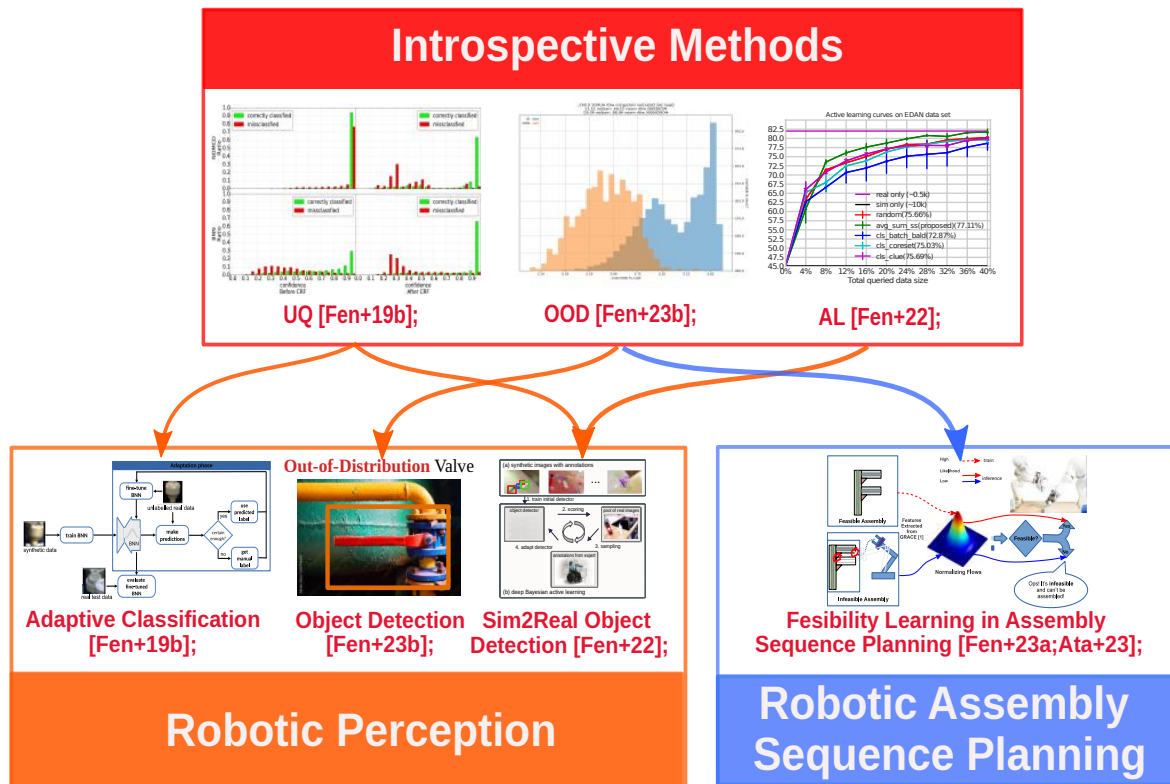


Figure 2.: Thesis Overview This figure outlines the thesis structure and the major key components introduced in this thesis. We develop introspective methods including uncertainty quantification (UQ), Out-of-Distribution Detection (OOD) and Active Learning (AL), which are further adapted and applied to a diverse set of tasks in robotic perception and robotic assembly sequence planning.

1.2. Contributions

With the introspective methods developed for robotic perception and planning visualized in Figure 2, we aim to contribute to the challenges introduced in the last section. The major focus has been placed on the axis of learning while we partially cover the reasoning and embodiment challenges, which will be described in more detail in this section.

The corresponding publications are listed in section 1.4 and detailed in the following chapters. To make the contributions more comprehensible for the readers, we position them in the coordinate of the aforementioned research challenges in Figure 3, denoted by red diamonds.

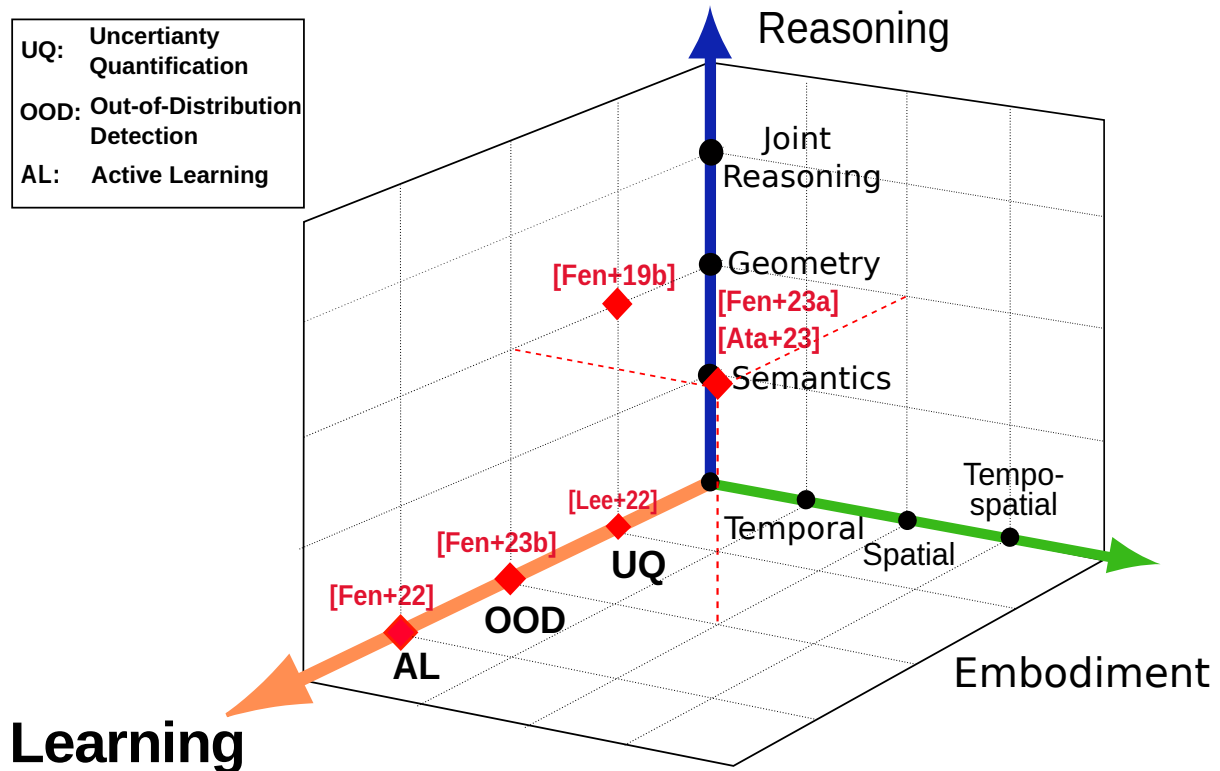


Figure 3.: Challenges Addressed in This Thesis The goal of this thesis is to develop introspective methods for the challenges of deep learning for robotics [Sün+18] primarily along the axis of learning while partially covering those in embodiment and reasoning. Red diamonds denote the contributions made in this thesis.

In the following, we start summarizing the contributions made in this thesis along the axis of learning:

- **Introspective Methods**

- For **Uncertainty Quantification (UQ)**, we proposed to leverage Bayesian Neural Networks (BNNs) for more reliable uncertainty estimates and fuse these uncertainty estimates with Conditional Random Field (CRFs) in **Publication 1**. This method provides not only more accurate uncertainty estimation but facilitate effective adaptation across different scenarios by incorporating the semantic regularities such as the object co-occurrence in the scene. This work provide a promising way to address the challenges of uncertainty quantification and semantics reasoning.

- For **OOD detection (OOD)**, we proposed to enhance density-based OOD detection by mitigating the fundamental constraint in the flow-based deep generative model in **Publication 2**. This is achieved with an expressive base distribution and information-theoretic objective. This method can yield better performance than the other baselines

including the one based on uncertainty estimation and largely mitigate the safety concern with OOD data.

- For **Active Learning (AL)**, we proposed an uncertainty-based **active learning** pipeline built upon BNNs in **Publication 3**. In contrast to common active learning, our scenario is more complex due to the distribution shift between the initial data (in simulation) and target test data (in reality). In this work, we tackle the label distribution shift problem with a simple yet effective sampling strategy. We believe that this work can attract more attention on achieving data efficiency with AL for cases with a hard-to-resolve domain gap.

- **Robotic Perception**

- In **Publication 1**, we devise an adaptive learning framework for object classification based on the improved uncertainty estimates from BNNs in a semi-supervised manner. Furthermore, we applied the proposed combination of BNNs and CRFs for efficient adaptation of an industrial object classification task.

- In **Publication 2**, based on the proposed topology-matching NFs, we develop a OOD-aware object detection that enjoy the merits of run-time efficiency and a wide compatibility with numerous existing detector architectures. More noteworthy, we showcased the applicability of the proposed method by deploying it on an embedding system (NVIDIA Jetson Orin) of an aerial manipulator robot system.

- In **Publication 3**, we introduce an active learning pipeline for an object detector deployed on an assistive robot. With the proposed sampling strategy, we can bridge the last mile in the sim-to-real adaptation with significantly less data annotation efforts. To verify the applicability, we conduct a real-robot grasping experiment based on this pipeline.

- **Robotic Assembly Sequence Planning**

To note that, we interpret introspective capabilities in this case as the ability to know whether a certain assembly is feasible for the assembly robot or not.

- F we proposed a holistic graphical method for Robotic Assemble Sequence Planning (RASP) based on Graph Neural Networks (GNNs) in **Publication 4**. We further endow the model with introspection on the assembly feasibility based on flow-based OOD detection.

- In **Publication 4**, we first propose a graphical method for efficient and feasible assembly sequence prediction based on Graph Neural Networks (GNNs). We further endow the model with **introspective capability** by training with both feasible and infeasible assemblies. The way to handle spatially structured data with graphs suggests a promising way for addressing the spatial reasoning challenge.

- In **Publication 5**, we handle the case without infeasible assemblies by formulating **feasibility learning in RASP as a OOD detection** problem. In this way, we can tackle it with the method based on NFs introduced in **Publication 2**. When coupling with the method proposed in **Publication 4**, to leverage OOD detection for feasibility learning based on graphical representations yields encouraging results, which are inspiring for addressing the spatial embodiment challenge

Last but not least, beyond the experimental evaluation on public benchmarking and self-collected datasets, we further demonstrate the applicability of the proposed methods, i.e., the active learning pipeline in **Publication 3** and OOD detection method in **Publication 2** with real-robot experiments on two robot systems, namely an assistance robot EDAN [Vog+20b] and an aerial manipulation robot SAM [Sar+19b]; [Lee+23].

1.3. Thesis Structure

This publication-based thesis presents a collection of introspective methods in order to address the challenges of learning-based methods in robotics. By tackling these challenges, especially in the learning aspect, these solutions enable a robot to establish introspection in perception and planning, hence being robust and reliable when operating in a complex environment. In the following chapters, the fundamentals of the proposed introspective methods are first outlined and described, method details and their related work are introduced and articulated, and detailed information of each publication is attached and explained.

Chapter 2 describes a study of relevant work in robotics literature and two machine learning models (Bayesian Neural Networks and Normalizing Flows), which serve as the core building blocks for the proposed methodology in the later chapters.

Chapter 3 presents three ideas proposed for introspective methods applied to robotic perception. Each of them targets one challenge along the axis of learning in Figure 3. Before going to the algorithmic details, a study of their related works is presented to highlight their technical contributions.

Chapter 4 focuses on the introspective methods for robotic assembly sequence planning with details of the proposed ideas and their related work in the literature.

Chapter 5 first demonstrates the applications of the proposed methods by deploying them on corresponding robots in both the real world and simulation. Then, we conclude with a synopsis of this work and a discussion about the limitations of the proposed methods followed by potential future directions that might be enlightening for the community.

Chapter 6 provides concise information about each publication discussed in the method chapter and outlines the contribution of the author of this thesis. A full version of each publication is attached in appendix A.

1.4. Publication Note

First authorship (*: equal contribution.):

Publication 1 [Fen+19b]

Jianxiang Feng*, Maximilian Durner*, Zoltán-Csaba Márton, Bálint-Benczédi Ferenc and Rudolph Triebel. "Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks". In: *International Symposium on Robotics Research (ISRR)*. 2019.

Publication 2 [Fen+23b]

Jianxiang Feng, Jongseok Lee, Simon Geisler, Stephan Günnemann and Rudolph Triebel. "Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning". In: *7th Annual Conference on Robot Learning (CoRL)*. 2023.

Publication 3 [Fen+22]

Jianxiang Feng, Jongseok Lee, Maximilian Durner and Rudolph Triebel. "Bayesian Active Learning for Sim-to-Real Robotic Perception". In: *the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022.

Publication 4 [Ata+23]

Matan Atad*, **Jianxiang Feng***, Ismael Rodríguez, Maximilian Durner and Rudolph Triebel. "Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning". In: *the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023.

Publication 5 [Fen+23a] (Pre-print)

Jianxiang Feng*, Matan Atad*, Ismael Rodríguez, Maximilian Durner, Stephan Günnemann and Rudolph Triebel. "Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly". In: *Workshop on Robotics and AI: The Future of Industrial Assembly Tasks , Robotics: Science and Systems (RSS)*. 2023.

Co-authorship:

[Lee+23] Jongseok Lee, Ribin Balachandran, Konstantin Kondak, Andre Coelho, Marco De Stefano, Matthias Humt, **Jianxiang Feng**, Tamim Asfour and Rudolph Triebel. "Virtual Reality via Object Poses and Active Learning: Realizing Telepresence Robots with Aerial Manipulation Capabilities". In: *Field Robotics*. 2023.

[JGH18] Gawlikowski, Jakob, Cedrique Rovile Njietcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, **Jianxiang Feng**, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler and Xiao Xiang Zhu. "A survey of uncertainty in deep neural networks." In: *Artificial Intelligence Review*. 2023.

[Lee+22] Jongseok Lee, **Jianxiang Feng**, Matthias Humt, Marcus G. Müller and Rudolph Triebel. Trust Your Robots! Predictive Uncertainty Estimation of Neural Networks with Sparse Gaussian Processes. In: *5th Conference on Robot Learning (CoRL)*. 2021.

[Lee+20] Jongseok Lee, Matthias Humt, **Jianxiang Feng** and Rudolph Triebel. "Estimating Model Uncertainty of Neural Networks in Sparse Information Form". In: *the Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2020.

2. Fundamentals of Robotic Introspection

Introspective methods offer a promising direction for addressing safety concerns when applying DL in robotics. In particular, the introspective capabilities can be exploited to resolve the research challenges motivated by the characteristics of data-driven learning paradigm in robotics [Sün+18]. In this chapter, we will dig into the fundamentals of introspective methods in robotics, spanning from the literature to the theory background of probabilistic machine learning models upon which our proposed methods are established, i.e. Bayesian Neural Networks (BNNs) and Normalizing Flows (NFs). With the preliminaries introduced in this chapter, the readers are well-prepared to understand the technical details of the proposed techniques presented in Chapter 3 and Chapter 4.

2.1. Introduction

“ To ascribe certain beliefs, knowledge, free will, intentions, consciousness, abilities or wants to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair or improve it. It is perhaps never logically required even for humans, but expressing reasonably briefly what is actually known about the state of a machine in a particular situation may require ascribing mental qualities or qualities isomorphic to them. ”

John McCarthy, 1979

Dating back to the time of logical Artificial Intelligence (AI), introspection has been investigated as an indispensable functionality for robots to reach human-level intelligence [McC99]. Unlike consciousness or awareness with explicitly represented beliefs, introspection is described as a process or capability to shape consciousness with a chain of *mental actions* that the

robot decides to take, e.g. observation of its previous consciousness. This way of creating consciousness is also called *self-consciousness*.

Thinking about an example of human introspection. Suppose I ask you whether your colleague has been on holiday recently and your first answer is "No idea". Then I ask you to *think harder*, you might recall that his place in the office is empty for a few days when you pass by or you *really don't know* and want to ask others for help. This is how "introspection" helps to get the final answer. Another simple example can be an electronic alarm clock getting power after being without power and blinking its display to signal that *it doesn't know* the time. It can be said that both the human and a simple system like a clock possess the introspective ability to be aware of their internal states and express it in their own way. Likewise, robots also need an analogous capability if they are to decide correctly whether to think more or solicit help externally. This capability to infer *non-knowledge* and do non-monotonic reasoning is deemed essential for a trustworthy intelligent robot.

In robotics, robotic introspection was first introduced to tackle critical problems for reliable robot operation and deployment, e.g. behavior modeling [Fox+06], operation failure identification and recovery [Mor07]. Later this concept gained a deluge of attention for robot perception as perception is often the weak link in robotic systems [Daf+16]; [Gri+16]; [HK17], requesting more effective failure detection to avoid potential hazardous outcomes. Moreover, the usefulness of introspection has been extended beyond the aforementioned tasks, to another under-investigated direction about how to exploit the introspective knowledge for long-term evolution such as active learning [Tri+16]; [Fen+22]. This is seamlessly in line with the description above, having introspective knowledge can facilitate more efficient learning capabilities, e.g., knowing the right timing to ask for help. In such a scenario, the data that is considered most informative needs to be selected for an oracle to annotate in order to avoid acquiring unnecessary data, hence saving labor and time from the expensive data labeling process. These works provide strong shreds of evidence for the benefits of equipping a robot with introspection.

Paralleled to the development of robotic introspection, probabilistic robotics [TBF05] is a vibrant field that leverages probabilistic approaches for robot state estimation. Meanwhile, it provides a technically appealing way to approach introspective capabilities. Probabilistic representations are able to yield beneficial properties for learning-enabled robots such as reliable uncertainty estimation, and elegant ways to handle incomplete data [Fox98]; [SV10]; [Kae+10]. For this reason, a substantial amount of algorithms from probabilistic robotics have been exploited for the development of robotic introspection. The most noteworthy example is using probability as an uncertainty measure for safe and robust robotic introspection [TGP13]. Such probability can be obtained from inference under a probabilistic framework in a natural and principled way.

With advances in Deep Learning (DL) over the last decade, the data-driven learning paradigm has played a major role in developing core modules of a robotic system such as perception and planning [Sün+18]. This paradigm can provide a remarkable generalization ability, significantly outperforming the non-DL-based counterpart. However, due to the notorious problem of overconfident predictions [Guo+17] and the closed-set assumption [Sin+22]; [Ova+19], the safety risk is hard to compensate when we deploy such methods on the robot [Sün+18]. On this account, a corpus of them focuses on how to attain reliable uncertainty estimation from DNNs [Gaw+23].

Thereby, in this thesis, we attempt to investigate DL-based introspective methods for robotics and show that they suggest a promising way of handling these issues. To this end, we exploit the models from probabilistic machine learning [Mur23] because this family is able to effectively combine nice properties of traditional Bayesian probabilistic inference with the marvelous learning capability of deep learning. To concretize, our work is built upon a bedrock of two popular models in this family, i.e. Bayesian Neural Networks (BNNs) and Normalizing Flows (NFs). BNNs yield the predictive output distribution via Bayesian inference, overcoming the flaw of the single prediction from deep neural nets learned by Maximum Likelihood Estimation (MLE). With this, we can attain more reliable uncertainty estimation and multiple hypotheses of the predictive model in a principled manner. On the other hand, as a deep generative model, NFs excel at density estimation and calculating the exact likelihoods for high-dimensional data based on the Change-of-Variables formula. Therefore, it can be utilized to model complex data distribution and detect data examples that are from the Out-of-Distribution.

2.2. Literature Study

In this section, we provide a literature review of introspective methods in robotics from a general point of view and defer the more detailed related work analysis of the specific proposed methods to chapters later. To motivate the usage of BNNs and NFs and ease the understanding of technical details of the proposed methods in the next chapters, we also briefly introduce their related work from the perspective of probabilistic machine learning before presenting the theoretical fundamentals of them.

Introspective Methods in Robotics The concept of introspection for robots was introduced as an important mental capability for achieving human-cognition [McC99]. Instead of abstracting the robot with a computing machine, robotic researchers [Fox+06]; [Mor07] brought introspection into robotics and implemented it as an additional module to model and understand its own behaviors, hence enabling self-monitoring for the whole system. Rather than serving as a separate component, introspection has been exploited for developing methods for core modules in a robotic system such as decision-making [Gri+13]; [Zho+20]. As

perception is normally the weak link in a robotic system, devising introspective methods for perception has gained a deluge of popularity [Daf+16]; [Gri+16]; [GTP16]; [HK17]; [Gur+18]. Though with the same aim, they achieved it with diversified techniques including model uncertainty estimation [Gri+16]; [HK17], learning from past experience [HK17]; [Gur+18] and so on. Moreover, some researchers pioneer on how to leverage introspection for more autonomous and intelligent robot learning such as active learning [Tri+16]. The impressive learning capability from DL and its safety concerns have sparked another hype for the introspective methods in robotics, covering perception [KG17]; [Sün+18]; [Mil+18]; [HLT20], navigation [Shi+20], planning [LEH19]; [LSS20] and control [Kuo+21]; [Hun+21].

Probabilistic robotics gained its attention first in robot state estimation ranging from perception and localization [Fox98]; [Fox+00]; [Thr+01]; [TBF05] to Simultaneous Localization And Mapping (SLAM) frameworks [Kae+10]; [DK+17]. When it comes to functions beyond state estimation, e.g. planning and control, the estimation problems can be formulated in a Bayesian sequential learning setting. One famous instantiation is the sequential decision-making frameworks such as Partially Observed Markov Decision Processes (POMDPs) [Ros+08]; [SV10] which assume a probabilistic treatment of the underlying planning problems. Likewise, in the domain of Reinforcement Learning (RL), a plethora of algorithms are backed up by stability guarantees for safe interactions in the real world [RBK18]; [BSK16]. In the era of Deep Learning (DL), due to the notorious problems of black-box mechanism and overconfident prediction, the majority of researchers are seeking methods that can provide reliable uncertainty estimates and exploit such information for downstream tasks. Notably, [RR17] proposed to perform novelty detection using auto-encoders, where the model can provide confidence about how much one can trust the network's predictions. [Per+20] developed a SO(3) representation and uncertainty estimation framework for the problem of rotational learning with uncertainty. [LEH19]; [Kah+17]; [Stu+11] demonstrated uncertainty-aware, real-world application of RL algorithms for robotics, while [TI18]; [FI18]; [Kuo+21] proposed to leverage spatial information with uncertainty estimated via a popular uncertainty estimation in ML, Monte-Carlo Dropout (MCD).

Bayesian Neural Networks Bayesian Neural Networks (BNNs) [Den+87]; [TLS89]; [BW91] are known to marry the best of both Bayesian statistics and modern DL. Thereby, they are promising to deliver a model that combines the scalability, expressiveness, and predictive performance of DNNs with principle probabilistic inference in Bayesian learning. One appealing merit from this may be the ability to yield more reliable uncertainty estimation thanks to the marginalization over the posterior distribution over the model [WI20b]. Moreover, BNNs also open up the possibility to bridge the powerful Bayesian toolboxes with DL. Notable examples include Bayesian model selection [Mac92a]; [Sat01]; [CB01]; [GYD19], model compression [LUW17]; [FUW17]; [Ach+18], active learning [Mac92c]; [GIG17]; [KVG19], continual learning [Ngu+18]; [Ebr+20]; [FG19]; [Li+20a], theoretic advances in Bayesian learning

[Kha+19] and beyond. However, approximate Bayesian inference techniques are often needed due to the fact that it is non-trivial to derive a closed-form solution for the posterior [Bis06] of complex and non-linear models, e.g. DNNs. In general, there are three mainstream types for BNNs inference based on the way to construct the posterior distribution. The first one is Variational Inference (VI) [HV93]; [BB98], which approximates the posterior distribution by optimizing over a family of tractable distributions. The second one is Sampling-based approaches, which deliver a specific mechanism for drawing samples from the target posterior distribution. One famous example method of this type is Hamiltonian Monte Carlo [Nea92] based on Markov Chain Monte Carlo (MCMC) sampling. The last one is Laplace Approximation (LA) [DL91]; [Mac92b], which approximates the log-posterior distribution with a normal distribution. These three types differ in multiple requirements that are of interest to diverse applications. While VI and LA offer an analytical expression of the uncertainty in a relatively more efficient manner, their approximation quality is hard to guarantee. The sampling-based approaches produce samples and lack such an analytical expression but can approximate the true posterior with higher fidelity in theory. Due to the lightweight computation overhead required by the robotic applications, approaches from VI and LA have received immense attention in recent years [LSS20]. In this regard, we also seek to leverage VI and LA for developing the proposed introspective methods in this thesis.

Normalizing Flows NFs [KPB20] are a popular class of deep generative models featured by its exact evaluation of probability density with a variety of successful applications including image generation [DSB16]; [KD18], variational inference [RM15], semi-supervised learning [Izm+20], inverse problem [Ard+18], uncertainty estimation [CZG20]; [Cha+21]; [Pos+20] and OOD detection [KIW20]; [Nal+18] to name a few. There has been a large body of research on designing expressive flow-based architectures with different trade-offs on computational efficiency and modeling capacity, such as affine coupling flows [DSB16]; [KD18], auto-regressive flows [Hua+18]; [Dur+19], invertible ResNet blocks [Che+19] and ODEs-based maps [Gra+18]. While all these flows focus on satisfying the requirements of Jacobians tractability and mappings invertibility, the fundamental topological problem raised by these requirements received less attention [KPB20]; [Pap+21]. Some existing works targeting this problem attempt to increase the learning capacity of the transformation via mixture models [Pos+21], latent variable models [Cor+20]; [Din+19] or injecting carefully specified randomness [Nie+20]; [WKN20]. These methods are less practical-attractive because they either increase the memory consumption by expanding the width of transformations or sacrifice the exact likelihood computation ability. By contrast, thoughts on mitigating this constraint have been navigated to improving the expressivity of the base distribution [SSH22]; [Jai+20], revealing the trade-off between an appropriate base density and a sufficiently expressive transformation. This class of methods only adds slight computation overheads and thus is better suited for robotic applications, which motivates us to exploit such a model for the introspective method development.

2.3. Preliminaries

2.3.1. Bayesian Neural Networks

In general, a neural network can be modeled as a function $f^\omega(\mathbf{x}) = \mathbf{y}$ that maps from an input space \mathcal{X} to an output space \mathcal{Y} , where $\omega = \{W_{1:L}, \mathbf{b}_{1:L}\}$ are the weights of the network consisting of matrices W_i and biases \mathbf{b}_i for each of its L layers. In the training phase, the weights ω are determined by optimizing a loss function $E(f^\omega(\mathbf{x}_i), \mathbf{y}_i)$ for a given training data set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\}$. In contrast, Bayesian Neural Networks (BNNs) not only aim to find an optimal ω , but also defines a *posterior distribution* $p(\omega | \mathcal{D})$. The posterior over the space of parameters $p(\omega | \mathbf{x}, \mathbf{y})$ is modelled by assuming a prior distribution over the parameters $p(\omega)$ and applying Bayes theorem:

$$p(\omega, \mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{x}, \omega)p(\omega)}{p(\mathbf{y} | \mathbf{x})} \propto p(\mathbf{y} | \mathbf{x}, \omega)p(\omega). \quad (2.1)$$

Here, the normalization constant in (2.1) is called the model evidence $p(\mathbf{y} | \mathbf{x})$ which is defined as

$$p(\mathbf{y} | \mathbf{x}) = \int p(\mathbf{y} | \mathbf{x}, \omega)p(\omega)d\omega. \quad (2.2)$$

Given this posterior, inference on a new test sample $(\mathbf{x}^*, \mathbf{y}^*)$ can be done using the *predictive distribution*

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^* | \mathbf{x}^*, \omega)p(\omega | \mathcal{D})d\omega, \quad (2.3)$$

where for classification tasks the likelihood $p(\mathbf{y}^* | \mathbf{x}^*, \omega)$ is usually obtained from the *softmax* of the prediction $f^\omega(\mathbf{x}^*)$. The benefit of using (2.3) for predictions instead of only using the likelihood is that the model also incorporates the *epistemic* uncertainty, i.e. the one that stems from incorrect model parameters, thereby providing better (less overconfident) uncertainty estimates. Unfortunately, inferring the parameter posterior $p(\omega | \mathcal{D})$ is not tractable in all but the simplest cases due to the high dimensionality of the parameter space. Therefore, approximations need to be adopted, and here we introduce two performant and practical alternatives: the Monte-Carlo Dropout (MCD) and Kronecker-factored Laplace Approximation (LA).

Monte-Carlo Dropout (VI) Dropout [Sri+14] was originally proposed to regularize the training process of DNNs to improve their generalization performance. [GG16] showed that using dropout can be interpreted as Variational Inference (VI), using a distribution $q_\theta(\omega)$ for approximating the posterior $p(\omega | X, Y)$ in terms of the Kull-Leibler Divergence (KLD).

$\theta = \{\omega, p\}$, p is the vector of dropout rates of layers in which dropout is inserted. Minimizing KLD is equivalent to minimizing the negative Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta) = -\sum_{i=1}^N \int q_{\theta}(\omega) \log p(y_i | f^{\omega}(x_i)) d\omega + \text{KL}(q_{\theta}(\omega) || p(\omega)) \quad (2.4)$$

$$\approx -\sum_{i \in \mathcal{S}} \frac{N}{K} \int q_{\theta}(\omega) \log p(y_i | f^{\omega}(x_i)) d\omega + \text{KL}(q_{\theta}(\omega) || p(\omega)), \quad (2.5)$$

where \mathcal{S} is a mini-batch of size K . To estimate the expected LL in the first term, Monte Carlo integration is used, i.e. samples are generated from $q_{\theta}(\omega)$, and the integral is approximated by summing likelihood terms over the samples. The problem here is that using this standard method, this first term can not be derived with respect to θ , which is necessary to minimize $\mathcal{L}(\theta)$. Therefore, the *re-parameterization trick* is used, i.e. a bivariate transformation $g(\theta, \epsilon)$ is used to separate the parameters θ from samples $\epsilon \sim p(\epsilon)$ that are generated from a distribution with fixed parameters. Originally, this could be done only for a Gaussian dropout distribution, later [GHK17] showed that for Bernoulli dropout, a *continuous* relaxation of this *discrete* distribution can be found, i.e. a concrete distribution [MMT16], which can then be derived wrt. θ for optimization.

Laplace Approximation The idea of LA is to employ a second-order Taylor expansion at the maximum of the log posterior:

$$\log p(\omega | X, Y) \approx \log p(\omega^* | X, Y) - \frac{1}{2}(\omega - \omega^*)^T H(\omega - \omega^*), \quad (2.6)$$

where ω^* is the parameter vector that maximizes the log posterior and H is the Hessian of the negative log posterior. Note that the first derivative vanishes at ω^* and H is positive semidefinite (p.s.d) because ω^* is assumed to be a local maximum. After taking the exponential and normalizing we obtain

$$p(\omega | X, Y) \approx \mathcal{N}(\omega^*, H^{-1}). \quad (2.7)$$

Unfortunately, the dimensionality of this multi-variate normal distribution is in most cases too high to be practical. Also, H needs to be computed on the entire data set, which is also infeasible. Instead, it is approximated by the expected Hessian $\mathbb{E}_{p(X,Y)}[H]$, computed on mini-batches. To reduce the dimensionality, the first step is to assume independence across the layers of the DNNs, i.e. H is block-diagonal with L blocks H_i , one for each layer.

Under certain conditions, the Fisher information matrix F , which is the outer product of the first derivatives, is an approximation to the expected Hessian. Furthermore, in each layer i the block F_i can be approximated by a Kronecker product of two much smaller matrices G_i

and A_i , where $G_i = g_i g_i^T$ is the outer product of gradients of pre-activation of i -th layer and $A_i = a_{i-1} a_{i-1}^T$ is the outer product of activation from the previous layer. This is known as the Kronecker-factored approximate curvature (K-FAC) [MG15]. If a Gaussian prior is used and F is scaled by the size of the training set N , then the resulting posterior can be written as matrix normal distribution [GN99]:

$$W_i \sim \mathcal{MN}(W_i^*, (\sqrt{N}\mathbb{E}[A_i] + \sqrt{\tau}I)^{-1}, (\sqrt{N}\mathbb{E}[G_i] + \sqrt{\tau}I)^{-1}) \quad (2.8)$$

where τ is the standard deviation of the Gaussian prior. In practice, N and τ can be treated as hyper-parameters as well and tuned on a validation set.

Thanks to the practical and effective inference techniques introduced above, we adopt BNNs with MCD and LA for reliable uncertainty estimation in both **Publication 1** and **Publication 3**, aiming for introspective robot perception. More noteworthy, we also exploit these high-quality uncertainty estimates for semi-supervised adaptive classification in **Publication 1** and Active Learning (AL) for object detection in **Publication 3**. More technical details are presented in chapter 3.

2.3.2. Normalizing Flows

Normalizing Flows (NFs) are known to be universal distribution approximators [Pap+21]. That is, they can model a complex target distribution \mathbf{u} on a space \mathcal{R}^d by defining \mathbf{u} as a transformation $T_\phi: \mathcal{R}^d \rightarrow \mathcal{R}^d$ parameterized by ϕ from a well-defined base distribution $p_\psi(\mathbf{z})$ with parameters ψ :

$$\mathbf{u} = T_\phi(\mathbf{z}) \quad \text{where } \mathbf{z} \sim p_\psi(\mathbf{z}) \quad (2.9)$$

where $\mathbf{z} \in \mathcal{R}^d$ and p_ψ is commonly chosen as a uni-modal Gaussian. By designing a random variable T_ϕ with certain distribution to be a *diffeomorphism*, that is, a bijection where both T_ϕ and T_ϕ^{-1} are differentiable. We can compute the likelihood of the input \mathbf{u} *exactly* based on the change-of-variables formula [BR07]:

$$p(\mathbf{u}) = p_\psi(T_\phi^{-1}(\mathbf{u})) |\det(J_{T_\phi^{-1}}(\mathbf{u}))|, \quad (2.10)$$

where $J_{T_\phi^{-1}}(\mathbf{u}) \in \mathcal{R}^{d \times d}$ is the Jacobian of the inverse T_ϕ^{-1} with respect to \mathbf{u} . When the target distribution is unknown but samples thereof are available, we can estimate the parameter (ϕ, ψ) by minimizing the forward Kull-Leibler Divergence (KLD), which is equivalent to maximizing the expected Log-Likelihoods (LL).

$$\text{LL}(\phi, \psi) = \mathbb{E}_{p(\mathbf{u})} \left[\log(p_\psi(T_\phi^{-1}(\mathbf{u}))) + \log |\det(J_{T_\phi^{-1}}(\mathbf{u}))| \right] \quad (2.11)$$

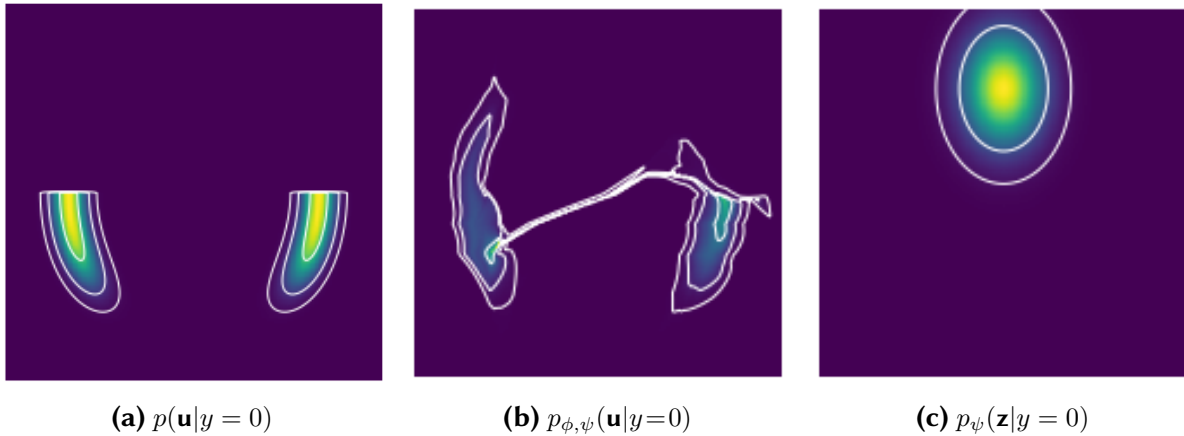


Figure 4.: Filament connect modes in the modeled class-conditional distribution (b) if using (trainable) uni-modal base (c) for the multi-modal target (a).

Topological Mismatch However, by definition, T_{ϕ} is a diffeomorphism and the base distribution $p_{\psi}(\mathbf{z})$ is usually a uni-modal Gaussian (e.g. Fig. 4c). This raises a problem for modeling data distribution with different topological properties, e.g. well-separated multi-modal distributions, and distributions with disconnected components (e.g., Fig. 4a), which seems critical for data hypothesized to follow a clustering structure e.g. the data with different class labels. For example, in Fig. 4b, the Mixture of Gaussians (MOG) with one uni-modal Gaussian per class struggles to recover the target distribution with two modes Fig. 4b. As proofed by [Cor+20], flows that can recover the target distribution perfectly need to have infinite *bi-Lipshitz constant* which would make the flows numerically "non-invertible", causing optimization instability and unreliability on computing the exact LL [Beh+21], which might be destructive when applying NFs for OOD detection. One way to mitigate this is by enriching the expressiveness of flows such as increasing depth (e.g. number of layers) or width (e.g. mixtures of flow), but this could escalate the computational cost and memory burden, which is unfavorable for robotic systems commonly with restricted computing resources. As this limitation is affected by both the transformation and the base distribution, there is a *trade-off* between a flexible base distribution and an expressive transform to capture desirable topological properties of the target distribution [Jai+20].

We attempt to compensate for the complexity of the transformation with the elasticity of the base distribution, which might be beneficial for diverse types of flow architectures.

In **Publication 2**, we tackle this fundamental problem in NFs with a novel combination of an expressive base distribution and an information-theoretic training objective. Targeting the problem of the close-set assumption, we employ NFs for OOD detection for robot perception in **Publication 2** and detecting feasible assembly in **Publication 5**. More key methodological details will be articulated in the following chapters.

2.3.3. Graph Neural Networks

A GNN operates on an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes \mathcal{V} and edges \mathcal{E} , where every node $v \in \mathcal{V}$ is assigned with a feature vector $\phi(v)$. It updates node features by exchanging information between neighboring nodes. This is done with multiple Message Passing layers [Gil+17]. For each layer l , let $\mathbf{h}_i^0 = \phi(v_i)$ be the input features of node v_i and \mathcal{N}_i its set of neighboring nodes. Then, we can define a three-step process to update these features:

- [1] *Gather* feature from neighboring nodes: $\{\mathbf{h}_j^{l-1}\}_{j \in \mathcal{N}_i}$.
- [2] *Aggregate* messages from the neighboring nodes: $\mathbf{m}_i^l = g_\omega(\{\mathbf{h}_j^{l-1}\}_{j \in \mathcal{N}_i})$.
- [3] *Update* features of node v_i : $\mathbf{h}_i^l = f_\phi(\mathbf{h}_i^{l-1}, \mathbf{m}_i^l)$.

The function g_ω can be either constant (e.g. sum) or learned during training. The term f_ϕ is a Neural Network parameterized by ϕ . Both, f_ϕ and g_ω , are shared across all nodes in the graph, making GNNs efficient and independent of the number of nodes in the graph.

In **Publication 4**, we apply a Graph Attention Network (GAT) [Vel+17]; [BAY21], a popular variant of GNNs, that defines g_ω as attention:

$$\mathbf{m}_i^l = \sum_{j \in \mathcal{N}_i} (\alpha_{i,j} \cdot \mathbf{h}_j^{l-1}), \quad (2.12)$$

$$\mathbf{h}_i^l = \mathbf{W}_1 \cdot \alpha_{i,i} \mathbf{h}_i^{l-1} + \mathbf{W}_1 \cdot \mathbf{m}_i^l, \quad (2.13)$$

$$\alpha_{i,j} = \frac{\exp(\mathbf{a} \cdot \sigma(\mathbf{W}_2[\mathbf{h}_i^{l-1} \parallel \mathbf{h}_j^{l-1} \parallel e_{i,j}]))}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp(\mathbf{a} \cdot \sigma(\mathbf{W}_2[\mathbf{h}_i^{l-1} \parallel \mathbf{h}_k^{l-1} \parallel e_{i,k}]))}, \quad (2.14)$$

where \mathbf{W}_1 , \mathbf{W}_2 , and \mathbf{a} are learned, σ is a Leaky ReLU activation function, and $[a \parallel b]$ is a concatenation operator between a and b .

Heterogeneous Graph Furthermore, for the graph construction of an assembly in **Publication 4**, we utilize heterogeneous Graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generalizes graphs to multiple types of nodes and edges [SH13]. Each node $v \in \mathcal{V}$ belongs to one particular node type $\psi_n(v)$ and analogously each edge $e \in \mathcal{E}$ to an edge type $\psi_e(e)$. In [Wan+19], the authors extend GAT to a heterogeneous graph setting. This is accomplished by obtaining for each node a different updated feature vector per group of specific neighboring source nodes and edge types and aggregating the features to obtain a single result, for instance using a sum. This formulation is essential, as every type of neighboring node may have a different feature dimension.

In contrast to the common DNNs, GNNs possess the capability to model graph-structured inductive basis, which is crucial for data modality with such a structure. In **Publication 4** and **Publication 5**, we apply it to model the spatial structure of an aluminum assembly. More noteworthy, in **Publication 5**, we propose an introspective method that uses NFs to capture

the density of embeddings from a GNN and detect infeasible assemblies based on the predicted likelihoods. We will introduce more details about the main idea and key innovations in Chapter 4.

3. Introspective Methods for Robotic Perception

Frequently, the Achilles' heel in robotics systems lies in perception. This is largely attributed to the complex real-world surroundings a robot may encounter, which often violate strong assumptions made by the off-the-shelf perception algorithms. These assumptions span from hardware (e.g. certain lighting conditions or image quality) to software (e.g. the closed set assumption of learning-based algorithms). Considering such vulnerability, a trustable robotic perception system demands methods that can overcome such limitations.

We believe that introspective methods possess huge potential to address such challenges. In this chapter, we present three introspective methods for robotic perception in the following sections. The first method (Uncertainty-based Adaptive Classification with Scene Contexts in **Publication 1**) can provide and utilize effective uncertainty estimates, based on which a robot can autonomously learn to adapt across different data distributions. The second one (Flow-based Open-Set Object Detection in **Publication 2**) is able to endow robots with introspection against OOD objects by mitigating a fundamental issue in NFs. The third method (Active Learning for Sim-to-Real Object Detection in **Publication 3**) presents how to leverage the introspection for long-term autonomous learning by using uncertainty estimation for active learning across domains, i.e. from simulation to reality.

In the following, to facilitate a concise and self-contained reading flow, we will primarily focus on a brief elaboration of the related work analysis and the method description followed by an outline of experimental results. For more details and full experimental results, we kindly refer the reader to the original manuscript of the corresponding publication in the appendix A.

3.1. Uncertainty-based Adaptive Classification with Scene Contexts

The gap between the training and test data distribution deteriorates the performance of most of classifiers. This problem is hardly avoidable when the classifier is trained on an easily obtainable dataset such as a public large-scale or synthetic dataset and then deployed in a real environment.

In this case, the effects of high-quality uncertainty estimates can be exploited by adapting the classifier to the test environment with *as little manual effort as possible*. To this end, in **Publication 1**, we propose methods to make robots learn new objects more introspectively, by improving their awareness of possible mistakes and leveraging this in two ways: first, for more effectively incorporating context information (if available) through smoothing over all object predictions using a CRF, which a popular model in Probabilistic Graphical Models (PGM), and second, for exploiting this in semi-supervised domain adaptation, where the mostly correct predictions are automatically obtained as adaptation data while asking humans for help with the more uncertain ones.

3.1.1. Related Work

Combining DL and PGM [LSL15] trained a DNN and a CRF jointly for depth estimation. At the same time, Tompson et al. [Tom+14] integrated Markov random fields with DNNs for pose estimation. [WY16] combined DL with Bayesian networks for recommendation systems and topic models. Johnson et al. [Joh+16] proposed Structured variational autoencoder (SVAE) to learn a structured and thus more interpretable latent representation. Our work differs from them in the way of training the models. In order to analyze the effect of fusing uncertainty estimates into a PGM, we train them separately. Similar to us, Liu et al. [LLS15] combined features learned from DNNs and CRFs for segmentation tasks. But they trained another classifier with these features for the unary potentials without evaluating the effects of uncertainty estimates. In contrast, we fuse the improved uncertainty estimates from a BNN into the CRF and infer the joint probability with the pairwise cues from the scene contexts.

Semi- and Self-Supervised Domain Adaptation Some works [KZB19]; [XXL19] aim to learn a more generalized feature distribution via designing specific *pretext* tasks without explicit human supervisions (e.g. class labels). Others [Lin+17]; [Wan+16]; [Zou+18] tried to employ true positives as self-supervisions for adaptation. [Zou+18] mentioned the class imbalance problem and proposed to mitigate it by normalizing the class-wise confidence. We also observe that this problem is serious for this task and resolve this issue with class-balanced augmentations in our experiments.

3.1.2. Methodology

Incorporating Scene Contexts via Conditional Random Fields

While BNNs are very useful in providing reliable uncertainty estimates for single object instances, it does not incorporate any *context clues* that are specific for a scene, such that, e.g. more likely object constellations can be accounted for. In order to exploit such contextual information within the classification, we combine the output of BNNs and the co-occurrence relationship between objects within a scene via a CRF (see Fig. 5).

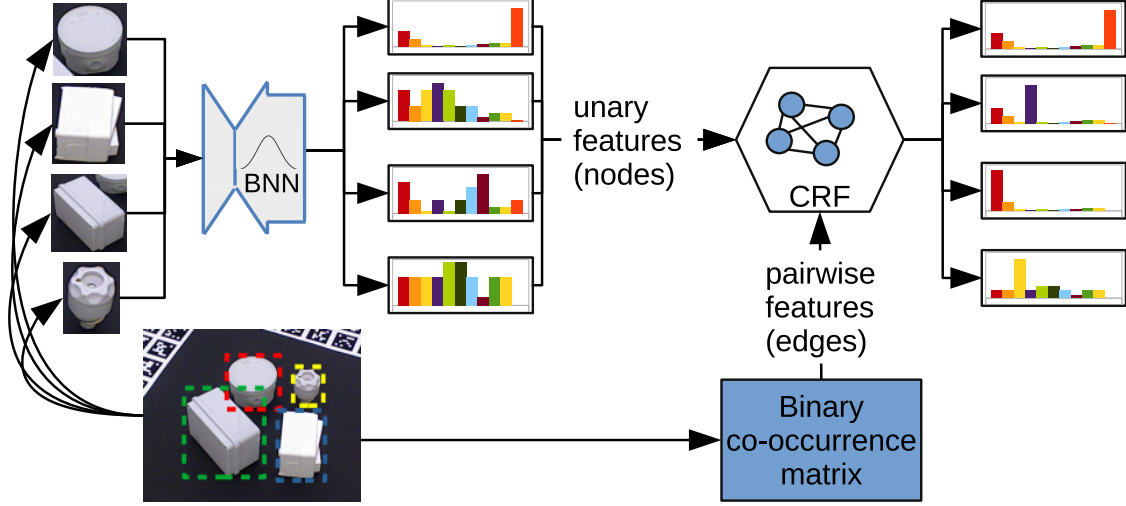


Figure 5.: The combination of BNNs and CRFs: the predictive distributions of objects in the scene from BNNs serve as unary features in the CRFs, which can take into account the contextual information from the scene of objects.

In detail, we define a scene as a set of n object instances $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with the corresponding class labels $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ represented as one-hot encodings, i.e. $\mathbf{y}_i \in \{0, 1\}^C$ and $\sum_{j=1}^C y_{ij} = 1$, where C is the number of object classes. The CRFs models the joint probability $p(\mathbf{y} | \mathbf{x})$ as an undirected graph consisting of cliques of random variables. A *pairwise* CRF is used, consisting of nodes \mathcal{V} and edges \mathcal{E} , where the node potentials are modeled as $\phi_u(\mathbf{x}_i, \mathbf{y}_i)$ for individual object instances and the edge potentials $\phi_p(\mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j)$ for pairs of objects $(\mathbf{x}_i, \mathbf{x}_j)$ which are in the scene. Concretely, we define ϕ_u as the predictive probability of each instance (see Eq. (2.3)) and ϕ_p as the co-occurrence probability of two objects. Co-occurrence probabilities can be obtained from an independent source, for example, we mined from word "co-occurrence" in WikiHow websites and many household objects have similar appearances and contexts at the same time.. In case the list of expected objects in the scene is known, the pairwise feature is binary and provided automatically per scene. Hence, the likelihood of the CRF has the following form:

$$p(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\theta})} \exp \left(\theta_u \sum_{i \in \mathcal{V}} p(\mathbf{y}_i | \mathbf{x}_i) + \theta_p \sum_{(i,j) \in \mathcal{E}} M(\mathbf{y}_i, \mathbf{y}_j) \right), \quad (3.1)$$

where $\boldsymbol{\theta} = \{\theta_u, \theta_p\}$ are the node and edge weights respectively, Z is the partition function, and M is a $C \times C$ binary matrix modelling the co-occurrence of two object classes \mathbf{y}_i and \mathbf{y}_j . The training process of CRFs involves minimizing the negative log-likelihood, i.e. finding optimal model parameters θ^* such that $\theta^* = \arg \min_{\theta} \{-\log p(\mathbf{y} | \mathbf{x}; \theta)\}$. For this, we employ stochastic gradient descent with momentum, which requires the calculation of gradients and thus an inference step for the likelihood shown in Eq. (3.1). We use a fully connected CRFs, i.e. an exact inference of the likelihood is intractable. Therefore, we apply the classical

technique Loopy Belief Propagation (LBP) [MWJ13] for approximate inference in CRFs. In our implementation, we use the C++ library UPGM++ [RGG15] for this purpose.

Adaptive Classification based on Uncertainty Estimation

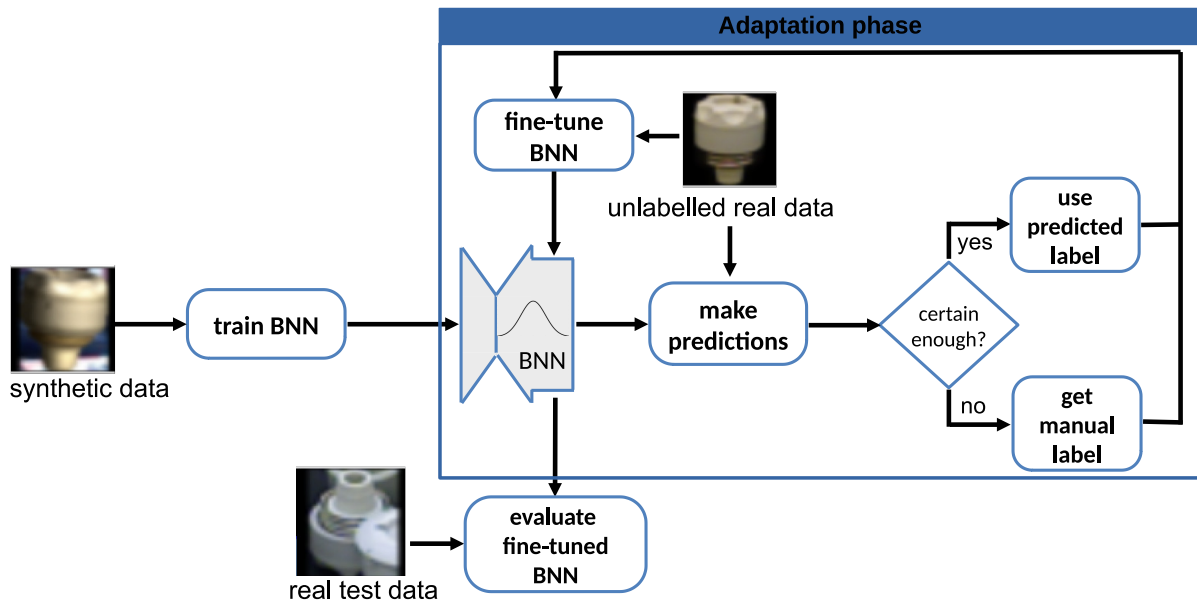


Figure 6.: The flowchart for adaptive learning in domain adaptation. Better uncertainty estimates can help distinguish certain predictions in automatic labeling during the adaptation phase (illustrated on the T-LESS dataset and best viewed in color).

It is common that the test data in the real environment does not have exactly the same distribution as the training set, which leads to a significant performance drop in testing. Therefore, it would be more efficient to enable the classifier to adapt to the test environment by fine-tuning itself. For this, the classifier needs to be introspective, that is, to express reliable confidence about its predictions.

First, the classifier is trained on an easily obtainable or accessible dataset, which can be a large-scale public or synthetic one. **Next**, in the adaptation phase the classifier is able to adapt to the test data by fine-tuning itself on the so-called adaptation dataset.

We aim to obtain this adaptation dataset with as little manual effort as possible. Thereby, annotations in this dataset are collected in a semi-supervised way (including both automatic and manual manner, as illustrated in Fig. 6). On the one hand, the predictions with high confidence are used for pseudo labels, thus requiring the classifier to provide reliable uncertainty estimation for both correct and false predictions. On the other hand, the classifier would ask people to label a small and random portion of data interactively. In the end, the adapted classifier will be deployed in the real environment.

3.1.3. Summary of Results

In **Publication 1**, we firstly compared performance on uncertainty estimates of two approximate inference techniques for BNNs, which are concrete dropout [GHK17] and LA [RBB18] on a household objects dataset in terms of comprehensive metrics. Then, the one with better performance was applied in the following experiments, which are to evaluate **(1)** the combination with CRFs and **(2)** the adaptive learning for domain adaptation respectively.

We observe the following tendencies in these experiments:

- In the uncertainty estimation comparison, we witness clear evidence of better performance brought by BNNs with both MCD and LA, while the former outperforms the latter slightly.
- When coupling with CRFs and fusing the scene contexts with the probabilistic predictions, we observe distinct improvements in both uncertainty estimation and predictive performance.
- Uncertainty-based adaptive classification can save significant labeling efforts (3%) while achieving similar performance compared to the fine-tuned version based on the whole data set (100%).

3.2. Flow-based Open-Set Object Detection

For reliable identification of OOD data, which is not well represented in the training set, we propose to equip NFs with efficient but flexible base distributions for OOD detection in robot learning. As illustrated in Fig. 7, we replace the frequently used uni-modal Gaussian base distribution with the Conditional Resampled Base Distribution (CRSB), a class-conditional version of a learnable base distribution for mitigating the topological problem in NFs – Resampled Base Distribution (RSB) [SSH22]. Moreover, we adapt our CRSB with an adapted Information Bottleneck (IB) objective [Ard+20] to balance fusing class-conditional information with the marginalized density estimation capabilities in NFs.

3.2.1. Related Work

NFs for OOD Detection NFs have been widely adapted for OOD detection due to its superior density estimation [Yan+21]. For example, though with some counter-intuitive observations on raw data space [Nal+18], NFs have demonstrated encouraging OOD detection results with additional refinements for raw data [Ren+19]; [Nal+19b]; [JSY22] or directly based on task-relevant feature embeddings [KIW20]; [Zha+20]; [CZG23]; [Fen+23a]. In this work, we directly apply NFs on the feature space. To note that, another principle direction is to estimate the error bound for this task [COB22]. Recently hybrid models [Nal+19a];

[Zha+20]; [CZ22] have shown remarkable performance gain on OOD detection by modeling the joint distribution of both data and its class labels. Such works suggest that class labels can provide useful information. However, directly performing class conditional modeling with NFs for OOD detection results in performance degradation. [TPB00]; [Ard+20] mitigate such performance degradation by utilizing IB for training NFs. This explicitly controls the trade-off between generative and discriminative modeling [Mac+21]. However, these works on OOD detection utilize NFs without much concern for the fundamental topological problem as the first citizen. Therefore, complementary to these approaches, we examine the problem of topological mismatch of NFs for OOD detection.

OOD Detection in Object Detectors OOD detection research has focused on image classification [Yan+21], which may be limited in relevance to robotic vision. In robotics, we may often need both categorization and localization of objects of interest. Therefore, we focus on object detection in open-set conditions here. In this domain, uncertainty estimation [Gaw+23] has been considered propitious for OOD detection but suffered from computation burdens on runtime [Mil+18]; [HSW20] or memory costs [LPB17]. To address this, instead of directly applying uncertainty estimation techniques for object detection [HSW20]; [Lee+22], another popular approach is to explicitly formulate the problem as OOD detection tasks [Mil+21]; [Du+22a]; [Li+22b]; [Kum+23]; [Du+22b]. Amongst them, NFs has been utilized as an expressive density estimator [Li+22b]; [Kum+23]. However, despite the encouraging results, these approaches have not examined the problem of topological mismatch in NFs. As this might prevent additional performance improvements, this work examines the topology-matching NFs for OOD detection in object detectors.

3.2.2. Methodology

Given an image $\mathbf{x} \in \mathcal{X}$ and a trained object detector F_θ that localizes a set of objects with corresponding bounding box coordinates $\mathbf{b}_i \in \mathcal{R}^4$ as well as class label $y_i \in \mathcal{Y} = \{1, 2, \dots, C\}$, the task is to distinguish if $(\mathbf{x}, \mathbf{b}_i, y_i)$ is ID, i.e., drawn from \mathcal{P}_{id} , or OOD, i.e., belongs to the unknown distribution \mathcal{P}_{ood} . For conciseness, from now on we omit the suffix i and use y to denote the class label without further notice. As discussed, a powerful OOD detection can be obtained via density estimation using NFs. This density estimator identifies OOD objects with low likelihoods after being trained *only* on data drawn from \mathcal{P}_{id} . Following relevant prior [Wei+22a]; [Mil+21], we use the semantically rich logit space (pre-softmax layer) for density estimation. To note that, our method can be readily applied to other (high-dimensional) latent feature spaces.

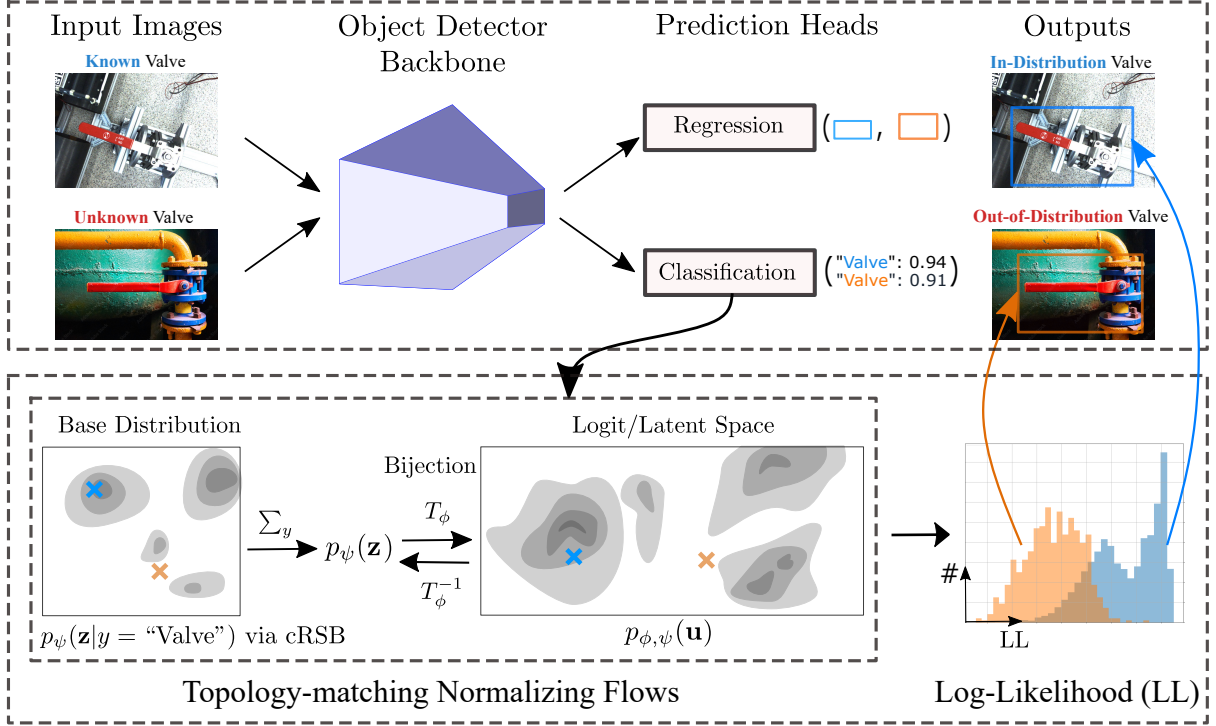


Figure 7.: The proposed architecture. We overcome the topological mismatch problem in NFs to accurately model ID density. That is, the CRSB base distribution trained with IB $p_\psi(\mathbf{z}|y)$ can, e.g., adapt the numbers of modes to match target distribution with complex topology. Then we can identify OOD objects by low predicted log-likelihoods more reliably (best viewed in color).

Topology-match Normalizing Flows

We propose to capture the complex topological properties in the target distribution with a more expressive base distribution instead of the uni-modal Gaussian. To the end, we introduce CRSB by extending a powerful unconditional base distribution RSB [SSH22] with class-conditional modeling. RSB deforms a uni-modal Gaussian in a learnable manner to obtain more complex distributions via Learned Accept/Reject Sampling (LARS) [BM19]. LARS iteratively re-weights samples drawn from a proposal distribution $\pi(\mathbf{z})$, e.g. a standard Gaussian, through a learned acceptance function $a_\psi : \mathcal{R}^d \rightarrow [0, 1]$. To reduce the computation cost in practice, this process is truncated by accepting the T -th samples if the previous $T - 1$ samples get rejected. To take into account class-conditional information, we conditionalize the learnable acceptance function $a_\psi(\mathbf{z}|y)$. As a result, we have the conditional base distribution:

$$p_\psi(\mathbf{z}|y) = (1 - \alpha_T) \frac{a_\psi(\mathbf{z}|y)\pi(\mathbf{z})}{Z_y} + \alpha_T \pi(\mathbf{z}), \quad (3.2)$$

where $a_\psi : \mathcal{R}^d \rightarrow [0, 1]^C$ and $\alpha_T = (1 - Z_y)^{T-1}$, where $Z_y \in \mathcal{R}$ is the normalization factor for $a_\psi(\mathbf{z}|y)\pi(\mathbf{z})$. This factor can be estimated via Monte Carlo Sampling.

Training with Information Bottleneck

Unfortunately, directly training NFs with a conditional base distribution can lead to under-performance as observed in experiments (reported by [Fet+19]). We attribute this to the lack of explicit control for the balance between generative and discriminative modeling in the likelihood-based training objective of NFs. To alleviate this, we train the normalizing flow with a class-conditional base distribution using the IB objective [TPB00]. To abuse the notations, we denote random variables by capital letters such as U, Z, Y , and their realizations by lowercase letters such as $\mathbf{u}, \mathbf{z}, y$. The IB minimizes the mutual information $I(U, Z)$ between U and Z , while simultaneously maximizing the mutual information $I(Z, Y)$ between Z and Y . Intuitively, the IB trades off between the objectives of modeling the class conditional information $p(\mathbf{u}|y)$ with the marginalized density $p(\mathbf{u})$, thus allowing to leverage the class-conditional structure to facilitate more effective density estimation for data characterized with semantic classes. However, the IB is not directly applicable to latent class-conditional distributions in NFs since the bijection T_ϕ is lossless by design. Thus, for trading off the class-conditional information with density estimation capabilities, we adapt the approach proposed by [Ard+20] for our CRSB. Specifically, we inject a small amount of noise ϵ into the input U and hence $Z_\epsilon = T_\phi^{-1}(U + \epsilon)$. Further we define an asymptotically exact version of mutual information, namely the Mutual Cross-information:

$$\mathcal{L}_{\text{IBNF}} = CI(U, Z_\epsilon) - \beta CI(Z_\epsilon, Y) \quad (3.3)$$

$$CI(U, Z_\epsilon) = \mathbb{E}_{p(\mathbf{u}), p(\epsilon)} \left[-\log \sum_{y'} p_\psi(\mathbf{z}_\epsilon | y') - \log |\det(J_{T_\phi^{-1}}(\mathbf{u} + \epsilon))| \right], \quad (3.4)$$

$$CI(Z_\epsilon, Y) = \mathbb{E}_{p(y)} \left[\log \frac{p_\psi(\mathbf{z}_\epsilon | y) p(y)}{\sum_{y'} p_\psi(\mathbf{z}_\epsilon | y') p(y')} \right], \quad (3.5)$$

where $\mathbf{z}_\epsilon = T_\phi^{-1}(\mathbf{u} + \epsilon)$, $p(\epsilon) = \mathcal{N}(0, \sigma^2 \mathcal{I}_d)$ is a zero-meaned Gaussian with variance σ^2 , and β trades off class information and generative density estimation. With flexible conditional base distributions defined in Eq. 3.2, we can train the *topology-matching* Normalizing Flows (NFs) with Information Bottleneck (IB) by substituting Conditional Resampled Base Distribution (CRSB) into the conditional base probability $p_\psi(\mathbf{z}|y)$ in Eq. 3.4 and 3.5. More noteworthy, we observed that the IB is able to regularize the acceptance rate learning for CRSB to better assimilate the topological structure of the target distribution, leading to an overall improved performance on accurately approximating the complex target distribution.

3.2.3. Summary of Results

In **Publication 2**, we conduct experiments to validate the proposed method and demonstrate its benefits and applicability. First, we evaluate the ability to match the topology of the

target distribution with synthetic density estimation for distributions with distinct topological properties. In this part, there are three synthetic data sets: two moons, two rings, and a circle of Gaussians. We then evaluate the OOD detection performance on two object-detection data sets adapted from their public counterparts [Eve+10]; [Lin+14] for open-set (OS) experiments: Pascal-VOC-OS and MS-COCO-OS based on Glow [KD18] and a pre-trained Faster-RCNN [Ren+15] provided by [Mil+21] for a fair comparison. To showcase the practicality, we deploy the one-stage object detector Yolov7 [WBL23] equipped with the proposed method on a real aerial manipulation robot along with the run-time and memory analysis (more details in Chapter 5).

To summarize, in this work, we present an OOD detection approach using topology-matching NFs, which is powerful and yet resource-efficient for open-set object detection. It is applicable to diverse object detectors with minor changes and no loss of prediction performance. Moreover, our approach is sampling-free, i.e., only a single forward pass is required for efficient test-time inference while keeping the space memory tractable.

3.3. Active Learning for Sim-to-Real Object Detection

A large amount of annotated data is required by many DL algorithms, which is not available in diverse scenarios in which we would like to deploy our robots such as service or field robots. A compelling solution is to learn from synthetic data. Like this, a large amount of annotated data can be obtained from simulation with relatively less time and manual efforts [Bou+18]; [Geo+17]; [Tob+17]. The so-called *Sim-to-Real gap* is the main barrier to transferring this technique to real-world robotic perception. In this work, we investigate the question: *How to bridge the Sim-to-Real gap with minimum annotation efforts by exploiting introspection?*

Having a model trained on synthetic images, we propose an Active Learning (AL) pipeline that can efficiently bridge the still present Sim-to-Real gap (see Fig. 8) by utilizing the uncertainty estimates from BNNs. In contrast to the method presented in **Publication 1**, we here aim for autonomous acquisition of as few annotated real images as possible. To this end, we devise a simple yet effective strategy to mitigate the lack of diversity in the selected data, caused by the label distribution shift between simulation and real domain [PDS19]; [Zha+21].

3.3.1. Related Work

Sim-to-Real Transfer Sim-to-real transfer is mainly tackled with Domain Randomization (DR) and Domain Adaptation (DA). The former treats the real test scenario as one instance of many synthetic ones generated by randomizing the parameters of the synthesizer such as materials, lightening, backgrounds, and plausible geometric configurations [Hin+18]; [Hod+19]. In contrast, DA focuses on learning domain-invariant representations across the

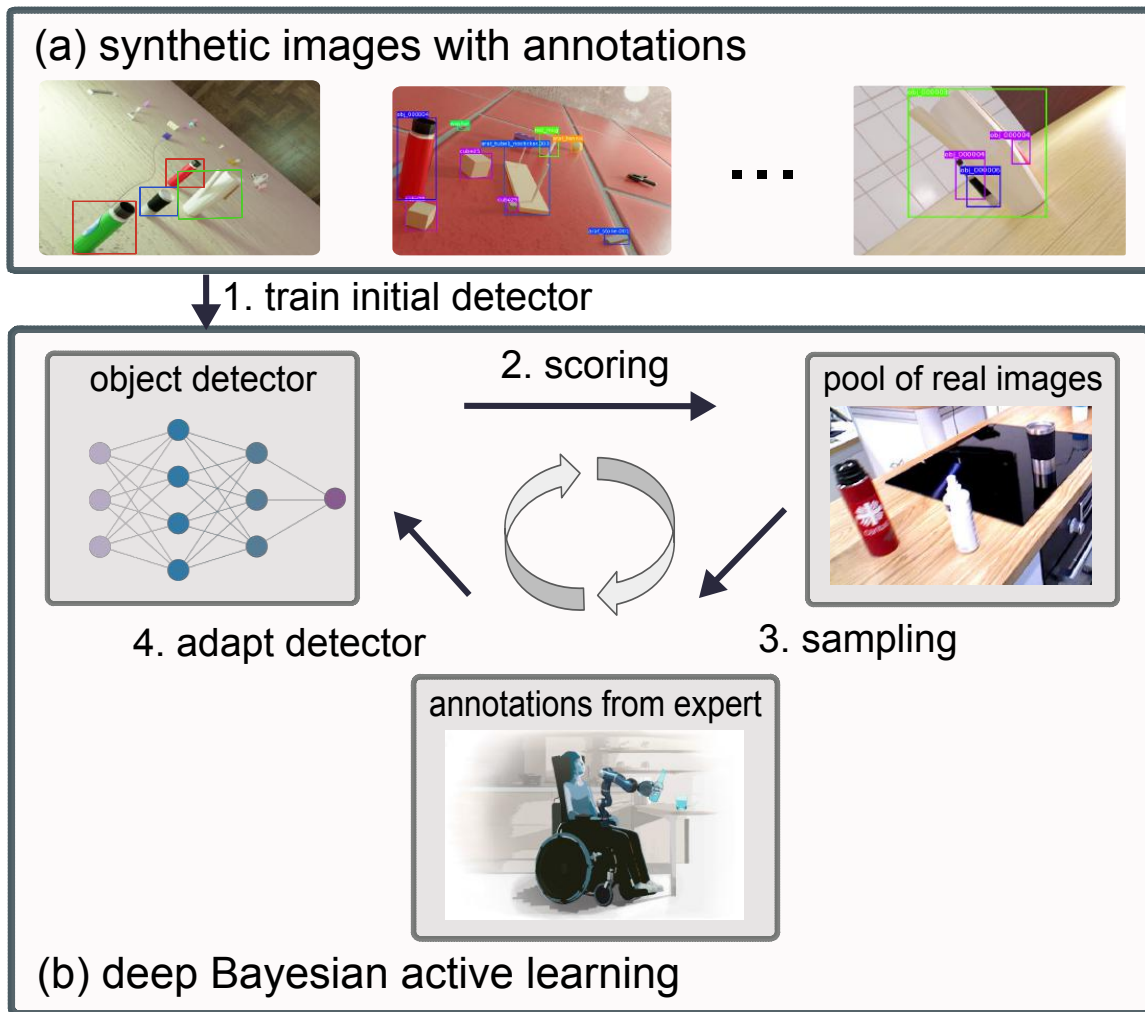


Figure 8.: The proposed Sim-to-Real pipeline. Using labeled synthetic images, we first train an initial BNNs object detector. Then, we rely on deep Bayesian AL to select the most informative images from a pool of unlabeled real images. The scoring of all the images in the pool is obtained via an acquisition function, while sampling is applied to deal with the foreground class imbalance problem. Based on the selected images, the human expert performs the annotation and the detector is adapted via fine-tuning. The process is repeated to close for Sim-to-Real transfer.

different domains (e.g. synthetic and real domain in this context) by sometimes including data of the target domain [Bou+18]. Though DA has achieved impressive performance, as mentioned by different researchers, when only relying on unlabeled data, the domain gap is hard to diminish both in theory [Tan20] and in practice [Zhu+19]; [Che+18]. Considering this issue, the paradigm of active learning is appealing to address the reality gap by utilizing annotated real data in an efficient way. In pool-set-based active learning [CGJ96], the aim is to reach a certain level of performance with as little data as possible. In the case of supervised learning, the data is selected based on their informativeness, which can be measured by different quantities such as the output uncertainty, the disagreement of a committee, or the expected

model change [Fen+19a]; [KVG19]. We also stress that active learning is complementary to the aforementioned techniques. While recent works such as [Su+20]; [Pra+21] argue for the fusion of DA and active learning to obtain better performance, we additionally use DR in this work. Nevertheless, none of them consider employing BNNs for this purpose and most of them focus on classification tasks, which are less relevant for the robots in the real world. [Wen+19] apply BNNs for DA, but they only focus on conventional passive learning paradigms and classification tasks. We aim to study the active learning paradigm for Sim-to-Real transfer on a more challenging real-world object detection task, which is arguably more relevant for various use cases.

Active Learning for Object Detection In the context of AL for object detection, specific metrics related to characteristics of the underlying network can be applied [Agh+19]. While in [RUN18] the margin of the bounding box scores in different layers is used, [Kao+18] consider the localization tightness and stability. Meanwhile, uncertainty-based approaches [Fen+19a]; [Cho+21]; [Pra+21] are also able to achieve competitive performances in the field of object detection. Most uncertainty-based approaches are built on BNNs [GG16] which can produce more reliable uncertainty estimates. Along with its theoretic soundness, the task-agnostic characteristic of these approaches can facilitate wider applicability for different fields. While some only exploit the classification branch for the uncertainty estimation [Mil+18], others [HSW20] consider both classification and regression branches. Yet, they rely on a larger amount of annotated real-world data to initialize the training of the model and update the model in each iteration, while we assume a relatively small amount of real data.

3.3.2. Methodology

With the uncertainty estimates of an object detector based on BNNs in prior work, called BayesOD [HSW20], the AL pipeline needs to choose the images for annotation. This selection of images is done via an acquisition function. Moreover, due to the domain shift between S and R , a sampling strategy is devised to mitigate the bias in the selected data set. We describe below these components and our design choices.

We consider two domains: the simulation domain S and the real domain R . In S , we assume the availability of annotated data set, i.e., given the synthetic data \mathbf{x}^S and annotated labels \mathbf{y}^S , we denote the synthetic data set as $\mathcal{D}_S = \{(\mathbf{x}_i^S, \mathbf{y}_i^S)\}_{i=1}^{N_S}$ where N_S is the number of data points. In contrary, R contains an unlabeled data set $\mathcal{D}_T = \{\mathbf{x}_i^R\}_{i=1}^{N_R}$ which constitutes of N_R number of real images \mathbf{x}^R . We further extend the notations to define an object detection task including classification (*cls*) and regression (*reg*) tasks. Given the space of inputs \mathcal{X} (both synthetic and real images) and outputs \mathcal{Y} (sets of object classes \mathbf{c} and their 2D location as bounding boxes \mathbf{b}), we define the object detector as a function $\mathcal{M}_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters θ . Naturally, our objective is to obtain an object detector in the real domain R , for which synthetic data \mathcal{D}_S can be exploited.

Acquisition Function

We define the acquisition function based on the uncertainty estimates from the BayesOD. In this step, the acquisition function is used to obtain the informativeness scores for each detected instance on one image, and then *aggregated* into one final score to represent the informativeness of the entire image. Once the scores are obtained for all the images in the pool set \mathcal{D}_{pool} , we sample a subset of them for annotation in order to adapt the model. Specifically, we consider uncertainty from both *category classification* and *bounding box regression*, which are referred to as semantic and spatial uncertainty respectively [Hal+20]. For the semantic uncertainty of the j -th detection instance on an image, given the Shannon Entropy measure $\mathcal{H}(\cdot)$, the *cls* acquisition function $\mathcal{U}_{j,cls}$ is modeled with a Bernoulli distribution as:

$$\begin{aligned}\mathcal{U}_{j,cls} &= \sum_{i=1}^{|\mathcal{C}|} \mathcal{H}(p(c_i|\mathbf{x}^*, \mathcal{D}_{train})), \\ &= \sum_{i=1}^{|\mathcal{C}|} [-p(c_i|\mathbf{x}^*, \mathcal{D}_{train}) \log p(c_i|\mathbf{x}^*, \mathcal{D}_{train}) \\ &\quad - (1 - p(c_i|\mathbf{x}^*, \mathcal{D}_{train})) \log (1 - p(c_i|\mathbf{x}^*, \mathcal{D}_{train}))].\end{aligned}\tag{3.6}$$

In (3.6), the steps follow from the definition of the entropy, and optimizing the given measure is equivalent to maximizing the information gain [Mac92c] or information content.

The uncertainty from regression is defined as differential entropy of $p(\mathbf{b}|\mathbf{x}^*, \mathcal{D}_{train})$ which is approximated by a multivariate Gaussian with covariance matrix \mathbf{C}_b calculated from the samples of predicted bounding boxes:

$$\begin{aligned}\mathcal{U}_{j,reg} &= \mathcal{H}(p(\mathbf{b}|\mathbf{x}^*, \mathcal{D}_{train})) \\ &= \frac{k}{2} + \frac{k}{2} \ln(2\pi) + \frac{1}{2} \ln(|\mathbf{C}_b|),\end{aligned}\tag{3.7}$$

where k is the dimensionality of random variable \mathbf{b} . Again, this regression acquisition function $\mathcal{U}_{j,reg}$ follows from the definition of entropy for Gaussian distributions and represents the information content of an image.

We choose to exploit these two quantities by a combination function $comb(\cdot)$, in order to produce the uncertainty score for each of N_k detected instances on k -th image. Then, the acquisition function for k -th image \mathcal{A} is defined by aggregating scores with a function $agg(\cdot)$ denoted by:

$$\mathcal{A}(\mathbf{x}_k) = agg_{j \in N_k}(comb(\mathcal{U}_{j,cls}, \mathcal{U}_{j,reg})),\tag{3.8}$$

The combination function $comb(\cdot)$ can be a weighted sum (*sum*) or maximum (*max*) operation [Cho+21]. The aggregation function $agg(\cdot)$ can be a maximum (*max*), summation (*sum*)

or average (*avg*) operation [RUN18]. What motivates this is the problem itself, i.e. object detection involves both *cls* and *reg* tasks and multiple instances in one image.

Sampling Strategy

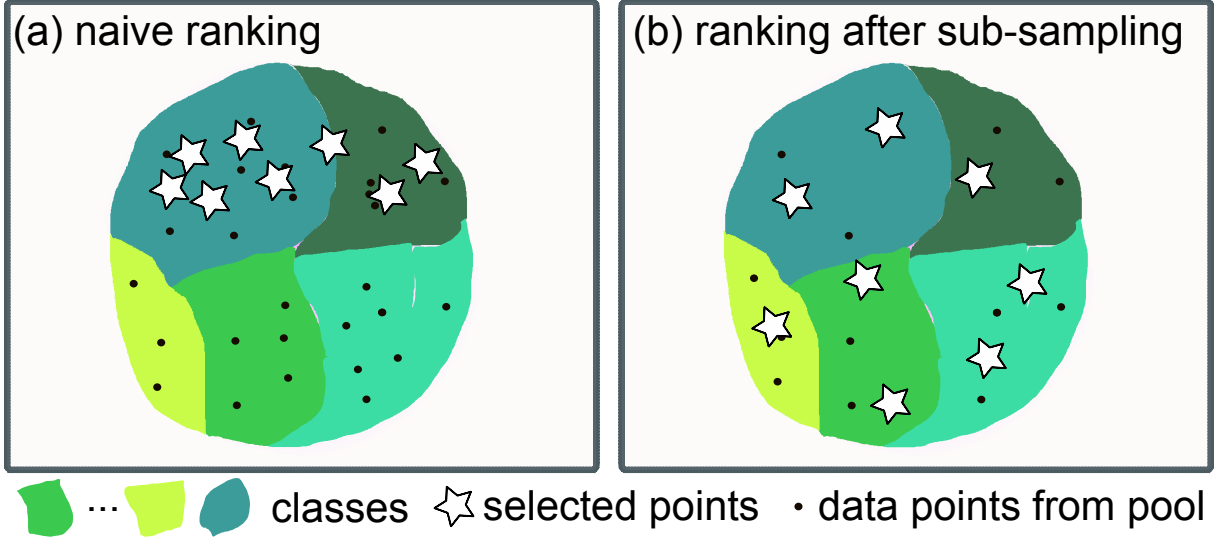


Figure 9.: Sub-sampling Strategy. We illustrate the ranking after the sub-sampling strategy. A naive ranking selects the most informative points from a few classes of pool data, while the ranking after sub-sampling enables to evenly select the most informative points across the variety of classes. This mitigates the class imbalance problem of AL for object detection, and introduced diversity can improve the performance.

In the naive TopN sampling, normally the simulation domain S and the real one R are assumed to be the same. This will further lead to performance degradation for both AL and object detection training [APH20]; [Oks+20]. In fact, the violation of this assumption motivates us to combine the TopN sampling with the popular sub-sampling technique [YM+10].

More specifically, selecting the B most informative images scored based on the model trained on S will result in an imbalance problem in the selected data set. Since the algorithm queries only images from real domain R , we attribute the under-performance during AL to the label distribution shift [PDS19].

To explain, we denote the distribution followed by sub-sampling as $P_{ss}(\mathbf{c}, \mathbf{b})$ and the distribution followed by uncertainty sampling as $P_{unc}(\mathcal{A}(\mathbf{c}, \mathbf{b}))$, which can be a product of delta distribution with probability mass placed at the top B scored predictions. Therefore, the selected data during AL follow the label distribution $P_{ss}P_{unc}$. Additionally, we use $P_r(\mathbf{c}, \mathbf{b})$ for the real label distribution, which is assumed to be uniform.

Our goal is to adapt the model with data points drawn from P_r , which is unavailable for unlabeled data. Instead, we adapt the model with data points drawn from $P_{ss}P_{unc}$, which ideally should be aligned with P_r . Unlike classification cases, in which the label distribution lies

in a discrete finite space and importance weighting correction [Zha+21] can be easily adapted, the label space for object detection is more complex when there is an additional regression task involved. The trade-off between alleviation of label distribution shift and utilization of information contained in the uncertainty estimates is thus determined by the distribution form of P_{ss} and the amount of data to be sub-sampled.

Intuitively (see Fig. 9), by assuming there is a certain degree of redundancy in the data set, we select the uniform distribution for P_{ss} , which works empirically well, as shown in the experiments. In practice, the pool set data is filtered by P_{ss} first, and then with P_{unc} , the learner thus can choose by considering the informativeness in the sub-sampled data. An illustrative explanation of the class imbalance problem, one instance of label distribution shift.

3.3.3. Summary of Results

In **Publication 3**, we first validate the proposed sampling strategy on a classification task, in which the model is transferred from MNIST [LC10] to MSNIST-M [Gan+16]. Then we move on to two *more challenging but task-relevant* self-collected data sets on *2D object detection*. To note that, we employ two data sets with different magnitudes of Sim-to-Real gap (one is large and the other small) to demonstrate that the proposed pipeline can efficiently bridge the gap for both cases. In all experiments, *we instantiate the Sim-to-Real gap by subtracting the performance of the corresponding models trained on purely the real and simulated data-set*. Nevertheless, we address the limitation of the proposed idea by including one *failure case* on the public YCBV data set [Xia+18] to further identify the operational scenario. In the end, we show the practical effectiveness of our idea by deploying the model on an assistive robot within a grasping task (more details in Chapter 5).

4. Introspective Methods for Robotic Assembly Sequence Planning

Assembly Sequence Planning refers to designing assembly plans – the order in which individual parts should be assembled, which is limited for certain-sized assemblies due to the NP-hard combinatorial problem and requires time-consuming feasibility checks. Data-driven Robotic Assemble Sequence Planning (RASP) based on DL is promising to improve the generalization and run-time efficiency. The resultant productivity enhancement is of paramount importance in the trend of shorter product life cycles and greater customization around the globe [Shi20]. However, the close-set limitation of the learning-based paradigm impedes the feasibility prediction in this setup. To put this in another way, the predictive model does not know whether the predicted plan for certain assemblies is feasible or not.

In this thesis, we attempt to tackle such problem by developing methods that are introspective, i.e./ being aware of the feasibility of the predicted plans. In this context we investigate this problem through the lens of two practical-relevant and representative scenarios, i.e. where infeasible assemblies are available in **Publication 4** and unavailable in **Publication 5**. In **Publication 4**, we first introduce a graphical data-driven approach for RASP and train it with both feasible and infeasible assembly plans, which yields decent performance and suggests an auspicious way to address the spatial embodiment challenge mentioned by [Sün+18]. In **Publication 5**, we study the case where infeasible assemblies are unavailable, which is quite common in the real world due to the risk of incomplete coverage of all possible infeasible cases and high time costs for generating sufficient infeasible training cases. Inspired by the success of NFs developed in the previous chapter, we exploit NFs for feasibility learning by reformulating it as OOD detection problem.

With the similar spirit to facilitate a concise and self-contained reading flow, we will provide a brief elaboration of the related work analysis and the method description followed by an outline of experimental results. For more details and full experimental results, we kindly refer the reader to the original manuscript of the corresponding publication in the appendix A.

4.1. Assembly Sequences Prediction via Graph Representations

In this work, we propose to use a graphical representation to faithfully describe the spatial structure of assemblies. Our so-called Assembly Graph is adapted from and more fine-grained than the one in [Rod+20] by representing the assembly as a heterogeneous graph whose edges denote geometrical relations between the assembly part surfaces. Based on this, we further develop a policy architecture based on GNNs, called **GRaph Assembly proCessing nEtworks**, for short GRACE, to extract useful information from the Assembly Graph and predict actions determining which parts should be assembled next. Apart from this, false predicted sequences and infeasible assemblies pose a severe problem for the efficiency of learning-based assembly robots, e.g., an incorrect sequence might require the robot to perform time-consuming re-planning. Therefore, it would be beneficial to detect these beforehand, e.g. being introspective against false predictions, hence we further develop and analyze various schemes to enhance the performance of feasibility prediction.

4.1.1. Related Work

Assembly Sequence Planning (ASP) A popular assembly graph representation for ASP is the AND/OR Graph [HS90], a formalism to encode the space of feasible assembly sequences, which can be created with the Disassembly For Assembly strategy [DW87]; [TBW03]; [Not+16a]; [TSR15]. However, these approaches are restricted in time to find a solution efficiently due to the feasibility checks. While graph search methods are impractical for larger assemblies because of the combinatorial explosion problem, heuristic intelligent search methods provide another alternative. They reject infeasible sequences and search for feasible ones close to the optimal based on manually designed termination criteria [Li+22a]; [IR16], learned [Che+08]; [SB05] or hand-crafted [RM17] energy functions. More recently, [Zha+19] and [WI20a] applied deep reinforcement learning for Assembly Sequence Planning. Targeting at RASP, [Rod+19]; [Rod+20] suggested inferring assembly rules (e.g. a specific part should be assembled before another), which can be transferred from previously identified sub-assemblies to those of larger sizes to prune the search space, thus reducing planning time. Their approach only produces rules, from which the final assembly sequences need to be derived additionally. It also requires further re-training when adapting to other product variants.

Graph Representation Learning in Task Planning In this setting, graphs commonly incorporate nodes for manipulated objects [Ngu+20]; [Bap+19]; [Zhu+21], their target positions [Lin+22]; [Fun+22] and the robot gripper [Ye+20]. Edges can represent high-level relations between objects [Ngu+20]; [Zhu+21]. With the graph representation, [Zhu+21] and [Ye+20] generated feasible candidate paths by sampling, and trained a network that predicts a

sequence of feasible actions in backward and forward search, respectively. [Ngu+20] performed sampling to find action sequences that transform the source to the target graph and then used optimization to eliminate invalid sequences subject to environmental constraints. Besides, some researchers resorted to RL methods such as [Bap+19]; [Fun+22], and [Li+20b], who used Graph Neural Networks for task planning. Recently, [Lin+22] utilized imitation learning to train two GNNs, one for selecting objects in the scene and another for picking a suitable goal state from a set of possible goal positions for long-horizon manipulation tasks.

4.1.2. Methodology

We describe the sequence prediction task for an assembly with N parts as Markov Decision Processes (MDP) [Bel57] with a discrete state space \mathcal{S} and a high-level discrete action space \mathcal{A} . Starting from state \mathbf{s}_t at time step t , executing action a_t produces a reward r_t and switches to state $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$ with a transition function p . State $\mathbf{s}_t \in \{0, 1\}^N$ is a binary vector indicating which parts are already placed in their target position by 1 (i.e. assembled) otherwise by 0. Action $a_t \in \{1, \dots, N\}$ represents the next part placement among the unplaced ones. For *feasible* assemblies, there are multiple different sequences leading to the final state, in which all N parts are placed correctly. For *infeasible* assemblies, no sequence exists, due to constraints of different aspects spanning from part geometries to kinematics and dynamics regarding the robotic system. Our objective is to learn a policy network $\pi_\theta(\mathbf{s}_t) = a_t$ parameterized by θ , which is optimized to imitate the assembly demonstrations $\tau_i = \{\mathbf{s}_{i,1}, a_{i,1}^{exp}, \dots, \mathbf{s}_{i,T}, a_{i,T}^{exp}\}$ in a data set of M sequences $\mathcal{D} = \{\tau_i\}_{i=1}^M$ and generalize across variants of different types and sizes at test time. In practice, our network predicts a set of multiple possible actions e.g. $K_t = \{a_{t,k}\}_{k=1}^{|K_t|}$ based on a tunable threshold to control the prediction quality.

Assembly Graphs

We represent the overall structure of an assembly with a heterogeneous graph. To make this representation agnostic to the rotation and mirroring of the assembly structure, we employ only relative distances instead of absolute positions for the features of edges between surfaces. More formally, given an assembly A (Fig. 10) at state \mathbf{s}_t it is modeled as a graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ containing two types of nodes: part nodes \mathcal{V}^p and surface nodes \mathcal{V}^s , and two types of edges: $\mathcal{E}^{s\text{-to-}s}$, connecting all surface nodes, and $\mathcal{E}^{s\text{-to-}p}$, connecting each surface node to its respective part. We detail each component as follows:

Part Nodes Responsible for encoding the current state of the assembly. A part node $v_i^p \in \mathcal{V}^p$ is associated with a feature vector $\phi(v_i^p) = [\text{assembled-flag} \in \{0, 1\}, \text{part-type} \in \mathbb{N}, \text{part-id} \in \mathbb{R}^d]$. There are three atomic part types: *long profile*, *short profile* and *angle bracket*.

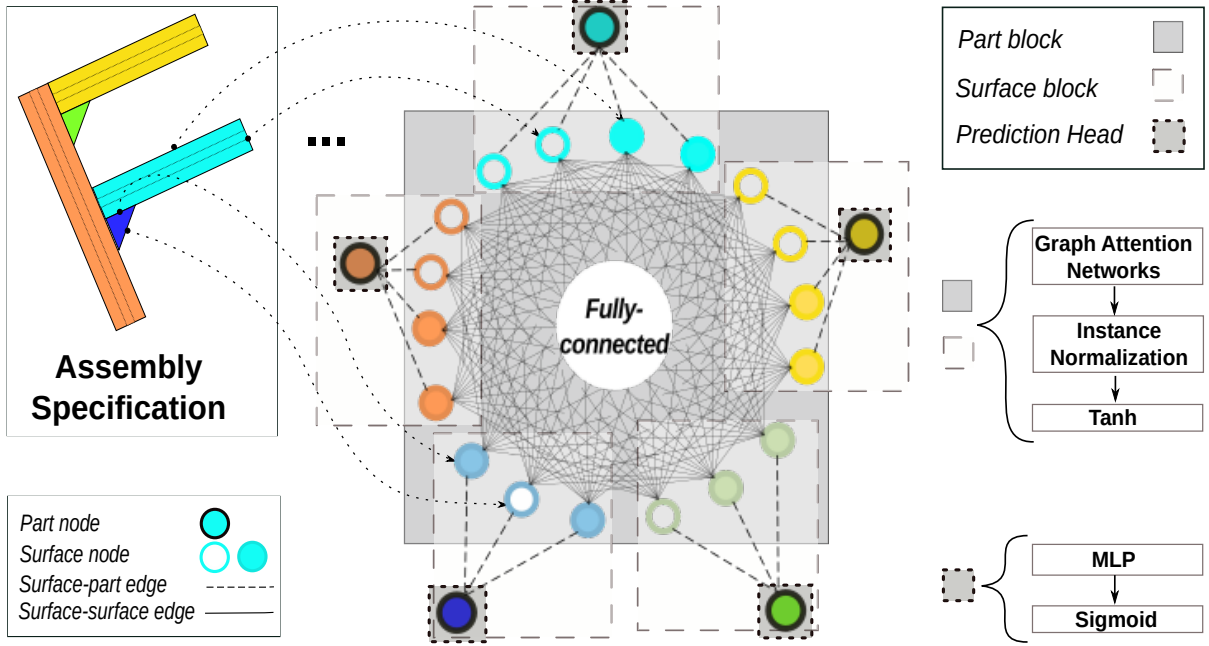


Figure 10.: Illustration of Assembly Graph and GRACE. Assembly Graph consists of edges connecting parts and their surfaces and edges among all part surfaces. In GRACE, the Part Block is shared for sub-graphs of part surfaces and the attached part, while the Surface Block is for the sub-graph of all part surfaces. To predict scores for parts to be assembled next, we apply a prediction head on each spare part.

Surface Nodes Different to the one in [Rod+20], we associate each surface node $v_i^s \in \mathcal{V}^s$ with the features $\phi(v_i^s) = [\text{surface-type} \in \mathbb{N}, \text{surface-id} \in \mathbb{R}^d]$. There are two surface types (*long* and *short*) for profiles and one (*lateral*) for brackets. Both the *part-id* and *surface-id* fields are encoded with a d -dimensional Sinusoidal Positional Encoding [Vas+17].

Surface-to-Surface Edges We design a fully-connected graph for all surface nodes \mathcal{V}^s to capture the relation between untouched surfaces, which is more fine-grained than those in [Rod+20] with only connects between touched surfaces. These edges are assigned with a feature $\phi(e_i) \in \mathbb{R}$, indicating the *relation* between the two surfaces: $\phi(e_i) = \text{relative distance}$ (parallel); 1 (belong to the same part); -1 (orthogonal); 0 (same-surface loop).

Surface-to-Part Edges These connect each surface and part node pair $(v_i^s, v_j^p) \in \mathcal{V}^s \times \mathcal{V}^p$, where surface v_i^s belongs to the part v_j^p . This type of edges is not associated with any feature vector.

Graph Assembly Processing Networks (GRACE)

Based on the formulation of a *step-by-step* sequential decision-making process per each part in the assembly, we introduce **GR**aph **A**ssembly **pr**o**C**essing **n**Etworks, for short GRACE,

$\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$, where $a_i = \{y_i | y_i \geq \lambda\}_{i=1}^N$, to extract useful information from the Assembly Graph and predict the next action given the current state of an assembly of N parts. $\lambda \geq 0$ is a threshold used to control the quality of predicted sequences. GRACE outputs a score per part $y_i \in [0, 1], i \in \{1, \dots, N\}$, reflecting the probability of placing the i -th part next. We further articulate the main components of this network (Fig. 10), describe the algorithm for predicting the entire sequence of length N by traversing predicted steps and the way we infer the feasibility of a given assembly.

Surface and Part Blocks The architecture is made of identical blocks, which are applied sequentially to obtain updated node features. Each block is made of a GAT [Vel+17], an Instance Normalization layer [UVL16] and a Tanh function. We choose GAT as it allows to utilize the rich semantics of edge features for updating node features in our graph representation. Surface Blocks are applied on surface nodes \mathcal{V}^s and surface-to-surface edges $\mathcal{E}^{s\text{-to-}s}$ for updating surface node features $\phi(v_i^s)$, while Part Blocks are applied on surface nodes \mathcal{V}^s , part nodes \mathcal{V}^p and surface-to-part edges $\mathcal{E}^{s\text{-to-}p}$ to update part node features $\phi(v_i^p)$.

Prediction Head and Loss Function To obtain a score per part, a fully-connected layer followed by a Sigmoid function is applied on each part node. During training, we minimize the loss between the network outputs and the ground-truth sequence steps from a data set of assembly sequences using binary cross-entropy. To note that, we apply this loss function for each part node separately. Our objective function (4.1) includes an additional regularization term (4.2), aiming at encouraging the network not to predict already placed parts:

$$L_\theta = \sum_{i=1}^M \sum_{j=1}^{N_i} (\hat{y}_{ij} \cdot \log(y_{ij}) + (1 - \hat{y}_{ij}) \log(1 - y_{ij})) + \delta L_{\text{reg}}, \quad (4.1)$$

$$L_{\text{reg}} = \sum_{i=1}^M \sum_{j=1}^{N_i} f_{ij} \cdot y_{ij}, \quad (4.2)$$

where M is the number of data examples in the data set, N_i is the number of nodes in the i -th graph. Abusing the notations, we denote y_{ij} and \hat{y}_{ij} the output score of the model π_θ and the ground-truth step in a sequence for the j -th node in the i -th graph respectively. δ is a weighing coefficient and f_{ij} the value of the *assembled-flag* in the input features.

Predicting Sequences As described, GRACE predicts a set of possible next steps based on the current state of an assembly. In order to generate a complete sequence (i.e. of length N), we repeatedly apply GRACE based on the current predicted state of the Assembly Graph. We devise an algorithm (Algo. 1) to traverse the assembly state tree using Depth-First-Search:

Algorithm 1 Assembly State Tree Traversal

```

function TRAVERSE-TREE(Model  $M$ , Assembly Graph  $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ , Threshold  $\lambda$ )
   $S \leftarrow \text{list}()$ 
  if ( $\forall v \in \mathcal{V} : v.\text{assembled-flag} == 1$ ) then
    return  $S$  ▷ Exit: all parts assembled
   $y \leftarrow M(\mathcal{G}_t)$ 
  for  $i \leftarrow 1$  to  $|\mathcal{V}|$  do
    if  $y[i] < \lambda$  then
      continue
       $\mathcal{G}_{t+1} \leftarrow \text{copy}(\mathcal{G}_t)$ 
       $[\mathcal{V}_{t+1}]_i.\text{assembled-flag} \leftarrow 1$  ▷ assembled node  $i$ 
       $S_* \leftarrow \text{TRAVERSE-TREE}(M, \mathcal{G}_{t+1}, \lambda)$ 
      for  $s$  in  $S_*$  do
         $s_* \leftarrow [i] + s$  ▷ Add current part to the sequence
         $S.\text{append}(s_*)$ 
  return  $S$ 

```

Starting with the graph in its initial state \mathcal{G}_0 – for all part nodes, *assembled-flags* are set to zero, the algorithm performs the following steps recursively: First, it checks the exit condition of the recursion – if all parts are already in place. Next, it predicts the probability for each part node y_i and picks those larger than the threshold λ , controlling the trade-off between precision and recall. Each of those nodes spawns a new branch individually. Therefore, we set the *assembled-flag* and call the recursion on the altered graph to retrieve possible sequences starting with the chosen node. Finally, we add the chosen nodes to the head of each returned sequence and return.

Feasibility Prediction To address the issue of miss-detecting infeasible assemblies, we develop two schemes to infer the feasibility of a given assembly: (1) We use the number of predicted complete sequences (output by Algo. 1) as an indicator for the feasibility of a given assembly. If no sequences were retrieved, the assembly is predicted as infeasible. (2) We aggregate the features of all part nodes from a pre-trained GRACE with a *mean-pooling* operation, creating a feature vector for the entire assembly graph. This feature vector is then used to train a binary classifier for feasibility prediction, where we analyze several classifiers i.e. Support Vector Machines (SVMs), Multi-layer Perceptrons (MLPs) and Nearest Neighbor.

4.1.3. Summary of Results

Through the experiments in **Publication 4**, we evaluate the Sequence Prediction under two experimental protocols with 4-fold cross-validation: (1) **intra-sized**: the assemblies in training and test set share *the same* sizes; (2) **inter-sized**: the assemblies in training and test set have *different* sizes, where there are two sub-protocols: Many-to-one and One-to-Many.

These experiments in simulation verify the capability of transferring knowledge between different assembly tasks, on which previous methods fall short. Further, our method can generalize knowledge gained on larger assemblies and then apply to smaller ones. Last but not least, it is worth mentioning, though only validated in simulation, our method should address the challenges during the real-world deployment like not finding a valid motion or a feasible grasping point if these cases are enclosed in the training data and learned to reject by GRACE.

4.2. Density-based Feasibility Learning

The goal of this part is to predict the feasibility of given assemblies relying only on feasible ones. We achieve this by formulating the problem as a density-based OOD detection using NFs (see Fig. 11). Given a data-set \mathcal{D} of N feature embeddings of feasible assemblies $\{\mathbf{a}_i\}_{i=1}^N$, where $\mathbf{a}_i \in \mathcal{R}^h$ is drawn from an unknown distribution $P_{feasible}$ with probability density function p_f , a density estimator, denoted by $q_\theta : \mathcal{R}^h \rightarrow \mathcal{R}$, approximates the true p_f with MLE for its parameters θ based on \mathcal{D} . During inference, given a threshold $\delta \in \mathcal{R}$, the feature of a test assembly $\hat{\mathbf{a}}_i$ is classified as OOD, i.e. infeasible, if $q_\theta(\hat{\mathbf{a}}_i) < \delta$, otherwise as ID, i.e. feasible.

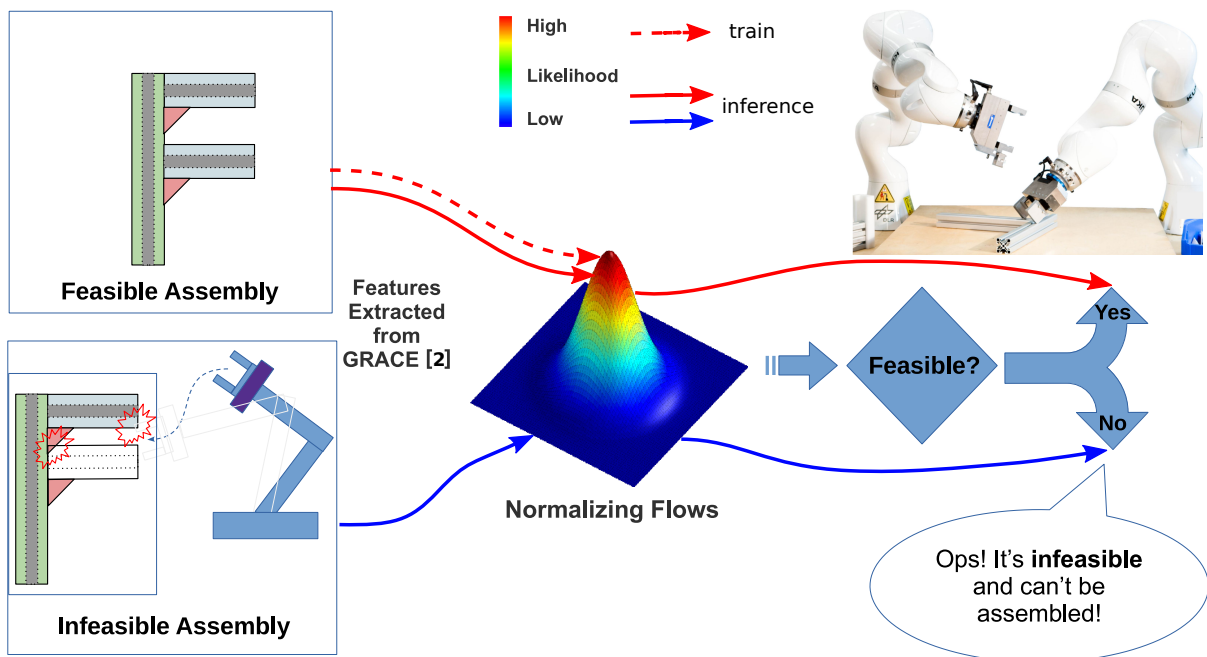


Figure 11.: Overview of the proposed method on an assembly scenario with a dual-armed robotic system (used in our setting). The distribution of feasible assemblies is modeled during training with NFs. In test time, infeasible assemblies are identified by their low-likelihood.

4.2.1. Related Work

Feasibility Learning The major body of work on feasibility learning is concentrated on plan or action feasibility learning in task and motion planning, while our goal is to learn the feasibility of assemblies directly by distilling the knowledge of assembly geometry and capability of the robot system. [Wel+19] trained a feature-based Support Vector Machine model to directly predict the feasibility of an action sequence based on experience, which is hard to scale to scenarios with different numbers and types of objects. [Dri+20] and a recent follow-up [Xu+22] predict if a mixed-integer program can find a feasible motion for a required action based on visual input. Besides, [YGF22] predict a plan’s feasibility with a transformer-based architecture using multi-model input embeddings. Different from us, these methods work in a two-class setting, requiring failing action sequences to be included in the training set and then use binary feasibility classifiers.

4.2.2. Methodology

Density-based Feasibility Learning with NFs

In this work, NFs are used to estimate the density of feasible assemblies. NFs, denoted by $f_\theta : \mathcal{R}^h \rightarrow \mathcal{R}^h$, are defined by a chain of *diffeomorphisms* (invertible and differentiable mappings) that transform a base distribution $p(\mathbf{z})$, $\mathbf{z} \in \mathcal{R}^h$ (e.g. an isotropic Gaussian) to the data distribution q_θ (in our case p_f). Based on the Change-of-Variables formula, the likelihood of an embedding of an assembly is obtained by

$$q_\theta(\mathbf{a}) = p(f_\theta^{-1}(\mathbf{a})) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \quad (4.3)$$

θ is optimized with MLE based on feasible data only, where the log likelihood is defined as:

$$\log q_\theta(\mathbf{a}) = \log p(f_\theta^{-1}(\mathbf{a})) + \log \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \quad (4.4)$$

To this end, the inverse flow f^{-1} and the log determinant of the Jacobian need to be tractable and efficient. We employ the Real-NVP [DSB16] that is composed of multiple layers of affine coupling flows. As the input to the NFs, a data-set of feature embeddings for feasible assemblies \mathcal{D} is extracted from a pre-trained GRACE, which represents each assembly structure as a graph of its parts and their respective surfaces. To create a single feature embedding per assembly, a channel-wise mean pooling is applied on the graph’s part nodes. Different to previous works, the dimension of this embedding is independent of the number of assembly parts.

During inference, given a test assembly embedding, the trained NFs q_θ predicts a log-likelihood score and determines its feasibility based on a pre-defined threshold δ , which we selected with a validation set.

4.2.3. Summary of Results

In the experiments conducted in **Publication 5**, we pre-trained GRACE proposed in **Publication 4** with its default parameters to retrieve a 94-dimensions embedding per assembly and implemented the NFs model using [Sti+23] and experimented with Gaussian and Resampling [SSH22] base distributions. The NFs model with Gaussian base distribution achieves the highest score with a deep 749-layered network, outperforming the One-class Support Vector Machine [Sch+99] and the naive GRACE. In this setting, GRACE, trained on *feasible assemblies only*, predicts an assembly sequence for a test instance and infer the assembly’s feasibility based on the success of its sequencing process. More practically relevant, the NFs variant with the more expressive Resampling base distribution [SSH22] can reach comparably good results with a much smaller network (109 vs. 749 layers). This benefit of memory efficiency is highly relevant for robotic systems with only restricted computation resources (e.g., mobile manipulators). Contrary to GRACE’s sequencing process, we only require a single-pass through the feature-extraction pipeline, independent of the size of the assembly, and could therefore determine the feasibility of multiple batched assemblies at once.

5. Applications and Discussions

Confronted with the indispensable consideration of safety and trustworthiness in DL applied to robotic perception and planning, this thesis is dedicated to tackling such challenges by developing introspective methods. These methods are designed to address the challenges step-by-step along the learning axis (see Fig. 3), i.e. delivering reliable uncertainty estimates, identifying OOD data, and enabling robots to learn actively. Rather than verifying the proposed ideas on the benchmark data sets, we further highlight the practicality by deploying them on robotic systems in the real world.

In this chapter, we first introduce the demonstrations of deploying the proposed introspective methods on two robots that function in different applications, namely caregiving and factory inspection and maintenance. Then, we summarize the main contributions of the proposed techniques articulated in previous chapters. In the end, we conclude the thesis with a discussion about the limitations of the methods and their promising future directions.

5.1. Applications

5.1.1. Introduction of Robotic Systems

EMG-controlled Daily AssistaNt (EDAN)

The robotic wheelchair EDAN (EMG-controlled daily assistant) [Vog+20b] is a fully integrated wheelchair-based manipulation assistance for people with severe motor impairment. It can be controlled by a joystick or via electromyographic (EMG) [VH18] signals and is designed to perform activities of daily living supported by shared control capabilities [Que+20] in combination with whole-body impedance control [Isk+19]. More noteworthy, when teaming up with the humanoid assistance robot Justin [Hul+08] and the haptic teleoperation device HUG [Bor+09], EDAN helps establish a holistic ecosystem for robotic assistants in caregiving to tackle the challenges due to demographic changes, e.g. labor shortage of caregivers [Vog+20a].

EDAN is composed of a mobile base (a commercial power wheelchair) and a torque-controlled robotic arm DLR LWR-III [Bis+10] mounted on the right side of the wheelchair (see Fig. 12). The mobile base provides a front-wheel drive and pivot-rear-wheels, re-enabling the mobile

ability. For manipulation, the integrated DLR Light-Weight Robot III (LWR3) is enhanced with an additional (8th) axis at the arms base, significantly extending the reachability of the manipulator. In particular, a dexterous torque-controlled five-fingered DLR-HIT hand for grasping and manipulation is used. Additionally, a pair of stereo cameras is mounted on an actuated pan-tilt unit to allow for remote teleoperation. To provide the user with necessary and task-dependent information, a tablet is attached to the wheelchair.

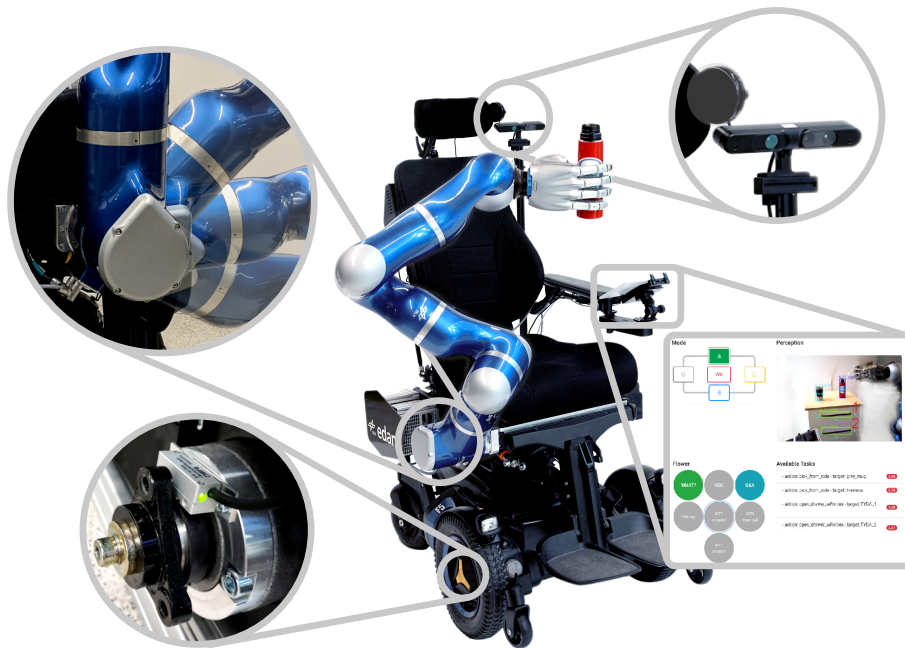


Figure 12.: EDAN system includes a closeup of the upgraded wheel-encoders (bottom left), the range of motion of the additional, eighth axis of the DLR LWR-III (top-left), the head-switch and the RGB-D Camera (top right) and the tablet interface (bottom right).

For perception, the system is equipped with an RGB-D camera to perceive the environment in order to assist the fine-grained object manipulation based on shared-control [Que+20]. The perception pipeline consists of two stages, i.e. object detection and pose estimation (see Fig. Fig. 13). To assist users in manipulating their environment, objects have to be identified and localized from the surroundings perceived by the robot. Object Detection is performed given a set of known classes from the RGB image data with a DL-based object detector, namely a fine-tuned RetinaNet with a ResNet 50 backbone and weights pre-trained based on the COCO dataset [Lin+14]. Once an object has been detected and its class identified, the cropped depth data from the bounding box detection is converted into point cloud data., whose pose is then estimated by the Iterative Closest Points (ICP) algorithm or by plane estimation with RANSAC. This perception pipeline can be deployed on an embedding system such as a NVIDIA Jetson TX2 or a workstation PC, the predictions (i.e. pose estimates of the detected objects) are then sent to the shared-control module via Links and Nodes (LN) middle-ware.

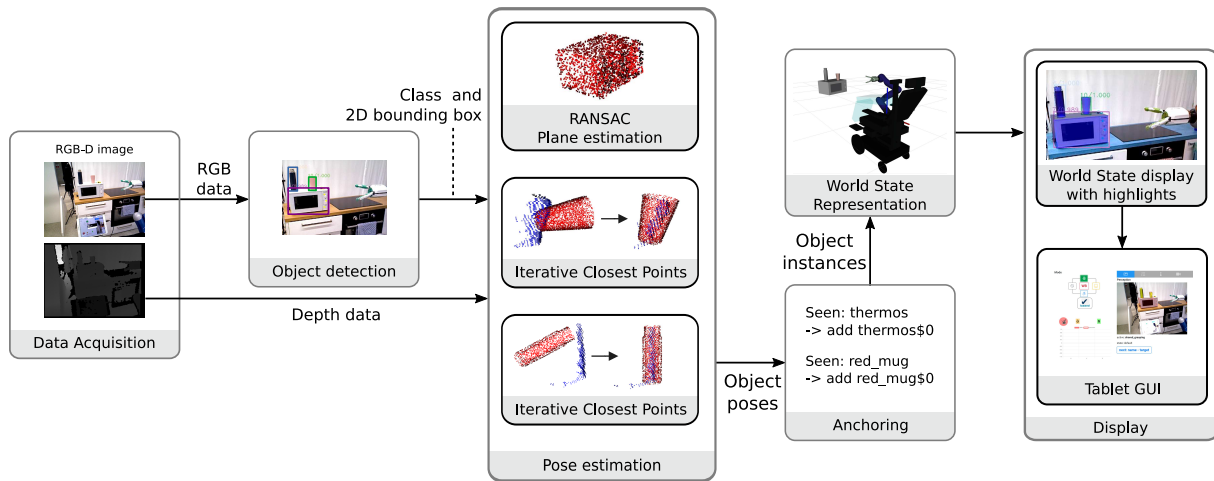


Figure 13.: The perception pipeline on the EDAN system.

EDAN has demonstrated its core functionalities such as a shared-control-based manipulation aid at the leading exhibition for smart automation and robotics, Aoutomatica 2022 (see Fig. 14a)¹. There, a user which can be people with motor disability sitting on the chair intends to control the robot arm for tasks like pouring by using an input device (EMG signal sensors or a spacemouse with lower degrees of freedom (DoFs) than that of the end effector (3 vs. 6). The mis-correspondence of DoFs between the input device and the manipulator demands that the user needs to tediously switch input mapping between them for task completion in a pure manual control mode. For example, even with a 6 DoF force feedback device, Lambda as in Fig. 14a, the high cognitive workload e.g., switching between translation and rotation, is required to complete simple tasks like pouring water. In this case, shared control is able to ease the execution of such daily living tasks based on the task-specific share control template [Que+20]. Furthermore, in CYBATHLON Challenges 2023, with our pilot Mattias Atzenhofer, the DLR's EDAN team won first place in the race of assistant robots for people with severe impairment of arms and legs (see Fig. 14b)². The CYBATHLON is a non-profit project of ETH Zurich that challenges teams of developers around the world to develop assistance technologies suitable for everyday use together with and for people with physical disabilities.

cable-Suspended Aerial Manipulator (SAM)

DLRs' SAM [Sar+19a] is a novel aerial manipulation system for inspection and maintenance applications. The envisioned application is the bilateral teleoperation concept, i.e. a human operator remotely controls the robotic manipulator located remotely in dynamic and unstructured environments, from a safe area on the ground. At the same time, the teleoperator can receive visual and haptic feedback from the robot.

¹Press for AUTOMATICA Exhibition 2022

²Video Link of EDAN Cybathlon Challenge 2023



Figure 14.: Images of EDAN demonstration at AUTOMATICA Exhibition 2022, where the author of this thesis was instructing a visitor how to complete the task with Lambda based on shared control (a) and Cyathlon Challenges 2023, in which our pilot Matthias Atzenhofer was trying place the disk onto the black IKEA shelf (b). Source: Courtesy of DLR.

SAM is composed of three modules, namely a carrier, a cable-suspended platform and a 7 DoF industrial robotic arm - KUKA LWR [Bis+10]. The carrier transports the manipulation system to a desired location. For instance, a crane or a helicopter can be used, depending on the requirement of safety, versatility, robustness and applicability for the considered industrial scenario. Then, the cable-suspended platform is attached to the carrier via a rope. This design can autonomously damp out the disturbances induced by the carrier, the environment, and the manipulator. This oscillation damping control is performed using eight propellers and three winches. Another important component of our system is the seven-DoF torque-controlled KUKA LWR [Bis+10], which is significantly more powerful and offers more versatile manipulation capabilities than many existing smaller manipulators. The main advantage of this concept is that the carrier supports the weight of SAM, which reduces the required energy to carry the robot arm.

To teleoperate the robot arm on SAM, two haptic devices, namely a space-qualified haptic device called the Space Joystick RJo, and also a 6-DoF force feedback device, Lambda (Force Dimension) have been integrated. For the perception module, an eye-to-hand camera (mako), an eye-in-hand stereo camera and a commercial 3D vision sensor Rcvisard that provides built-in visual-inertial SLAM have been integrated. For 3D object pose estimation in outdoor environments, a Velodyne PUK-LITE lidar is also installed on the frame of SAM, which provides 3D point clouds of the scene at 10Hz.

Most excitingly, the proposal based on the SAM system has been selected as one of three finalists of the Kuka Innovation Award 2023 around the globe. There, SAM needs to be teleoperated to finish two tasks in the common factory inspection and maintenance scenarios, namely, rotating a valve and pick-and-place of an inspection robot on the pipe (see Fig. 16)

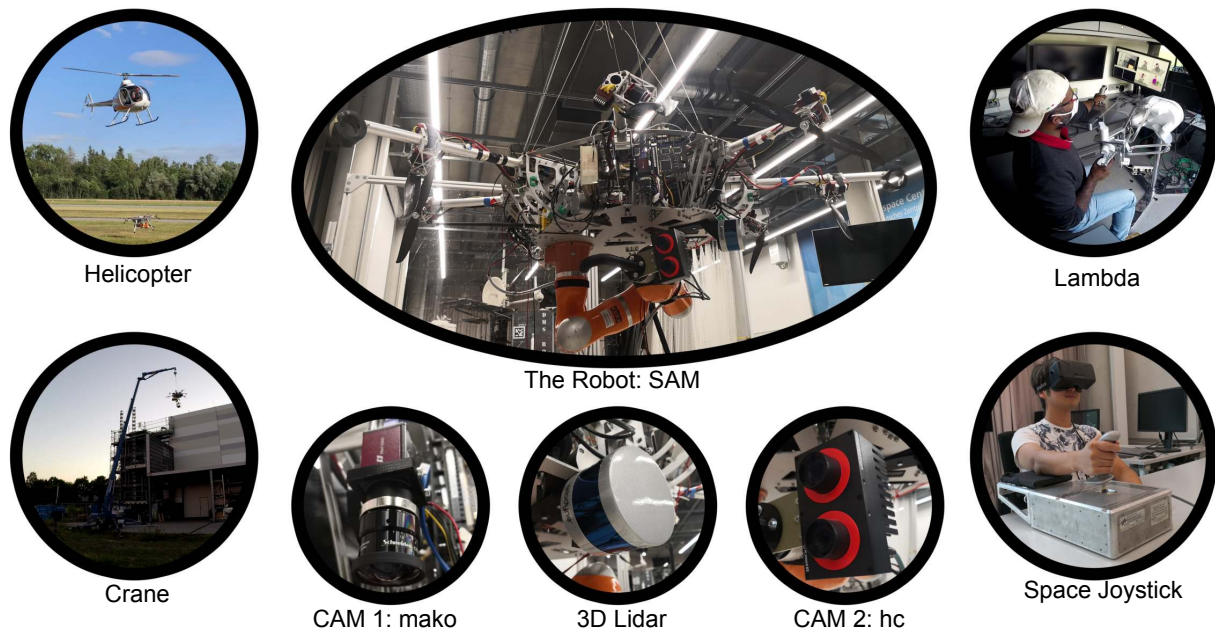


Figure 15.: System description of SAM. Left: the concept involves the carriers such as a manned helicopter or a crane, which transports SAM to a desired location. Middle: SAM is equipped with a stereo camera at the end effector or the manipulator, a monocular camera as well as a lidar on the flying platform. Right: haptic interfaces are integrated for teleoperating the robotic arm [Lee+23].

^{3 4}. The KUKA Innovation Award is a competition that KUKA announces every year under a different motto. The competition aims to accelerate the pace of innovation in the field of robotic automation and improve technology transfer from research to industry.

Small and Medium Enterprise (SME) Robotic Assembly System

The SME [Rod+19] system is composed of two KUKA LBR iiwa arms, each with a Schunk WSG 50 gripper (see Fig. 17). The system is designed to solve the assembly tasks with various robotic skills available through a skill library [Not+16b]. The library consists of general skills and a number of specialized skills for the given use case. For example, the *PickUpObject* skill can be used for picking parts of various types, whereas the skill *MoveSlotNut* is used only for the positioning of slot nuts within profiles. The force-based contact detection and the impedance controller of the robot can increase the robustness with respect to position uncertainties. The skills can be parameterized for the given assembly tasks and are connected to a world representation and an object database for symbol grounding [LBH12]. When given a target assembly, an assembly planner is responsible for decomposing the target into a series of sub-tasks. Then, a task classification approach [Not+16b] is used for the mapping of these sub-tasks to the robotic actions or skills. It is crucial to note the difference between (sub-)tasks and skills. Tasks refer to descriptions of the objects to be manipulated, that is independent

³Press for Kuka Innovation Award 2023.

⁴Video Link for Kuka Innovation Award 2023.

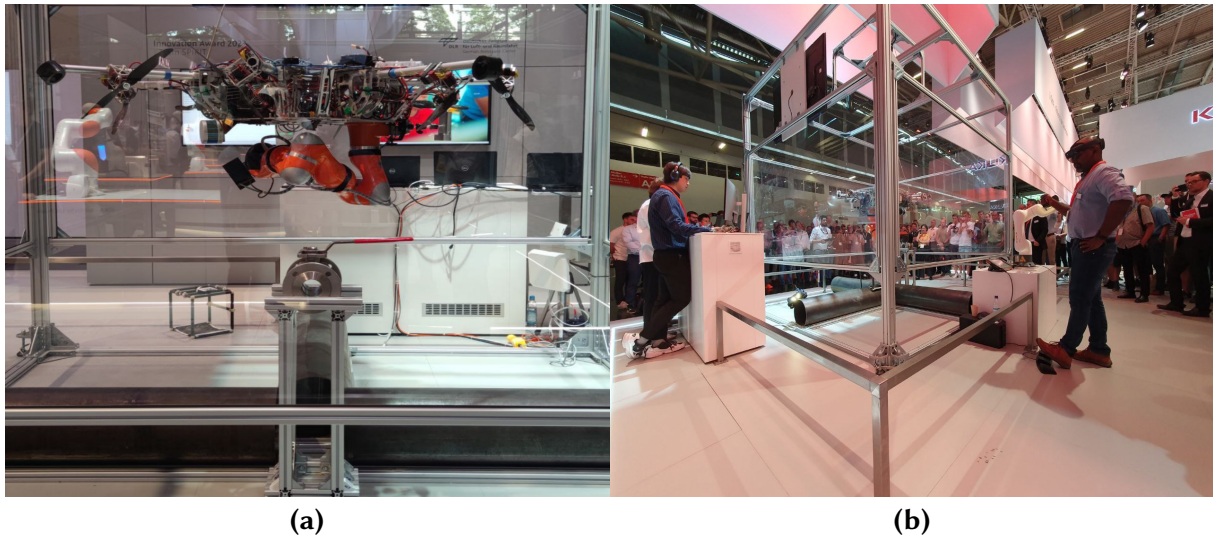


Figure 16.: The setup of SAM for Kuka Innovation Award 2023 (a) and the demo performed during the competition at AUTOMATICA Exhibition 2023, where SAM is placed in transparent cage, in which there is a mocked scenario for inspection and maintenance in the factory. The teleoperator on the right was using the kuka iisy robot arm as the input device to control the arm on SAM (b).

of the robotic system. The skills are the actions to achieve the corresponding tasks, which are highly correlated to the specific robotic system. Additionally, a workspace analyzer and a motion planning module are important because now not all the tasks are executed in the same place. These modules can be used for motion generation within different robotic skills. The robotic assembly system is capable of solving all required sub-tasks: 1) insert slot nuts in the profiles. 2) position profiles. 3) add angle bracket. 4) add screws.

In Chapter 4, we mainly focus on the assembly planner module and propose learning-based methods to increase efficiency and generalizability, which paves the way to the introspective methods. However, even though the accuracy of predicting a feasible assembly sequence is more than satisfactory, we have not tested it on the real robot due to system maintenance. Therefore, we leave this in the future work. At the same time, we are also aware that our method is trained on the data generated in simulation, a simulation can not take into account all of the issues that an actual execution may encounter (e.g., unexpected interactions with the environment, and imperfections in the mechanics of the robot). There might be a sim-to-real gap when we want to evaluate the method on the real robot later.

5.1.2. Saving Annotation Efforts for Real Robot Perception

In this part, we introduce the demonstration of deploying the proposed Bayesian Active Learning pipeline described in Chapter 3.1 on an assistive robot EDAN [Vog+20b] and an Aerial Manipulator SAM [Lee+23].

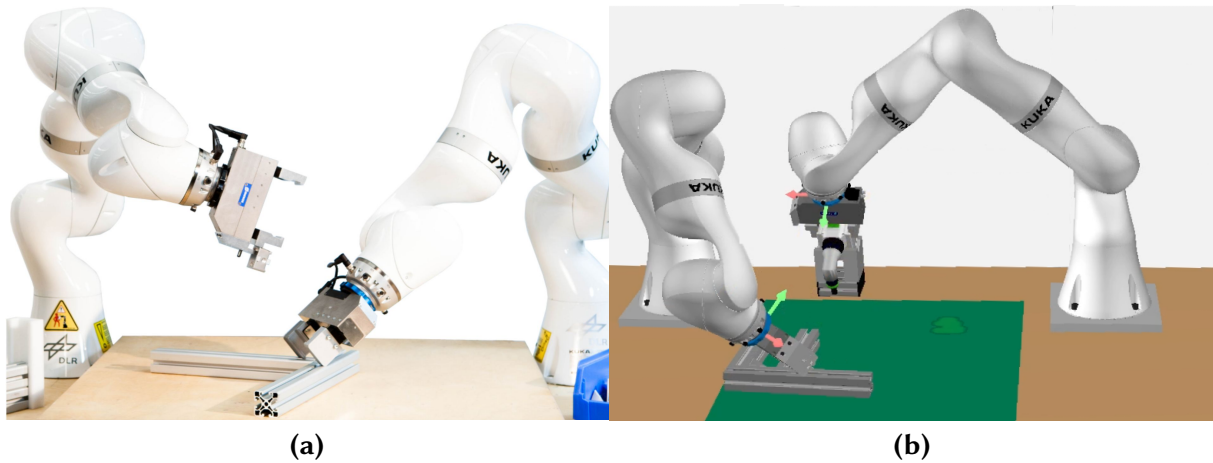
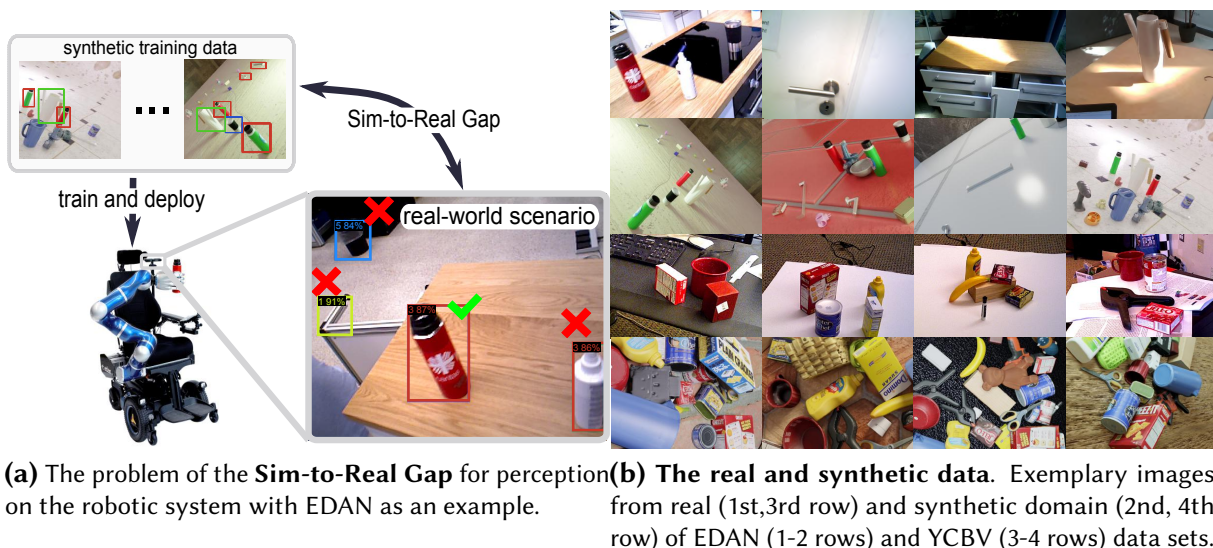


Figure 17.: The setup of SME in the real world (a) and simulation (b).

Task Description On account of the working scenarios (e.g. caregiving or factories) for an assistive robot such as EDAN or an aerial manipulator such as SAM, a variety of objects need to be detected and the manual efforts required for adaptation must be kept *as minimum as possible*. We first attempted to replace real images of the training data – which require massive manual labeling – with synthetic counterparts to reduce the annotation efforts in preparing the training data set. Recent advances in realistic image synthesizers realize cost-efficient synthetic image generation [Den+19] in combination with existing 3D object models (see Fig. 18b). However, having observed the sub-optimal performance due to the sim-to-real gap (see Fig. 18a), we further take an initial step to devise a novel pipeline with Bayesian active learning to bridge the gap as proposed in **Publication 3** and described in Chapter 3.



Demonstration To demonstrate the practical effectiveness of the proposed idea, we evaluate the method on two real robotic systems i.e. EDAN and SAM.

EDAN: With the proposed active learning pipeline, we show that the object detector can be more accurate with fewer annotation efforts and at the same time yield better poses estimation results. This can greatly improve the success rate of the task execution based on shared control. We evaluate three tasks, i.e. pouring, drawer opening and water can grasping (see Fig. 18). We also provide a video to showcase the deployment⁵.

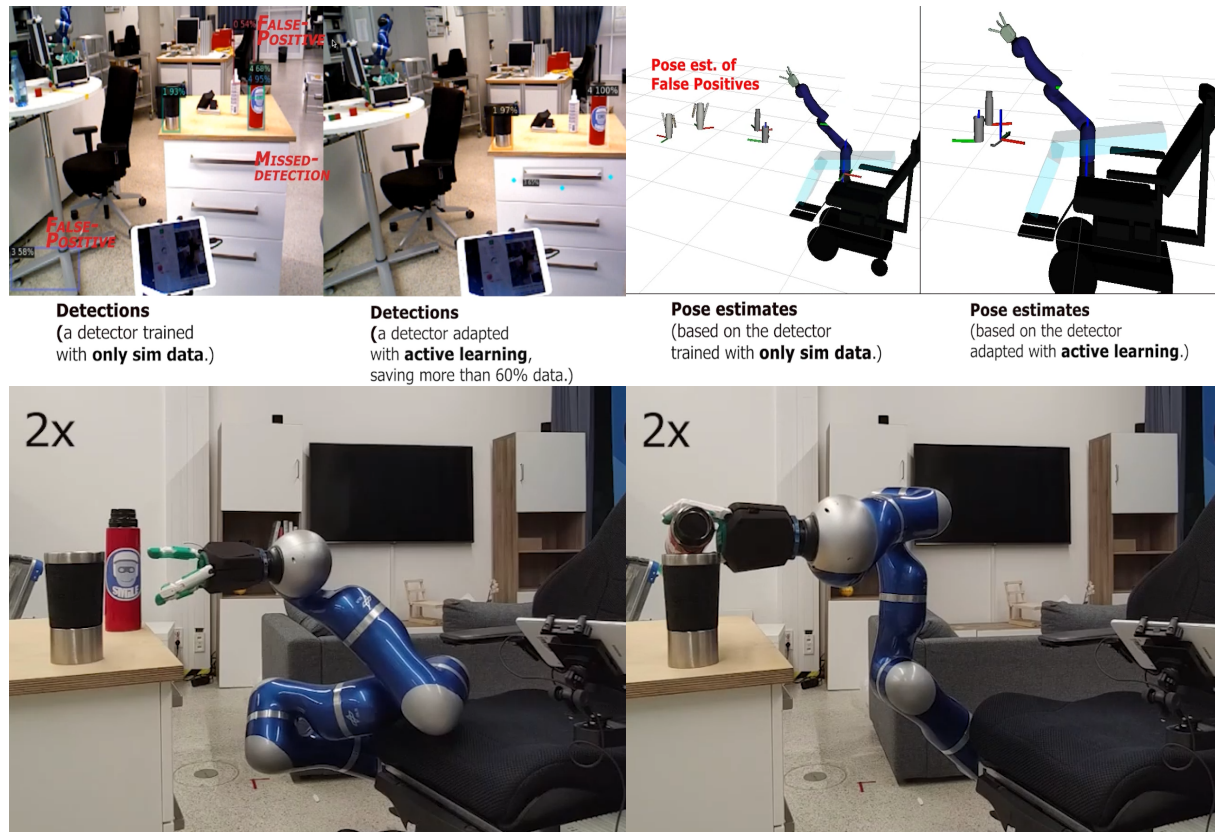


Figure 18: Exemplary screenshots of a pouring task via shared control on EDAN. The two screenshots on the top show the performance of the detector and the corresponding pose estimates (visualized in Rviz) before (left in each column) and after (right in each column) adaptation via the proposed pipeline. The two screenshots at the bottom show the sequence of a grasping and pouring task execution with shared control.

SAM: Assuming a robotic manipulation task far away from the human operator, who does not have direct visual contact with the scene. For this, SAM creates a virtual reality of its environment and workspace using onboard sensing and computations and further provides haptic guidance via virtual fixtures. The images are obtained either from the eye-in-hand stereo camera or a monocular camera at the base. The depth data is acquired by a lidar at the base. A SLAM system at the end effector estimates the transformation between the coordinate

⁵Demo Video Link of Bayesian Active Learning on EDAN.

camera frame and the world frame. The task is to obtain the 6D pose of the objects, which resembles the perception pipeline on EDAN. Therefore, we apply the proposed active learning framework in order to save the annotation workload for the object detector (see Fig. 19). We also provide a video for this demo⁶.

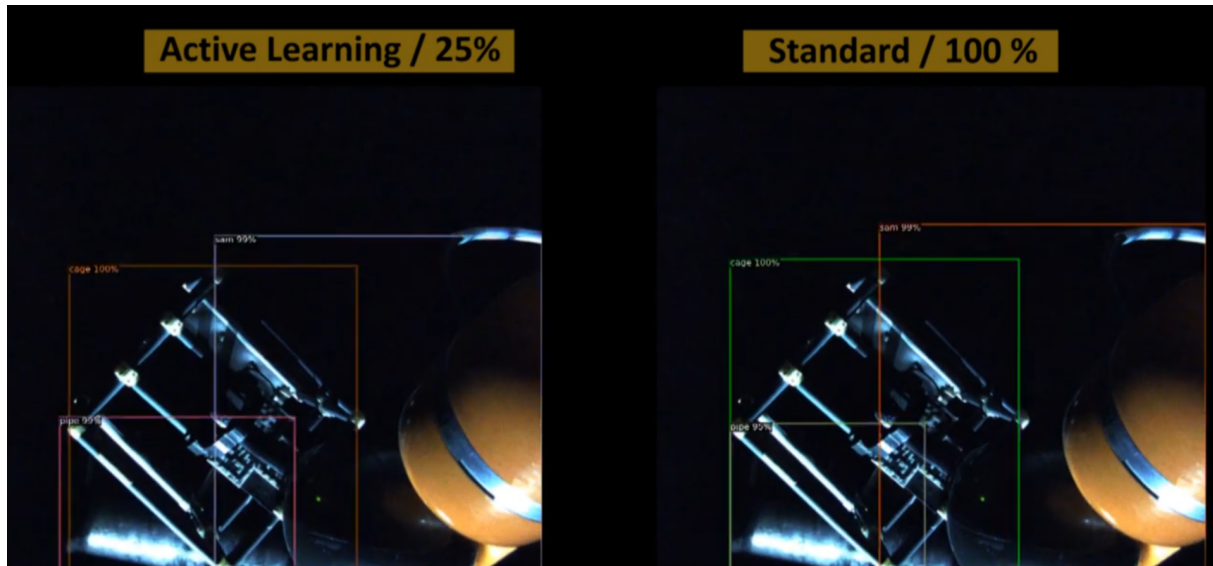


Figure 19.: Exemplary screenshots of Active Learning for Object Detection on SAM. This screenshot shows the difference between a detector trained with AL and without. We can see that the labeling workload can be saved up to 75% while achieving similar performance.

5.1.3. Detecting Out-of-Distribution Objects for Inspection and Maintenance

In this part, we showcase the demonstration of deploying the OOD detector introduced in Chapter 3.2 on SAM in the application scenario of inspection and maintenance in factories.

Task Description The perception system has to understand its surroundings semantically and DL-based methods are the current golden standards. Unfortunately, learning-based methods often assume that the test samples are generated from the same distribution as the training data. This assumption is routinely violated in the real world, and out-of-distribution detectors aim to identify such failure cases of learning-based methods.

In this work, since the semantics of the scenes may rely on vision as its main modalities, we utilize one monocular camera to detect the objects of interest, which are an industrial valve, and an inspection robotic crawler for oil and gas pipes in refineries. For computing, the robot is equipped with two NVIDIA Jetson Orin. In the experiments adapted, the real images were captured in a mock-up facility, and tested with the NVIDIA Jetson ORIN on the robot. The

⁶Demo Video Link of Bayesian Active Learning on SAM.

experimental data were collected with 30W mode with JETPACK 5.1.1. Auvideo carrier board is used.

For the object detection, we used the open-sourced implementation ⁷ for training and testing the object detector yolov7 [WBL23]. The detector was then deployed on the embedded computing module NVIDIA Jetson Orin on our aerial manipulation robot.

Demonstration In this demonstration, we validate the applicability in an application of robotic inspection and maintenance, where it is crucial to avoid false positives of OOD objects that appear routinely in outdoor environments. We train a Yolov7 object detector with only synthetic images of two objects (a valve and a crawler robot) and deploy on the robot around only real objects. The task is to identify the falsely detected real objects as OOD since they are from a distribution different to the synthetic ones. The objects from the real data distribution serve as OOD data. After being trained on the data from the simulation data distribution, we wish our OOD detector to distinguish such challenging OOD data due to their semantic closeness. A video is recorded for this demonstration⁸.

5.2. Conclusions

In order to enhance the reliability of learning-based perception and assembly sequence planning in robotics, we addressed the challenges [Sün+18] that impede the adoption of these approaches with the proposed introspective methods. Rooted in the reminiscence of robotic introspection, we have studied how to facilitate two main capabilities: 1. how to attain introspection and 2. how to exploit introspection for challenging tasks such as AL.

The introduced techniques leverage probabilistic machine learning models to be capable of providing introspection for their outputs (i.e. uncertainty estimation and OOD detection) and effectively making use of them for AL. Being capable of providing introspection can resolve the challenges of overconfident predictions and vulnerability against OOD data, paving the way toward robust failure detection or catastrophic consequence avoidance. These remedies are particularly favorable for a trustable robot deployment in an unpredictable and outside-the-lab environment, such as an unknown factory or streets in a small countryside. With them, we showed that robotic perception including object classification and detection can be more reliable and resistant to silent failures and OOD objects. More noteworthy, we also demonstrate that robotic assembly feasibility learning can be expressed as an OOD detection problem so that it is able to detect what it cannot assemble with the proposed technique. On the other hand, the method that can utilize introspection for AL addresses the challenge of

⁷<https://github.com/WongKinYiu/yolov7>

⁸Video Link for Normalizing Flows based OOD detection on SAM.

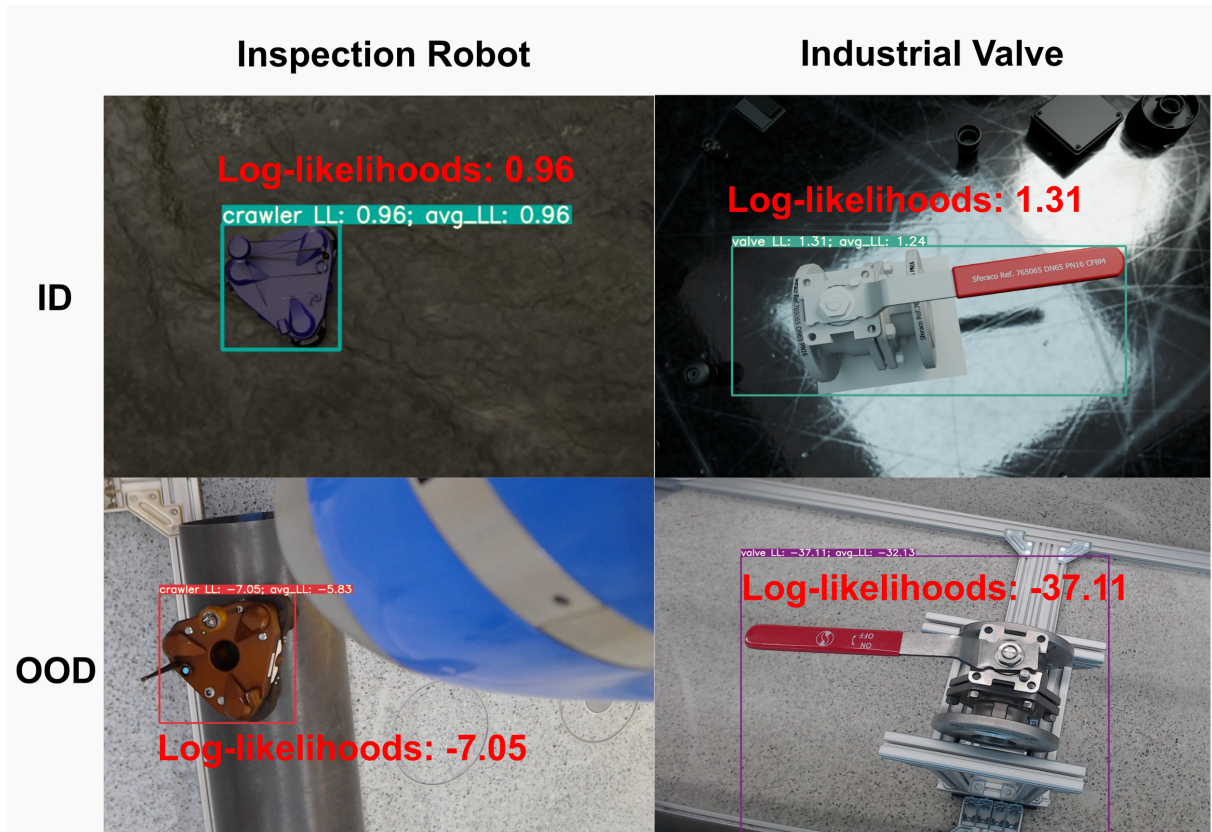


Figure 20.: Exemplar screenshots of the proposed OOD-aware object detector deployment on the aerial manipulator SAM. The proposed OOD detector can output high likelihoods for ID data, i.e. the synthetic inspection robot and valve and low likelihoods for OOD data, i.e. the real inspection robot and valve.

facilitating more autonomous and independent learning capability. This method is shown to effectively save the laborious and time-consuming data annotation workload on a real robot.

During the journey of actualizing the ideas mentioned above, we also harvest some bitter lessons learned from the development of the introduced introspective methods. We would like to share them in this chapter and describe the limitations and future directions in detail.

5.3. Limitations

The proposed methods attempt to push forward in the direction of equipping the robot with introspective capability, but there are some remaining limitations to be addressed from both the technical and methodological aspects, which are explained as follows.

Trade-offs of Run-time

In **Publication 1** and **Publication 3**, uncertainty estimation based on BNNs in this thesis is not real-time capable as it requires repeated forward passes to generate samples. Although this is sufficient for AL due to the relatively longer time for re-training the model, it is less useful for general robotic perception scenarios such as a dynamic and unpredictable environment, where the robot needs to perceive the surroundings in a high frame-rate.

Likewise, in **Publication 2** and **Publication 5**, although NFs can function in real-time without much burden on the run-time efficiency. There is a remaining limitation during the initialization, which is the prolonged initialization time for calculating the normalization factor in LARS based on Monte Carlo sampling. This might not be desirable for applications that require instant response at the beginning.

Feature Embeddings Quality for NFs

The proposed method is envisioned to work on feature embeddings instead of raw data to counteract the NFs artifacts of assigning higher likelihoods to OOD data [Pap+21].

This leads to two limitations. First, it's can't directly applied to the tasks/models that could not provide useful feature embeddings extracted from the raw data. The second is that its performance is restricted to the quality of the features. As reported by previous work [Mil+21]; [Li+22b], learning more compact and centralized features can often lead to increased performance for OOD detection while feature collapse can be harmful to OOD detection.

5.4. Future Directions

Merging Merits of BNNs and NFs

In this thesis we have explored two probabilistic machine models, i.e. BNNs and NFs for developing introspective methods. While BNNs excels at estimating accurate uncertainty and suffers from resource-intensive probabilistic inference, NFs are shown to be more effective than BNNs on OOD detection without much burden on run-time latency. One interesting future direction is to develop methods that can combine the best of both worlds. More noteworthy, as uncertainty estimates from BNNs can advance active learning, the information/introspection provided by NFs should be capable of doing a similar job.

Extending Introspection to Other Modules on the Robot

In this thesis, we only investigate introspective methods for robot perception and assembly sequence planning. When designing the methods, we kept in mind that the methodology needs to be as general as possible so that it can be adapted to other learning-based components such as data-driven control or task and motion planning. Therefore, it would be meaningful to extend the concept of introspection to other sub-modules on the robot, paving the way to a trustworthy data-driven robotic system.

Developing Actual Introspection

In this thesis, we interpret introspection as the ability to achieve self-understanding, knowing the limitations of the acquired knowledge and what the robot does not know. This interpretation motivates us to develop methods for uncertainty estimation, OOD detection and uncertainty-based AL. These methods can establish introspection at the early stage, which has already been shown to enhance the safety and reliability of the robot or the algorithm. However, a fully introspective method should enable the robot to shape self-awareness via a chain of mental actions. The methods proposed in this thesis attempt to provide such self-awareness directly without the reasoning from a chain of thoughts. Considering this, one interesting future direction would be to develop a "real" introspective method that can go through a sequence of mental actions. The recent proposed Chain-of-Thoughts prompting in Large Language Model [Wei+22b] seems to provide an appealing way to achieve this goal.

6. Summary of Publications

This chapter presents one-page summarisation for each publication upon which the dissertation is based:

- Introspective methods for robotic perception:
 - **Publication 1** presents an uncertainty estimation method for introspective robotic perception and demonstrates its benefits in semantic reasoning and uncertainty-based selective classification.
 - **Publication 2** tackles the challenge of deploying perception module such as an object detector in an open-world with Out-of-distribution detection method.
 - **Publication 3** addresses the problem on how to exploit introspection for an object detector to reduce manual annotation efforts by actively soliciting the most informative data.
- Introspective methods for robotic assembly planning:
 - **Publication 4** introduces a data-driven method for predicting the robotic assembly sequences based on graph representation learning and approach the feasibility learning problem by including infeasible assemblies.
 - **Publication 5** (Pre-print) proposes to learn the feasibility of the predicted sequences via a density-based approach, facilitating introspection in learning-based robotic assembly planning.

For each publication, the reference and abstract are provided followed by the description of the individual contributions made by the author of the dissertation. In particular, for clarity and completeness, the contributions of all authors of the publication are listed using the roles defined by CRediT¹ besides the textual explanation. Further contributions of other persons mentioned in the acknowledgments are listed in gray text color.

The full-text versions of the publications are enclosed in the Appendix together with copyright information. Furthermore, all aforementioned publications can be found using the ORCID iD² of the author of this dissertation: 0000-0003-2492-4358.

¹Contributor Roles Taxonomy, <http://credit.niso.org/>

²Open Researcher and Contributor ID, <https://orcid.org>

6.1. Publication 1 : Uncertainty-based Adaptive Classification with Scene Contexts

Reference and Abstract

Jianxiang Feng*, Maximilian Durner*, Zoltán-Csaba Márton, Bálint-Benczédi Ferenc, Rudolph Triebel. “Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks”. In: *Robotics Research. ISRR 2019. Springer Proceedings in Advanced Robotics*. 2019.

Full text of the publication enclosed in the Appendix, reference in bibliography [Fen+19b].

Abstract – This work focuses on improving uncertainty estimation in the field of object classification from RGB images and demonstrates its benefits in two robotic applications. We employ a Bayesian Neural Network (BNN), and evaluate two practical inference techniques to obtain better uncertainty estimates, namely Concrete Dropout (CDP) and Kronecker-factored Laplace Approximation (LAP). We show a performance increase using more reliable uncertainty estimates as unary potentials within a Conditional Random Field (CRF), which is able to incorporate contextual information as well. Furthermore, the obtained uncertainties are exploited to achieve domain adaptation in a semi-supervised manner, which requires less manual efforts in annotating data. We evaluate our approach on two public benchmark datasets that are relevant for robot perception tasks.

Author’s Contributions

The author of the dissertation designed and investigated the concept of combining Bayesian Neural Networks and Conditional Random Fields with the major support from Maximilian Durner. He took the lead of developing the software for processing the data and implementing the ideas yielded from the discussion with Maximilian Durner and Zoltan-Csaba Marton. With the data provided by the co-authors, he carried out most of the experiments for the idea validation. Besides, he wrote the original draft, created the visualization of the method and iterated the draft with the co-authors for readability improvement and publication.

CRedit: **Jianxiang Feng**: Conceptualization; Methodology; Software; Investigation; Visualization; Data Curation; Validation; Writing – original draft; Writing – review & editing. **Maximilian Durner**: Conceptualization; Methodology; Investigation; Resources; Supervision; Data Curation; Writing – review & editing. **Zoltán-Csaba Márton**: Conceptualization; Methodology; Formal Analysis; Supervision; Writing – review & editing. **Bálint-Benczédi Ferenc**: Data Curation. **Rudolph Triebel**: Funding acquisition; Visualization; Writing – review & editing.

6.2. Publication 2 : Flow-based Open-Set Object Detection

Reference and Abstract

Jianxiang Feng, Jongseok Lee, Simon Geisler, Stephan Günnemann, Rudolph Triebel. “Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning”. In: *7th Annual Conference on Robot Learning (CoRL)*. 2023.

Full text of the publication enclosed in the Appendix, reference in bibliography [Fen+23b].

Abstract – To facilitate reliable deployments of autonomous robots in the real world, Out-of-Distribution (OOD) detection capabilities are often required. A powerful approach for OOD detection is based on density estimation with Normalizing Flows (NFs). However, we find that prior work with NFs attempts to match the complex target distribution topologically with naive base distributions leading to adverse implications. In this work, we circumvent this topological mismatch using expressive class-conditional base distributions that we train with an information-theoretic objective to match the required topology. The proposed method enjoys the merits of wide compatibility with existing learned models, efficient runtime, and low memory overhead while enhancing the OOD detection performance. We demonstrate the benefits of our method in density estimation, 2D object detection benchmarks and in particular, showcase the applicability in a real-robot deployment.

Author’s Contributions

The author of the dissertation came up with the idea of exploiting topology-matching normalizing flows for Out-of-distribution detection. Besides conceptualization, he investigated the technical details (i.e. training with an information theoretic objective) and implemented the concept as a software package, which is later used by the other co-authors for experimental validation. With the supports from Jongseok Lee and Simon Geisler, he designed the experiments, conducted the major part of the experimental validation and took the lead of project management. He also wrote the first draft and created the initial visualizations in the manuscript.

CRedit: **Jianxiang Feng**: Conceptualization; Methodology; Software; Investigation; Data Curation; Visualization; Validation; Writing – original draft; Writing – review & editing. **Jongseok Lee**: Software; Investigation; Data Curation; Writing – review & editing. **Simon Geisler**: Software; Investigation; Visualization; Writing – review & editing. **Stephan Günnemann**: Writing – review & editing. **Rudolph Triebel**: Funding Acquisition; Resources.

6.3. Publication 3 : Bayesian Active Learning for Sim-to-Real Object Detection

Reference and Abstract

Jianxiang Feng, Jongseok Lee, Maximilian Durner, Rudolph Triebel. “Bayesian Active Learning for Sim-to-Real Robotic Perception”. In: *the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022.

Full text of the publication enclosed in the Appendix, reference in bibliography [Fen+22].

Abstract – While learning from synthetic training data has recently gained an increased attention, in real-world robotic applications, there are still performance deficiencies due to the so-called Sim-to-Real gap. In practice, this gap is hard to resolve with only synthetic data. Therefore, we focus on an efficient acquisition of real data within a Sim-to-Real learning pipeline. Concretely, we employ deep Bayesian active learning to minimize manual annotation efforts and devise an autonomous learning paradigm to select the data that is considered useful for the human expert to annotate. To achieve this, a Bayesian Neural Network (BNN) object detector providing reliable uncertainty estimates is adapted to infer the informativeness of the unlabeled data. Furthermore, to cope with misalignments of the label distribution in uncertainty-based sampling, we develop an effective randomized sampling strategy that performs favorably compared to other complex alternatives. In our experiments on object classification and detection, we show benefits of our approach and provide evidence that labeling efforts can be reduced significantly. Finally, we demonstrate the practical effectiveness of this idea in a grasping task on an assistive robot.

Author’s Contributions

The author of the dissertation and Jongseok Lee initialized the concept of using Bayesian active learning to bridge the last mile of the sim-to-real gap for robotic perception. The first author is responsible for the software development of the ideas shaped from the discussion with Jongseok Lee. He performed all the experiments on the both the data sets and the real robot based on the aforementioned software and supports from other colleagues in the EDAN team at DLR. He provided the first draft and created the visualization with the help from Jongseok Lee and Maximilian Durner.

CRedit: **Jianxiang Feng**: Conceptualization; Investigation; Methodology; Software; Data Curation; Validation; Visualization; Writing – original draft; Writing – review & editing. **Jongseok Lee**: Conceptualization; Methodology; Data Curation; Visualization; Writing – review & editing. **Maximilian Durner**: Visualization; Writing – review & editing. **Rudolph Triebel**: Funding Acquisition; Writing – review & editing. Annette Hagenruber, Gabriel Quere and the Re-enabling robot (EDAN) team at DLR: Visualization; Resources.

6.4. Publication 4 : Predicting Assembly Sequences via Graph Representations

Reference and Abstract

Matan Atad*, **Jianxiang Feng***, Ismael Rodríguez, Maximilian Durner, Rudolph Triebel. “Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning”. In: *In the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023.

Full text of the publication enclosed in the Appendix, reference in bibliography [Ata+23].

Abstract – Automatic Robotic Assembly Sequence Planning (RASP) can significantly improve productivity and resilience in modern manufacturing along with the growing need for greater product customization. One of the main challenges in realizing such automation resides in efficiently finding solutions from a growing number of potential sequences for increasingly complex assemblies. Besides, costly feasibility checks are always required for the robotic system. To address this, we propose a holistic graphical approach including a graph representation called Assembly Graph for product assemblies and a policy architecture, Graph Assembly Processing Network, dubbed GRACE for assembly sequence generation. With GRACE, we are able to extract meaningful information from the graph input and predict assembly sequences in a step-by-step manner. In experiments, we show that our approach can predict feasible assembly sequences across product variants of aluminum profiles based on data collected in simulation of a dual-armed robotic system. We further demonstrate that our method is capable of detecting infeasible assemblies, substantially alleviating the undesirable impacts from false predictions, and hence facilitating real-world deployment soon. Code and training data are available at <https://github.com/DLR-RM/GRACE>.

Author’s Contributions

The author of the dissertation shaped the concept of exploiting graph representation learning for robotic assembly sequence planning together with Maximilian Durner and Ismael Rodriguez. He decided the actual research direction to proceed for the concept realization and investigate how to perform elaborate experimental validation with Matan Atad. Meanwhile, he provided supervision and guidance for Matan Atad on the software implementation. He took the lead of providing the first draft with supports from Matan Atad and created visualization to significantly enhance the readability of the manuscript.

CRedit: **Matan Atad**: Methodology; Software; Investigation; Writing – original draft; Writing – review & editing. **Jianxiang Feng**: Conceptualization; Methodology; Investigation; Supervision; Visualization; Writing – original draft; Writing – review & editing. **Ismael Rodríguez**: Conceptualization; Data Curation; Writing – review & editing. **Maximilian Durner**: Conceptualization; Methodology; Writing – review & editing. **Rudolph Triebel**: Funding Acquisition.

6.5. Publication 5 (Pre-Print): Introspective Robotic Assembly via Feasibility Learning

Reference and Abstract

Jianxiang Feng*, Matan Atad*, Ismael Rodríguez, Maximilian Durner, Stephan Günemann, Rudolph Triebel. “Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly”. In: *Workshop on Robotics and AI: The Future of Industrial Assembly Tasks , Robotics: Science and Systems (RSS)*. 2023.

Full text of the publication enclosed in the Appendix, reference in bibliography [Fen+23a].

Abstract – Machine Learning (ML) models in Robotic Assembly Sequence Planning (RASP) need to be introspective on the predicted solutions, i.e. whether they are feasible or not, to circumvent potential efficiency degradation. Previous works need both feasible and infeasible examples during training. However, the infeasible ones are hard to collect sufficiently when re-training is required for swift adaptation to new product variants. In this work, we propose a density-based feasibility learning method that requires only feasible examples. Concretely, we formulate the feasibility learning problem as Out-of-Distribution (OOD) detection with Normalizing Flows (NFs), which are powerful generative models for estimating complex probability distributions. Empirically, the proposed method is demonstrated on robotic assembly use cases and outperforms other single-class baselines in detecting infeasible assemblies. We further investigate the internal working mechanism of our method and show that a large memory saving can be obtained based on an advanced variant of NFs.

Author’s Contributions

The author of the dissertation initialized the idea of using Normalizing Flows to learn the assembly feasibility for introspective robotic assembly. With the supports from other co-authors, he refined the methodology for achieving this concept. Together with Matan Atad, he contributed to the software implementation of the proposed idea and designed the structure of the experimental validation. He provided the first draft and created the visualization in the publication. He prepared the slides and video for the presentation at the on-site workshop.

CRedit: **Jianxiang Feng**: Conceptualization; Methodology; Software; Investigation; Supervision; Visualization; Writing – original draft; Writing – review & editing. **Matan Atad**: Methodology; Software; Investigation; Writing – review & editing. **Ismael Rodríguez**: Methodology; Data Curation; Writing – review & editing. **Maximilian Durner**: Methodology; Writing – review & editing. **Stephan Günemann**: Writing – review & editing. **Rudolph Triebel**: Funding Acquisition.

6.6. Publication Licenses

In this section, the re-use licenses for doctoral dissertation of publications included in this thesis are shown except for **Publication 5** , which is a pre-print version.

Publication 1

**SPRINGER NATURE LICENSE
TERMS AND CONDITIONS**

Apr 29, 2024

This Agreement between Mr. Jianxiang Feng ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

| | |
|------------------------------|---|
| License Number | 5778111118970 |
| License date | Apr 29, 2024 |
| Licensed Content Publisher | Springer Nature |
| Licensed Content Publication | Springer eBook |
| Licensed Content Title | Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks |
| Licensed Content Author | Jianxiang Feng, Maximilian Durner, Zoltán-Csaba Márton et al |
| Licensed Content Date | Jan 1, 2022 |
| Type of Use | Thesis/Dissertation |
| Requestor type | academic/university or research institute |
| Format | print and electronic |
| Portion | full article/chapter |
| Will you be translating? | no |
| Circulation/distribution | 30 - 99 |

| | |
|--|---|
| Author of this Springer Nature content | yes |
| Title of new work | Introspective Methods for Learning-enabled Robotic Perception and Planning |
| Institution name | Technical University of Munich (TUM) |
| Expected presentation date | Aug 2024 |
| Requestor Location | Mr. Jianxiang Feng Scheurlinstr. 16 Munich, 81241 Germany Attn: Wenhan Hao |
| Billing Type | Invoice |
| Billing Address | Mr. Jianxiang Feng Scheurlinstr. 16 Munich, Germany 81241 Attn: Wenhan Hao |
| Total | 0.00 EUR |

Terms and Conditions

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and

agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced

with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at <https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/>.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

| | |
|---|--|
| Reuse in a medical communications project | Reuse up to distribution or time period indicated in License |
| Reuse in a dissertation/thesis | Lifetime of thesis |
| Reuse in a journal/magazine | Lifetime of journal/magazine |
| Reuse in a book/textbook | Lifetime of edition |
| Reuse on a website | 1 year unless otherwise specified in the License |
| Reuse in a presentation/slide kit/poster | Lifetime of presentation/slide kit/poster. Note: publication whether electronic or in print of presentation/slide kit/poster may require further permission. |
| Reuse in conference proceedings | Lifetime of conference proceedings |
| Reuse in an annual report | Lifetime of annual report |
| Reuse in training/CME materials | Reuse up to distribution or time period indicated in License |
| Reuse in newsmedia | Lifetime of newsmedia |
| Reuse in coursepack/classroom materials | Reuse up to distribution and/or time period indicated in license |

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: *'Reproduced with permission from Springer Nature'*.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany's choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit <https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution>

Other Conditions:

Version 1.4 - Dec 2022

Questions? customercare@copyright.com.

Publication 2

Reprint Permission

Author: Jianxiang Feng

Publication: Proceedings of The 7th Conference on Robot Learning (CoRL 2023)

Publisher: PMLR 229:3214-3241, 2023.

Date: 11.2023

Copyright ©The authors and PMLR 2024. MLResearchPress.


Reprint Permission

This is an open-access article distributed under the terms of the Creative Commons CC BY license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Publication 3



Bayesian Active Learning for Sim-to-Real Robotic Perception

Conference Proceedings: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Author: Jianxiang Feng

Publisher: IEEE

Date: 23 October 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

© 2024 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Data Security and Privacy | For California Residents | Terms and Conditions

Comments? We would like to hear from you. E-mail us at customer-care@copyright.com

Publication 4



Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning

Conference Proceedings: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Author: Matan Atad

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

© 2024 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Data Security and Privacy | For California Residents | Terms and Conditions

Comments? We would like to hear from you. E-mail us at customer-care@copyright.com

Bibliography

- [Ach+18] J. Achterhold, J. M. Koehler, A. Schmeink, and T. Genewein. “Variational network quantization”. In: *International Conference on Learning Representations*. 2018 (cited on page 15).
- [APH20] U. Aggarwal, A. Popescu, and C. Hudelot. “Active learning for imbalanced datasets”. In: *WACV*. 2020 (cited on page 35).
- [Agh+19] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. López. “Active learning for deep detection neural networks”. In: *Int. Conf. on Computer Vision (ICCV)*. 2019, pp. 3672–3680 (cited on page 33).
- [Ard+18] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe. “Analyzing inverse problems with invertible neural networks”. In: *arXiv preprint arXiv:1808.04730* (2018) (cited on page 16).
- [Ard+20] L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe. “Training normalizing flows with the information bottleneck for competitive generative classification”. In: *Advances in Neural Information Processing Systems 33* (2020), pp. 7828–7840 (cited on pages 27, 28, 30).
- [Ata+23] M. Atad, J. Feng, I. V. Rodriguez Brena, M. Durner, and R. Triebel. “Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2023*. IEEE, 2023. URL: <https://elib.dlr.de/195845/> (cited on pages 10, 63).
- [Bap+19] V. Bapst, A. Sanchez-Gonzalez, C. Doersch, K. Stachenfeld, P. Kohli, P. Battaglia, and J. Hamrick. “Structured agents for physical construction”. In: *ICML*. PMLR. 2019, pp. 464–474 (cited on pages 38, 39).
- [BB98] D. Barber and C. M. Bishop. “Ensemble learning in Bayesian neural networks”. In: *Nato ASI Series F Computer and Systems Sciences* 168 (1998), pp. 215–238 (cited on page 16).
- [BM19] M. Bauer and A. Mnih. “Resampled priors for variational autoencoders”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 66–75 (cited on page 29).

- [Beh+21] J. Behrmann, P. Vicol, K.-C. Wang, R. Grosse, and J. Jacobsen. “Understanding and mitigating exploding inverses in invertible neural networks”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 1792–1800 (cited on page 20).
- [Bel57] R. Bellman. “A Markovian decision process”. In: *Journal of mathematics and mechanics* (1957), pp. 679–684 (cited on page 39).
- [BSK16] F. Berkenkamp, A. P. Schoellig, and A. Krause. “Safe controller optimization for quadrotors with Gaussian processes”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 491–496 (cited on page 15).
- [Bis+10] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppel, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. “The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing”. In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. June 2010, pp. 1–8 (cited on pages 46, 49).
- [Bis06] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006 (cited on page 16).
- [BR07] V. I. Bogachev and M. A. S. Ruas. *Measure theory*. Vol. 1. Springer, 2007 (cited on page 19).
- [Bor+09] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, R.-G. Paolo, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schäffer, and G. Hirzinger. “Rollin’ Justin - Mobile Platform with Variable Base”. In: *Best Video*. May 2009. URL: <https://elib.dlr.de/62526/> (cited on page 46).
- [Bou+18] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, et al. “Using simulation and domain adaptation to improve efficiency of deep robotic grasping”. In: *ICRA*. 2018 (cited on pages 31, 32).
- [BAY21] S. Brody, U. Alon, and E. Yahav. “How attentive are graph attention networks?” In: *arXiv preprint arXiv:2105.14491* (2021) (cited on page 21).
- [BW91] W. L. Buntine and A. S. Weigend. “Bayesian back-propagation”. In: *Complex systems* 5.6 (1991), pp. 603–643 (cited on page 15).
- [CZ22] S. Cao and Z. Zhang. “Deep Hybrid Models for Out-of-Distribution Detection”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 4723–4733. doi: 10.1109/CVPR52688.2022.00469 (cited on page 28).

- [Cha+21] B. Charpentier, O. Borchert, D. Zügner, S. Geisler, and S. Günnemann. “Natural Posterior Network: Deep Bayesian Uncertainty for Exponential Family Distributions”. In: *arXiv preprint arXiv:2105.04471* (2021) (cited on page 16).
- [CZG23] B. Charpentier, C. Zhang, and S. Günnemann. “Training, Architecture, and Prior for Deterministic Uncertainty Methods”. In: *arXiv preprint arXiv:2303.05796* (2023) (cited on page 27).
- [CZG20] B. Charpentier, D. Zügner, and S. Günnemann. “Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts”. In: *arXiv preprint arXiv:2006.09239* (2020) (cited on page 16).
- [Che+19] R. T. Chen, J. Behrmann, D. K. Duvenaud, and J. Jacobsen. “Residual flows for invertible generative modeling”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 16).
- [Che+08] W.-C. Chen, P.-H. Tai, W.-J. Deng, and L.-F. Hsieh. “A three-stage integrated approach for assembly sequence planning using neural networks”. In: *Expert Systems with Applications* 34.3 (2008), pp. 1777–1786 (cited on page 38).
- [Che+18] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. “Domain adaptive faster r-cnn for object detection in the wild”. In: *CVPR*. 2018 (cited on page 32).
- [Cho+21] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez. “Active Learning for Deep Object Detection via Probabilistic Modeling”. In: *ICCV* (2021) (cited on pages 33, 34).
- [COB22] G. Chou, N. Ozay, and D. Berenson. “Safe output feedback motion planning from images via learned perception modules and contraction theory”. In: *International Workshop on the Algorithmic Foundations of Robotics*. Springer. 2022, pp. 349–367 (cited on page 27).
- [CGJ96] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. “Active learning with statistical models”. In: *Journal of Artificial Intelligence Research* 4 (1996) (cited on page 32).
- [CB01] A. Corduneanu and C. M. Bishop. “Variational Bayesian model selection for mixture distributions”. In: *Artificial intelligence and Statistics*. Vol. 2001. Morgan Kaufmann Waltham, MA. 2001, pp. 27–34 (cited on page 15).
- [Cor+20] R. Cornish, A. Caterini, G. Deligiannidis, and A. Doucet. “Relaxing bijectivity constraints with continuously indexed normalising flows”. In: *International conference on machine learning*. PMLR. 2020, pp. 2133–2143 (cited on pages 3, 16, 20).
- [Daf+16] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert. “Introspective perception: Learning to predict failures in vision systems”. In: *2016 IEEE/RSJ International Conference*

- on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 1743–1750 (cited on pages 13, 15).
- [DW87] T. De Fazio and D. Whitney. “Simplified generation of all mechanical assembly sequences”. In: *IEEE Journal on Robotics and Automation* 3.6 (1987), pp. 640–658. doi: 10.1109/JRA.1987.1087132 (cited on page 38).
- [DK+17] F. Dellaert, M. Kaess, et al. “Factor graphs for robot perception”. In: *Foundations and Trends® in Robotics* 6.1-2 (2017), pp. 1–139 (cited on page 15).
- [Den+87] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield. “Large automatic learning, rule extraction, and generalization”. In: *Complex systems* 1.5 (1987), pp. 877–922 (cited on page 15).
- [DL91] J. S. Denker and Y. LeCun. “Transforming neural-net output levels to probability distributions”. In: *Advances in Neural Information Processing Systems* (1991) (cited on page 16).
- [Den+19] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam. “BlenderProc”. In: *arXiv:1911.01911* (2019) (cited on page 52).
- [DSB16] L. Dinh, J. Sohl-Dickstein, and S. Bengio. “Density estimation using real nvp”. In: *arXiv preprint arXiv:1605.08803* (2016) (cited on pages 16, 44).
- [Din+19] L. Dinh, J. Sohl-Dickstein, H. Larochelle, and R. Pascanu. “A RAD approach to deep mixture models”. In: *arXiv preprint arXiv:1903.07714* (2019) (cited on page 16).
- [Dri+20] D. Driess, O. Oguz, J.-S. Ha, and M. Toussaint. “Deep visual heuristics: Learning feasibility of mixed-integer programs for manipulation planning”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 9563–9569 (cited on page 44).
- [Du+22a] X. Du, G. Gozum, Y. Ming, and Y. Li. “SIREN: Shaping Representations for Detecting Out-of-Distribution Objects”. In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022. URL: <https://openreview.net/forum?id=8E8tgnYImN> (cited on page 28).
- [Du+22b] X. Du, X. Wang, G. Gozum, and Y. Li. “Unknown-Aware Object Detection: Learning What You Don’t Know from Videos in the Wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13678–13688 (cited on page 28).
- [Dur+19] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. “Neural spline flows”. In: *Advances in neural information processing systems* 32 (2019) (cited on page 16).

- [Ebr+20] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach. “Uncertainty-guided continual learning with Bayesian neural networks”. In: *International Conference on Learning Representations*. 2020 (cited on page 15).
- [Eve+10] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge.” In: *Int. J. Comput. Vis.* 88.2 (2010), pp. 303–338. URL: <http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWZ10> (cited on page 31).
- [FG19] S. Farquhar and Y. Gal. “A unifying Bayesian view of continual learning”. In: *arXiv preprint arXiv:1902.06494* (2019) (cited on page 15).
- [FUW17] M. Federici, K. Ullrich, and M. Welling. “Improved Bayesian compression”. In: *arXiv preprint arXiv:1711.06494* (2017) (cited on page 15).
- [FI18] Y. Feldman and V. Indelman. “Bayesian viewpoint-dependent robust classification under model and localization uncertainty”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3221–3228 (cited on page 15).
- [Fen+19a] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer. “Deep active learning for efficient training of a lidar 3d object detector”. In: *Intelligent Vehicles Symposium (IV)*. 2019 (cited on page 33).
- [Fen+23a] J. Feng, M. Atad, I. V. Rodriguez Brena, M. Durner, and R. Triebel. “Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly”. In: *18th Robotics: Science and System 2023 Workshops*. 2023. URL: <https://elib.dlr.de/195846/> (cited on pages 10, 27, 64).
- [Fen+19b] J. Feng, M. Durner, Z.-C. Márton, F. Bálint-Benczédi, and R. Triebel. “Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks”. In: *Robotics Research*. Ed. by T. Asfour, E. Yoshida, J. Park, H. Christensen, and O. Khatib. Cham: Springer International Publishing, 2019, pp. 660–675. ISBN: 978-3-030-95459-8 (cited on pages 10, 60).
- [Fen+22] J. Feng, J. Lee, M. Durner, and R. Triebel. “Bayesian Active Learning for Sim-to-Real Robotic Perception”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10820–10827 (cited on pages 10, 13, 62).
- [Fen+23b] J. Feng, J. Lee, S. Geisler, S. Günemann, and R. Triebel. “Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning”. In: *7th Annual Conference on Robot Learning*. 2023. URL: <https://openreview.net/forum?id=BzjLaVvr955> (cited on pages 10, 61).

- [Fet+19] E. Fetaya, J. Jacobsen, W. Grathwohl, and R. Zemel. “Understanding the limitations of conditional generative models”. In: *arXiv preprint arXiv:1906.01171* (2019) (cited on page 30).
- [Fox98] D. Fox. “Markov localization—a probabilistic framework for mobile robot localization and navigation.” PhD thesis. Universität Bonn, 1998 (cited on pages 13, 15).
- [Fox+00] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. “A probabilistic approach to collaborative multi-robot localization”. In: *Autonomous robots* 8.3 (2000), pp. 325–344 (cited on page 15).
- [Fox+06] M. Fox, M. Ghallab, G. Infantes, and D. Long. “Robot introspection through learned hidden Markov models”. In: *Artificial Intelligence* 170.2 (2006), pp. 59–113. ISSN: 0004-3702. DOI: <https://doi.org/10.1016/j.artint.2005.05.007>. URL: <https://www.sciencedirect.com/science/article/pii/S000437020500113X> (cited on pages 4, 13, 14).
- [Fun+22] N. Funk, G. Chalvatzaki, B. Belousov, and J. Peters. “Learn2assemble with structured representations and search for robotic architectural construction”. In: *CoRL*. PMLR. 2022, pp. 1401–1411 (cited on pages 38, 39).
- [GG16] Y. Gal and Z. Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *Int. Conf. on Machine Learning (ICML)*. 2016 (cited on pages 17, 33).
- [GHK17] Y. Gal, J. Hron, and A. Kendall. “Concrete dropout”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 3581–3590 (cited on pages 18, 27).
- [GIG17] Y. Gal, R. Islam, and Z. Ghahramani. “Deep bayesian active learning with image data”. In: *International conference on machine learning*. PMLR. 2017, pp. 1183–1192 (cited on page 15).
- [Gan+16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. “Domain-adversarial training of neural networks”. In: *Journal of Machine Learning Research* 17.1 (2016) (cited on page 36).
- [Gaw+23] 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* (2023), pp. 1–77 (cited on pages 1, 14, 28).
- [Geo+17] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka. “Synthesizing training data for object detection in indoor scenes”. In: *arXiv:1702.07836* (2017) (cited on page 31).

- [GYD19] S. Ghosh, J. Yao, and F. Doshi-Velez. “Model Selection in Bayesian Neural Networks via Horseshoe Priors.” In: *J. Mach. Learn. Res.* 20.182 (2019), pp. 1–46 (cited on page 15).
- [Gil+17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. “Neural message passing for quantum chemistry”. In: *ICML*. PMLR. 2017, pp. 1263–1272 (cited on page 21).
- [Gra+18] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. “Ffjord: Free-form continuous dynamics for scalable reversible generative models”. In: *arXiv preprint arXiv:1810.01367* (2018) (cited on page 16).
- [Gri+13] H. Grimmert, R. Paul, R. Triebel, and I. Posner. “Knowing when we don’t know: Introspective classification for mission-critical decision making”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 4531–4538 (cited on pages 4, 14).
- [Gri+16] H. Grimmert, R. Triebel, R. Paul, and I. Posner. “Introspective classification for robot perception”. In: *The International Journal of Robotics Research* 35.7 (2016), pp. 743–762 (cited on pages 13, 15).
- [Guo+17] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. “On Calibration of Modern Neural Networks”. In: *International Conference on Machine Learning*. 2017, pp. 1321–1330 (cited on page 14).
- [GN99] A. Gupta and D. Nagar. *Matrix Variate Distributions*. Vol. 104. CRC Press, 1999 (cited on page 19).
- [Gur+18] C. Gurău, D. Rao, C. H. Tong, and I. Posner. “Learn from experience: probabilistic prediction of perception performance to avoid failure”. In: *The International Journal of Robotics Research* 37.9 (2018), pp. 981–995 (cited on page 15).
- [GTP16] C. Gurău, C. H. Tong, and I. Posner. “Fit for purpose? Predicting perception performance based on past experience”. In: *International Symposium on Experimental Robotics*. Springer. 2016, pp. 454–464 (cited on page 15).
- [Hal+20] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf. “Probabilistic object detection: Definition and evaluation”. In: *WACV*. 2020 (cited on page 34).
- [HSW20] A. Harakeh, M. Smart, and S. L. Waslander. “Bayesod: A bayesian approach for uncertainty estimation in deep object detectors”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 87–93 (cited on pages 28, 33).

- [Hin+18] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige. “On pre-trained image features and synthetic images for deep learning”. In: *Europ. Conf. on Computer Vision (ECCV) Workshops*. 2018 (cited on page 31).
- [HV93] G. E. Hinton and D. Van Camp. “Keeping the neural networks simple by minimizing the description length of the weights”. In: *Proceedings of the sixth annual conference on Computational learning theory*. 1993, pp. 5–13 (cited on page 16).
- [Hod+19] T. Hodan, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter. “Photorealistic Image Synthesis for Object Instance Detection”. In: *arXiv* (2019). eprint: 1902.03334 (cited on page 31).
- [HS90] L. Homem de Mello and A. Sanderson. “AND/OR graph representation of assembly plans”. In: *IEEE Transactions on Robotics and Automation* 6.2 (1990), pp. 188–199. doi: 10.1109/70.54734 (cited on page 38).
- [HK17] H. Hu and G. Kantor. “Introspective Evaluation of Perception Performance for Parameter Tuning without Ground Truth”. In: *Robotics: Science and Systems*. 2017 (cited on pages 13, 15).
- [Hua+18] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. “Neural autoregressive flows”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2078–2087 (cited on page 16).
- [Hul+08] T. Hulin, M. Sagardia, J. Artigas, S. Schaetzle, P. Kremer, and C. Preusche. “Human-Scale Bimanual Haptic Interface”. In: *Proceedings*. 2008. URL: <https://elib.dlr.de/55612/> (cited on page 46).
- [HLT20] M. Humt, J. Lee, and R. Triebel. “Bayesian optimization meets laplace approximation for robotic introspection”. In: *arXiv preprint arXiv:2010.16141* (2020) (cited on page 15).
- [Hun+21] C.-M. Hung, L. Sun, Y. Wu, I. Havoutis, and I. Posner. “Introspective visuomotor control: exploiting uncertainty in deep visuomotor control for failure recovery”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 6293–6299 (cited on page 15).
- [Isk+19] M. Iskandar, G. Quere, A. Hagenhuber, A. Dietrich, and J. Vogel. “Employing whole-body control in assistive robotics”. In: *Int. Conf. on Intelligent Robots and Systems (IROS)*. 2019 (cited on page 46).
- [IR16] Iwankowicz and R.R. “An efficient evolutionary method of assembly sequence planning for shipbuilding industry”. In: *Assembly Automation* 36.1 (2016), pp. 60–71. doi: <https://doi.org/10.1108/AA-02-2015-013> (cited on page 38).

- [Izm+20] P. Izmailov, P. Kirichenko, M. Finzi, and A. G. Wilson. “Semi-supervised learning with normalizing flows”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4615–4630 (cited on page 16).
- [JGH18] A. Jacot, F. Gabriel, and C. Hongler. “Neural tangent kernel: Convergence and generalization in neural networks”. In: *Advances in neural information processing systems*. 2018, pp. 8571–8580 (cited on page 11).
- [Jai+20] P. Jai, I. Kobyzev, Y. Yu, and M. Brubaker. “Tails of Lipschitz triangular flows”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4673–4681 (cited on pages 16, 20).
- [JSY22] D. Jiang, S. Sun, and Y. Yu. “Revisiting flow generative models for Out-of-distribution detection”. In: *International Conference on Learning Representations*. 2022. URL: <https://openreview.net/forum?id=6y2KBh-0Fd9> (cited on page 27).
- [Joh+16] M. Johnson, D. K. Duvenaud, A. Wiltchko, R. P. Adams, and S. R. Datta. “Composing graphical models with neural networks for structured representations and fast inference”. In: *Advances in neural information processing systems*. 2016, pp. 2946–2954 (cited on page 24).
- [Kae+10] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. “The Bayes tree: An algorithmic foundation for probabilistic robot mapping”. In: *Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 157–173 (cited on pages 13, 15).
- [Kah+17] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. “Uncertainty-aware reinforcement learning for collision avoidance”. In: *arXiv preprint arXiv:1702.01182* (2017) (cited on page 15).
- [Kao+18] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu. “Localization-aware active learning for object detection”. In: *ACCV*. 2018 (cited on page 33).
- [KG17] A. Kendall and Y. Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *arXiv preprint arXiv:1703.04977* (2017) (cited on page 15).
- [Kha+19] M. E. E. Khan, A. Immer, E. Abedi, and M. Korzepa. “Approximate Inference Turns Deep Networks into Gaussian Processes”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 3088–3098 (cited on page 16).
- [KD18] D. P. Kingma and P. Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *Advances in neural information processing systems* 31 (2018) (cited on pages 16, 31).
- [KIW20] P. Kirichenko, P. Izmailov, and A. G. Wilson. “Why normalizing flows fail to detect out-of-distribution data”. In: *Advances in neural information processing systems* 33 (2020), pp. 20578–20589 (cited on pages 16, 27).

- [KVG19] A. Kirsch, J. Van Amersfoort, and Y. Gal. “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on pages 15, 33).
- [KPB20] I. Kobyzev, S. J. Prince, and M. A. Brubaker. “Normalizing flows: An introduction and review of current methods”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 3964–3979 (cited on page 16).
- [KZB19] A. Kolesnikov, X. Zhai, and L. Beyer. “Revisiting self-supervised visual representation learning”. In: *arXiv preprint arXiv:1901.09005* (2019) (cited on page 24).
- [Kum+23] N. Kumar, S. Šegvić, A. Eslami, and S. Gumhold. “Normalizing Flow based Feature Synthesis for Outlier-Aware Object Detection”. In: *arXiv preprint arXiv:2302.07106* (2023) (cited on page 28).
- [Kuo+21] C.-Y. Kuo, A. Schaarschmidt, Y. Cui, T. Asfour, and T. Matsubara. “Uncertainty-aware contact-safe model-based reinforcement learning”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3918–3925 (cited on page 15).
- [LPB17] 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. 6402–6413 (cited on page 28).
- [LC10] Y. LeCun and C. Cortes. “MNIST handwritten digit database”. In: (2010) (cited on page 36).
- [Lee+22] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel. “Trust your robots! predictive uncertainty estimation of neural networks with sparse gaussian processes”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 1168–1179 (cited on pages 11, 28).
- [Lee+20] J. Lee, M. Humt, J. Feng, and R. Triebel. “Estimating Model Uncertainty of Neural Networks in Sparse Information Form”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 5702–5713 (cited on page 11).
- [Lee+23] J. Lee, R. Radhakrishna Balachandran, K. Kondak, A. Coelho, M. De Stefano, M. Humt, J. Feng, T. Asfour, and R. Triebel. “Virtual Reality via Object Pose Estimation and Active Learning: Realizing Telepresence Robots with Aerial Manipulation Capabilities”. In: *Field Robotics* 3 (2023), pp. 323–367 (cited on pages 2, 8, 11, 50, 51).
- [LBH12] D. Leidner, C. Borst, and G. Hirzinger. “Things are made for what they are: Solving manipulation tasks by using functional object classes”. In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE. 2012, pp. 429–435 (cited on page 50).

- [Li+22a] B. Li, Y. Wu, H. Sun, Z. Cheng, and J. Liu. “Unity 3D-Based Simulation Data Driven Robotic Assembly Sequence Planning Using Genetic Algorithm”. In: *2022 14th International Conference on Computer and Automation Engineering (ICCAE)*. 2022, pp. 1–7. doi: 10.1109/ICCAE55086.2022.9762444 (cited on page 38).
- [Li+20a] H. Li, P. Barnaghi, S. Enshaeifar, and F. Ganz. “Continual learning using Bayesian neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.9 (2020), pp. 4243–4252 (cited on page 15).
- [Li+20b] R. Li, A. Jabri, T. Darrell, and P. Agrawal. “Towards practical multi-object manipulation using relational reinforcement learning”. In: *2020 IEEE ICRA*. IEEE. 2020, pp. 4051–4058 (cited on page 39).
- [Li+22b] R. Li, C. Zhang, H. Zhou, C. Shi, and Y. Luo. “Out-of-Distribution Identification: Let Detector Tell Which I Am Not Sure”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer. 2022, pp. 638–654 (cited on pages 28, 57).
- [Lin+17] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang. “Active self-paced learning for cost-effective and progressive face identification”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.1 (2017), pp. 7–19 (cited on page 24).
- [Lin+14] 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: *Europ. Conf.on Computer Vision (ECCV)*. 2014 (cited on pages 31, 47).
- [Lin+22] Y. Lin, A. S. Wang, E. Undersander, and A. Rai. “Efficient and interpretable robot manipulation with graph neural networks”. In: *IEEE RAL* 7.2 (2022), pp. 2740–2747 (cited on pages 38, 39).
- [LLS15] F. Liu, G. Lin, and C. Shen. “CRF learning with CNN features for image segmentation”. In: *Pattern Recognition* 48.10 (2015), pp. 2983–2992 (cited on page 24).
- [LSL15] F. Liu, C. Shen, and G. Lin. “Deep convolutional neural fields for depth estimation from a single image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5162–5170 (cited on page 24).
- [LSS20] A. Loquercio, M. Segu, and D. Scaramuzza. “A general framework for uncertainty estimation in deep learning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3153–3160 (cited on pages 15, 16).
- [LUW17] C. Louizos, K. Ullrich, and M. Welling. “Bayesian compression for deep learning”. In: *Advances in Neural Information Processing Systems* 30 (2017) (cited on page 15).

- [LEH19] B. Lütjens, M. Everett, and J. P. How. “Safe reinforcement learning with model uncertainty estimates”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8662–8668 (cited on page 15).
- [Mac92a] D. MacKay. “Bayesian model comparison and backprop nets”. In: *Advances in Neural Information Processing Systems 4* (1992) (cited on page 15).
- [Mac92b] D. J. MacKay. “A practical Bayesian framework for backpropagation networks”. In: *Neural computation* 4.3 (1992), pp. 448–472 (cited on page 16).
- [Mac92c] D. J. MacKay. “Information-based objective functions for active data selection”. In: *Neural Computation* 4.4 (1992) (cited on pages 15, 34).
- [Mac+21] R. Mackowiak, L. Ardizzone, U. Kothe, and C. Rother. “Generative classifiers as a basis for trustworthy image classification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2971–2981 (cited on page 28).
- [MMT16] C. J. Maddison, A. Mnih, and Y. W. Teh. “The concrete distribution: A continuous relaxation of discrete random variables”. In: *arXiv preprint arXiv:1611.00712* (2016) (cited on page 18).
- [MG15] J. Martens and R. Grosse. “Optimizing neural networks with kronecker-factored approximate curvature”. In: *International conference on machine learning*. 2015, pp. 2408–2417 (cited on page 19).
- [McC99] J. McCarthy. “Making Robots Conscious of Their Mental States”. In: *Machine Intelligence 15, Intelligent Agents [St. Catherine’s College, Oxford, July 1995]*. GBR: Oxford University, 1999, pp. 3–17. ISBN: 0198538677 (cited on pages 4, 12, 14).
- [Mil+18] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. “Dropout sampling for robust object detection in open-set conditions”. In: *ICRA*. 2018 (cited on pages 15, 28, 33).
- [Mil+21] D. Miller, N. Sünderhauf, M. Milford, and F. Dayoub. “Uncertainty for identifying open-set errors in visual object detection”. In: *IEEE Robotics and Automation Letters* 7.1 (2021), pp. 215–222 (cited on pages 28, 31, 57).
- [Mor07] A. C. Morris. *Robotic introspection for exploration and mapping of subterranean environments*. Carnegie Mellon University, 2007 (cited on pages 4, 13, 14).
- [MWJ13] K. Murphy, Y. Weiss, and M. I. Jordan. “Loopy belief propagation for approximate inference: An empirical study”. In: *arXiv preprint arXiv:1301.6725* (2013) (cited on page 26).

- [Mur23] K. P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL: <http://probml.github.io/book2> (cited on page 14).
- [Nal+18] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. “Do deep generative models know what they don’t know?” In: *arXiv preprint arXiv:1810.09136* (2018) (cited on pages 16, 27).
- [Nal+19a] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. “Hybrid models with deep and invertible features”. In: *International Conference on Machine Learning*. PMLR, 2019, pp. 4723–4732 (cited on page 27).
- [Nal+19b] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. “Detecting out-of-distribution inputs to deep generative models using typicality”. In: *arXiv preprint arXiv:1906.02994* (2019) (cited on page 27).
- [Nea92] R. M. Neal. *Bayesian training of backpropagation networks by the hybrid Monte Carlo method*. Tech. rep. University of Toronto, 1992 (cited on page 16).
- [Ngu+18] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. “Variational continual learning”. In: *International Conference on Learning Representations*. 2018 (cited on page 15).
- [Ngu+20] S. Nguyen, O. S. Oguz, V. N. Hartmann, and M. Toussaint. “Self-Supervised Learning of Scene-Graph Representations for Robotic Sequential Manipulation Planning.” In: *Conference on Robot Learning (CoRL)*. 2020, pp. 2104–2119 (cited on pages 38, 39).
- [Nie+20] D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling. “Survae flows: Surjections to bridge the gap between vaes and flows”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12685–12696 (cited on page 16).
- [Not+16a] K. Nottensteiner, T. Bodenmueller, M. Kassecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas. “A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution”. In: *Proceedings of ISR 2016: 47th International Symposium on Robotics*. 2016, pp. 1–8 (cited on page 38).
- [Not+16b] K. Nottensteiner, T. Bodenmüller, M. Kaßecker, M. A. Roa Garzon, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas. “A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution”. In: *47th International Symposium on Robotics, ISR 2016*. VDE Verlag, June 2016. URL: <https://elib.dlr.de/133190/> (cited on page 50).
- [Oks+20] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas. “Imbalance problems in object detection: A review”. In: *Trans. on Pattern Analysis and Machine Intelligence* (2020) (cited on page 35).

- [Ova+19] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. “Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift”. In: *Advances in neural information processing systems* 32 (2019) (cited on page 14).
- [Pap+21] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. “Normalizing flows for probabilistic modeling and inference”. In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 2617–2680 (cited on pages 16, 19, 57).
- [Per+20] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly. “A smooth representation of belief over so (3) for deep rotation learning with uncertainty”. In: *arXiv preprint arXiv:2006.01031* (2020) (cited on page 15).
- [Pos+20] J. Postels, H. Blum, Y. Strümler, C. Cadena, R. Siegwart, L. Van Gool, and F. Tombari. “The hidden uncertainty in a neural networks activations”. In: *arXiv preprint arXiv:2012.03082* (2020) (cited on page 16).
- [Pos+21] J. Postels, M. Liu, R. Spezialetti, L. Van Gool, and F. Tombari. “Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction”. In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 1249–1258 (cited on page 16).
- [PDS19] A. Prabhu, C. Dognin, and M. Singh. “Sampling bias in deep active classification: An empirical study”. In: *arXiv:1909.09389* (2019) (cited on pages 31, 35).
- [Pra+21] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman. “Active domain adaptation via clustering uncertainty-weighted embeddings”. In: *CVPR*. 2021 (cited on page 33).
- [Que+20] G. Quere, A. Hagengruber, M. S. Z. Iskandar, S. Bustamante Gomez, D. Leidner, F. Stulp, and J. Vogel. “Shared Control Templates for Assistive Robotics”. In: *ICRA*. 2020 (cited on pages 46–48).
- [RM17] A. Rashid and M.F.F. “A hybrid Ant-Wolf Algorithm to optimize assembly sequence planning problem”. In: *Assembly Automation* 37.2 (2017), pp. 238–248. DOI: <https://doi.org/10.1108/AA-11-2016-143> (cited on page 38).
- [Ren+19] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan. “Likelihood ratios for out-of-distribution detection”. In: *Advances in neural information processing systems* 32 (2019) (cited on page 27).
- [Ren+15] 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) (cited on page 31).

- [RM15] D. Rezende and S. Mohamed. “Variational inference with normalizing flows”. In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538 (cited on page 16).
- [RBK18] S. M. Richards, F. Berkenkamp, and A. Krause. “The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 466–476 (cited on page 15).
- [RR17] C. Richter and N. Roy. “Safe visual navigation via deep learning and novelty detection”. In: *Robotics, Science and Systems (RSS)*. 2017 (cited on page 15).
- [RBB18] H. Ritter, A. Botev, and D. Barber. “A Scalable Laplace Approximation for Neural Networks”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=Skdvd2xAZ> (cited on page 27).
- [Rod+19] I. Rodríguez, K. Nottensteiner, D. Leidner, M. Kaßbecker, F. Stulp, and A. Albu-Schäffer. “Iteratively refined feasibility checks in robotic assembly sequence planning”. In: *IEEE RAL* 4.2 (2019), pp. 1416–1423 (cited on pages 2, 38, 50).
- [Rod+20] I. Rodríguez, K. Nottensteiner, D. Leidner, M. Durner, F. Stulp, and A. Albu-Schäffer. “Pattern recognition for knowledge transfer in robotic assembly sequence planning”. In: *IEEE Robotics and Automation Letters (RAL)* 5.2 (2020), pp. 3666–3673 (cited on pages 38, 40).
- [Ros+08] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa. “Online planning algorithms for POMDPs”. In: *Journal of Artificial Intelligence Research* 32 (2008), pp. 663–704 (cited on page 15).
- [RUN18] S. Roy, A. Unmesh, and V. P. Namboodiri. “Deep active learning for object detection.” In: *British Machine Vision Conf. (BMVC)*. Vol. 362. 2018 (cited on pages 33, 35).
- [RGG15] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez. “UPGMpp: a Software Library for Contextual Object Recognition”. In: *3rd. Workshop on Recognition and Action for Scene Understanding (REACTS)*. 2015 (cited on page 26).
- [Sar+19a] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak. “Development of sam: cable-suspended aerial manipulator”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5323–5329 (cited on page 48).
- [Sar+19b] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak. “Development of SAM: cable-Suspended Aerial Manipulator”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 5323–5329. doi: 10.1109/ICRA.2019.8793592 (cited on page 8).

- [Sat01] M.-A. Sato. “Online model selection based on the variational Bayes”. In: *Neural computation* 13.7 (2001), pp. 1649–1681 (cited on page 15).
- [Sch+99] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. “Support vector method for novelty detection”. In: *Advances in neural information processing systems* 12 (1999) (cited on page 45).
- [Shi20] W. C. Shih. “Global supply chains in a post-pandemic world”. In: *Harvard Business review*, 98(5), 82-89 (2020) (cited on page 37).
- [Shi+20] K. Shinde, J. Lee, M. Humt, A. Sezgin, and R. Triebel. “Learning Multiplicative Interactions with Bayesian Neural Networks for Visual-Inertial Odometry”. In: *arXiv preprint arXiv:2007.07630* (2020) (cited on page 15).
- [SV10] D. Silver and J. Veness. “Monte-Carlo planning in large POMDPs”. In: *Advances in Neural Information Processing Systems* 23 (2010) (cited on pages 13, 15).
- [SB05] C. Sinanoğlu and H. R. Börklü. “An assembly sequence-planning system for mechanical parts using neural network”. In: *Assembly Automation* (2005) (cited on page 38).
- [Sin+22] R. Sinha, A. Sharma, S. Banerjee, T. Lew, R. Luo, S. M. Richards, Y. Sun, E. Schmerling, and M. Pavone. “A System-Level View on Out-of-Distribution Data in Robotics”. In: *arXiv preprint arXiv:2212.14020* (2022) (cited on pages 1, 14).
- [Sri+14] 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 (cited on page 17).
- [Sti+23] V. Stimper, D. Liu, A. Campbell, V. Berenz, L. Ryll, B. Schölkopf, and J. M. Hernández-Lobato. “normflows: A PyTorch Package for Normalizing Flows”. In: *arXiv preprint arXiv:2302.12014* (2023) (cited on page 45).
- [SSH22] V. Stimper, B. Schölkopf, and J. M. Hernández-Lobato. “Resampling Base Distributions of Normalizing Flows”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 4915–4936 (cited on pages 16, 27, 29, 45).
- [Stu+11] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal. “Learning to grasp under uncertainty”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 5703–5708 (cited on page 15).
- [Su+20] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker. “Active adversarial domain adaptation”. In: *Winter Conf. on Applications of Computer Vision (WACV)*. 2020 (cited on page 33).

- [SH13] Y. Sun and J. Han. “Mining heterogeneous information networks: a structural analysis approach”. In: *Acm Sigkdd Explorations Newsletter* 14.2 (2013), pp. 20–28 (cited on page 21).
- [Sün+18] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, et al. “The limits and potentials of deep learning for robotics”. In: *The International journal of robotics research* 37.4-5 (2018), pp. 405–420 (cited on pages 1, 2, 6, 12, 14, 15, 37, 55).
- [Tan20] A. K. Tanwani. “Domain Invariant Representation Learning for Sim-to-Real Transfer”. In: *CoRL*. 2020 (cited on page 32).
- [Tl18] V. Tchuiev and V. Indelman. “Inference over distribution of posterior class probabilities for reliable Bayesian classification and object-level perception”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4329–4336 (cited on page 15).
- [TBW03] U. Thomas, M. Barrenscheen, and F. Wahl. “Efficient assembly sequence planning using stereographical projections of C-space obstacles”. In: *Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2003*. 2003, pp. 96–102. doi: 10.1109/ISATP.2003.1217194 (cited on page 38).
- [TSR15] U. Thomas, T. Stouraitis, and M. A. Roa. “Flexible assembly through integrated assembly sequence planning and grasp planning”. In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. 2015, pp. 586–592. doi: 10.1109/CoASE.2015.7294142 (cited on page 38).
- [TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Cambridge: MIT press, Aug. 2005. ISBN: 978-0-262-20162-9 (cited on pages 13, 15).
- [Thr+01] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. “Robust Monte Carlo localization for mobile robots”. In: *Artificial intelligence* 128.1-2 (2001), pp. 99–141 (cited on page 15).
- [TLS89] N. Tishby, E. Levin, and S. A. Solla. “Consistent inference of probabilities in layered networks: Predictions and generalization”. In: *International Joint Conference on Neural Networks*. Vol. 2. IEEE. 1989, pp. 403–409 (cited on page 15).
- [TPB00] N. Tishby, F. C. Pereira, and W. Bialek. “The information bottleneck method”. In: *arXiv preprint physics/0004057* (2000) (cited on pages 28, 30).
- [Tob+17] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World”. In: *IROS* (2017). doi: 10.1109/iros.2017.8202133 (cited on page 31).

- [Tom+14] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. “Joint training of a convolutional network and a graphical model for human pose estimation”. In: *Advances in neural information processing systems*. 2014, pp. 1799–1807 (cited on page 24).
- [Tri+16] R. Triebel, H. Grimmett, R. Paul, and I. Posner. “Driven learning for driving: How introspection improves semantic mapping”. In: *Robotics Research*. Springer, 2016, pp. 449–465 (cited on pages 4, 13, 15).
- [TGP13] R. Triebel, H. Grimmett, and I. Posner. “Confidence Boosting: Improving the Introspectiveness of a Boosted Classifier for Efficient Learning”. In: *Workshop on Autonomous Learning. IEEE International Conference on Robotics and Automation (ICRA)*. May 2013 (cited on page 13).
- [UVL16] D. Ulyanov, A. Vedaldi, and V. Lempitsky. “Instance normalization: The missing ingredient for fast stylization”. In: *arXiv preprint arXiv:1607.08022* (2016) (cited on page 41).
- [Vas+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017) (cited on page 40).
- [Vel+17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. “Graph attention networks”. In: *Stat* 1050.20 (2017), pp. 10–48550 (cited on pages 21, 41).
- [Vog+20a] J. Vogel, D. Leidner, A. Hagenruber, M. Panzirsch, B. Bauml, M. Denninger, U. Hillenbrand, L. Suchenwirth, P. Schmaus, M. Sewtz, et al. “An ecosystem for heterogeneous robotic assistants in caregiving: Core functionalities and use cases”. In: *IEEE Robotics & Automation Magazine* 28.3 (2020), pp. 12–28 (cited on page 46).
- [VH18] J. Vogel and A. Hagenruber. “An sEMG-based interface to give people with severe muscular atrophy control over assistive devices”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 2136–2141 (cited on page 46).
- [Vog+20b] J. Vogel, A. Hagenruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Höppner, D. Leidner, and A. Albu-Schäffer. “EDAN: An EMG-controlled Daily Assistant to Help People With Physical Disabilities”. In: *IROS*. 2020 (cited on pages 2, 8, 46, 51).
- [WBL23] 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 (cited on pages 31, 55).
- [WY16] H. Wang and D.-Y. Yeung. “Towards bayesian deep learning: A survey”. In: *arXiv preprint arXiv:1604.01662* (2016) (cited on page 24).

- [Wan+16] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. “Cost-effective active learning for deep image classification”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016), pp. 2591–2600 (cited on page 24).
- [Wan+19] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. “Heterogeneous graph attention network”. In: *The world wide web conference*. 2019, pp. 2022–2032 (cited on page 21).
- [WI20a] K. Watanabe and S. Inada. “Search algorithm of the assembly sequence of products by using past learning results”. In: *International Journal of Production Economics* 226 (2020), p. 107615 (cited on page 38).
- [Wei+22a] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. “Mitigating Neural Network Overconfidence with Logit Normalization”. In: *International Conference on Machine Learning (ICML)* (2022) (cited on page 28).
- [Wei+22b] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. “Chain-of-thought prompting elicits reasoning in large language models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24824–24837 (cited on page 58).
- [Wel+19] A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki. “Learning feasibility for task and motion planning in tabletop environments”. In: *IEEE RAL* 4.2 (2019), pp. 1255–1262 (cited on page 44).
- [Wen+19] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen. “Bayesian uncertainty matching for unsupervised domain adaptation”. In: *arXiv:1906.09693* (2019) (cited on page 33).
- [WI20b] A. G. Wilson and P. Izmailov. “Bayesian deep learning and a probabilistic perspective of generalization”. In: *Advances in Neural Information Processing Systems* 33 (2020) (cited on page 15).
- [WKN20] H. Wu, J. Köhler, and F. Noé. “Stochastic normalizing flows”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5933–5944 (cited on page 16).
- [Xia+18] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes”. In: *Robotics: Science and Systems (RSS)* (2018) (cited on page 36).
- [XXL19] J. Xu, L. Xiao, and A. M. Lopez. “Self-supervised Domain Adaptation for Computer Vision Tasks”. In: *arXiv preprint arXiv:1907.10915* (2019) (cited on page 24).
- [Xu+22] L. Xu, T. Ren, G. Chalvatzaki, and J. Peters. “Accelerating Integrated Task and Motion Planning with Neural Feasibility Checking”. In: *arXiv preprint arXiv:2203.10568* (2022) (cited on page 44).

- [Yan+21] J. Yang, K. Zhou, Y. Li, and Z. Liu. “Generalized out-of-distribution detection: A survey”. In: *arXiv preprint arXiv:2110.11334* (2021) (cited on pages 27, 28).
- [YM+10] Y. Yang, G. Ma, et al. “Ensemble-based active learning for class imbalance problem”. In: *Journal of Biomedical Science and Engineering* 3.10 (2010) (cited on page 35).
- [YGF22] Z. Yang, C. R. Garrett, and D. Fox. “Sequence-Based Plan Feasibility Prediction for Efficient Task and Motion Planning”. In: *arXiv preprint arXiv:2211.01576* (2022) (cited on page 44).
- [Ye+20] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani. “Object-centric forward modeling for model predictive control”. In: *CoRL*. PMLR. 2020, pp. 100–109 (cited on page 38).
- [Zha+20] H. Zhang, A. Li, J. Guo, and Y. Guo. “Hybrid models for open set recognition”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III* 16. Springer. 2020, pp. 102–117 (cited on pages 27, 28).
- [Zha+21] E. Zhao, A. Liu, A. Anandkumar, and Y. Yue. “Active Learning under Label Shift”. In: *Int. Conf. on Artificial Intelligence and Statistics*. 2021 (cited on pages 31, 36).
- [Zha+19] M. Zhao, X. Guo, X. Zhang, Y. Fang, and Y. Ou. “ASPW-DRL: assembly sequence planning for workpieces via a deep reinforcement learning approach”. In: *Assembly Automation* (2019) (cited on page 38).
- [Zho+20] X. Zhou, H. Wu, J. Rojas, Z. Xu, and S. Li. “Introduction to Robot Introspection”. In: *Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection*. Singapore: Springer Singapore, 2020, pp. 1–10. ISBN: 978-981-15-6263-1. DOI: 10.1007/978-981-15-6263-1_1. URL: https://doi.org/10.1007/978-981-15-6263-1_1 (cited on page 14).
- [Zhu+19] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin. “Adapting object detectors via selective cross-domain alignment”. In: *Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cited on page 32).
- [Zhu+21] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu. “Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 6541–6548 (cited on page 38).
- [Zou+18] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 289–305 (cited on page 24).

A. Full Text of Publications

A.1. Publication 1

Jianxiang Feng*, Maximilian Durner*, Zoltán-Csaba Márton, Bálint-Benczédi Ferenc, Rudolph Triebel (2019): “*Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks*”. In Robotics Research. ISRR 2019. Springer Proceedings in Advanced Robotics, vol 20. Springer, Cham.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- https://link.springer.com/chapter/10.1007/978-3-030-95459-8_40

Please refer to the final published version for citation:

```
@InProceedings{feng2019,
  author="Feng, Jianxiang
  and Durner, Maximilian
  and M{\`a}rton, Zolt{\`a}n-Csaba
  and B{\`a}lint-Bencz{\`e}di, Ferenc
  and Triebel, Rudolph",
  title="Introspective Robot Perception Using Smoothed Predictions from
  Bayesian Neural Networks",
  booktitle="Robotics Research",
  year="2022",
  publisher="Springer International Publishing",
  address="Cham",
  pages="660--675",
  isbn="978-3-030-95459-8"
}
```



Introspective Robot Perception Using Smoothed Predictions from Bayesian Neural Networks

Jianxiang Feng^{1(✉)}, Maximilian Durner¹, Zoltán-Csaba Márton¹,
Ferenc Bálint-Benczédi², and Rudolph Triebel^{1,3}

¹ Institute of Robotics and Mechatronics, German Aerospace Center (DLR),
Oberpfaffenhofen-Weßling, Germany

{jianxiang.feng,maximilian.durner,zoltan-Csaba.marton,rudolph.triebel}@dlr.de

² Institute for Artificial Intelligence, University of Bremen, Bremen, Germany
balintbe@cs.uni-bremen.de

³ Department of Computer Science, Technical University of Munich,
Munich, Germany
rudolph.triebel@in.tum.de

Abstract. This work focuses on improving uncertainty estimation in the field of object classification from RGB images and demonstrates its benefits in two robotic applications. We employ a Bayesian Neural Network (BNN), and evaluate two practical inference techniques to obtain better uncertainty estimates, namely Concrete Dropout (CDP) and Kronecker-factored Laplace Approximation (LAP). We show a performance increase using more reliable uncertainty estimates as unary potentials within a Conditional Random Field (CRF), which is able to incorporate contextual information as well. Furthermore, the obtained uncertainties are exploited to achieve domain adaptation in a semi-supervised manner, which requires less manual efforts in annotating data. We evaluate our approach on two public benchmark datasets that are relevant for robot perception tasks.

Keywords: BNN · CRF · Introspective classification

1 Introduction

Visual scene understanding plays an important role in the field of robotic perception. In recent years, deep learning showed promising results within this context (e.g. object classification, detection or segmentation). Yet, although the applied deep neural networks outperform most traditional methods, they lack a significant property for robots in real world: a reliable uncertainty estimation. Advanced

J. Feng and M. Durner—Equal contributions.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-030-95459-8_40.

robotics highly rely on perceptual systems in order to be able to understand and adapt to its environment. Providing also the confidence of predictions based on the perceived information enhances the ability of robotic systems even further. It equips robots with the ability to know when it does and when it does not know. Besides the safety issue—for the robot itself and its surroundings—introspection about the predictions also has a positive impact on decision making, failure recovery and human-robot interaction. Furthermore, reliable uncertainty estimation is beneficial for active learning [1], reinforcement learning [2–4], detection of the unknown classes and adversarial attacks [5–7]. Recent research on improving the uncertainty estimation of deep neural networks includes BNNs [2, 8–15], bootstrapping [3], ensemble methods [16] and so on. Among them, a BNN is more theoretically sound and able to provide promising performances. By taking into account the practicality in real-world applications, we evaluate BNNs with two inference techniques which are CDP [11] and LAP [14] in term of comprehensive metrics. However, we are more curious about the question, to which extent the improved uncertainty estimates can boost the performances on uncertainty-relevant tasks. Therefore, in this work we focus on studying the improvements by exploiting uncertainty estimates from BNNs which are demonstrated by applying them to (1) support CRFs which can incorporate additional contextual information as well and (2) reduce the manual efforts for data annotations in domain adaptation tasks.

In the line of combining deep learning and Probabilistic Graphical Models (PGMs) [17], previous works [18–21] mainly focus on joint training of these two kinds of model in order to share the advantages of both, which are abilities of expressive representation learning and structured learning, respectively. None of them emphasize the role of uncertainty estimation when combining them as sub-modules, which can improve the robustness of the system in practical applications such as real world robotics. In this work, we propose to use uncertainty estimates to improve classification by combining CRFs (see Fig. 1).

On the other hand, robots deployed in a new situation are often confronted with environmental changes and novel objects. Nevertheless, in most of the time a base classifier trained on an easily obtainable dataset (e.g. public large-scale or synthetic) is available beforehand. The classifier needs to be adapted to the test environment, while the manual efforts of collecting and annotating the adaptation data should be kept as low as possible. This requirement can be cast into the field of domain adaptation in a self/semi-supervised manner. Self-supervised learning refers to learning with self-provided supervisions such as geometrical cues within images [22] instead of strong but laborious human-supervisions and these self-supervisions can be extended to self-generated pseudo labels by the model itself, which can be used for domain adaptation naturally [23–25]. This task can also be framed into a semi-supervised manner, when a small amount of manual annotations are allowed to be taken into the procedure [26, 27]. Among these prior works, none of them highlights the importance of uncertainty estimates which can help distinguishing true positives (served for automatic-annotation) and false positives in both self-supervised and semi-supervised manner. To the best of our knowledge, we are the first to utilize uncertainty estimates from BNNs in this kind of tasks.

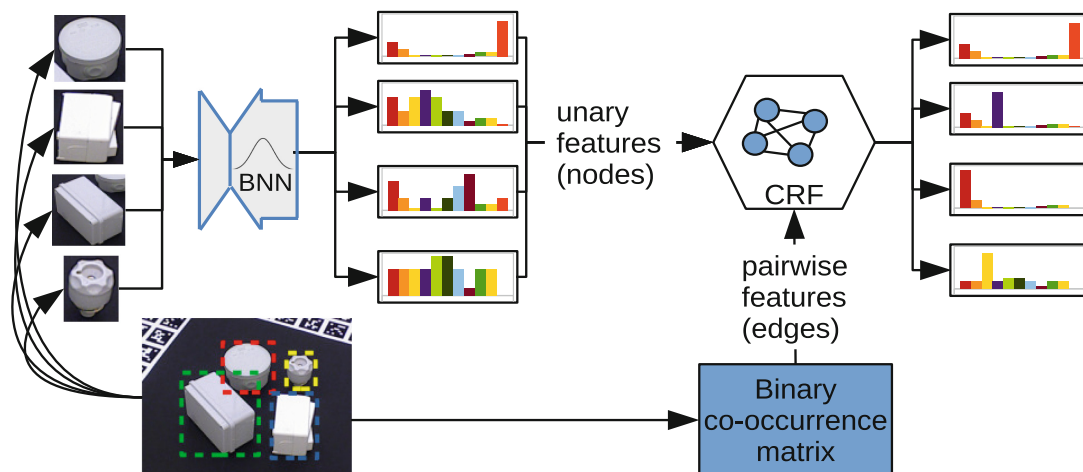


Fig. 1. The combination of BNN and CRF: the predictive distributions of objects in the scene from BNN serve as unary features in the CRF, which can take into account the contextual information from the scene of objects.

The remaining of the paper is organized as follows: we review prior works in the related areas in Sect. 2. While Sect. 3 recaps the theoretical concept of BNNs, Sect. 4 explains our proposed approaches. Then we show experimental results demonstrating their effectiveness in Sect. 5 and conclude in Sect. 6.

2 Related Work

A BNN [28, 29] provides a principal way to obtain model uncertainty by considering the distribution on model parameters. However, it has difficulty scaling to complex network architectures and large training sets nowadays. Besides sampling based methods [8, 15], Variational Inference (VI) [30] suits practical applications due to its ability of fast inference. In the era of deep learning, there is a bunch of research works in this direction [2, 10, 12–14]. CDP [11] is an extension of Monte Carlo Dropout (MCD) [9] which can learn dropout rates from the data without efforts of manual tuning. More than that, CDP can be inserted into existing network architectures very easily. On the other hand LAP does not require re-training and thus suits most of the already-trained networks as well.

Combination of Deep Learning and PGMs: Liu et al. [19] trained a Convolutional Neural Network (CNN) and CRF jointly for depth estimation, while Tompson et al. [18] integrated Markov random fields with CNN for pose estimation. Wang et al. [20] combined deep learning with Bayesian networks for recommendation systems and topic models. Johnson et al. [21] proposed Structured variational autoencoder (SVAE) to learn a structured and thus more interpretable latent representation. Our work differ from them in the way of training. Since we want to evaluate the effects of uncertainty estimates, it's better to analyze them separately. Similar to us, Liu et al. [31] combined features learned from deep neural nets and CRF for segmentation tasks. But they trained another clas-

sifier with these features for the unary potentials without evaluating the effects of uncertainty estimates.

Semi/Self-supervised Domain Adaptation: Some works [22, 25] aim to learn a more generalized feature distribution via designing specific *pretext* tasks without explicit human supervisions (e.g. class labels). Others [24, 26, 27] tried to employ true positives as self-supervisions for adaptation. Zou et al. [24] mentioned the class imbalance problem and proposed to mitigate it by normalizing the class-wise confidence. To note that this problem is obvious in this kind of task, which was verified and mitigated by class-balanced augmentations in our experiments.

3 Bayesian Neural Networks

In general, a neural network can be modelled as a function $f^\omega(\mathbf{x}) = \mathbf{y}$ that maps from an input space \mathcal{X} to an output space \mathcal{Y} , where $\omega = \{W_{1:L}, \mathbf{b}_{1:L}\}$ are the weights of the network consisting of matrices W_i and biases \mathbf{b}_i for each of its L layers. In the training phase, the weights ω are determined by optimizing a loss function $E(f^\omega(\mathbf{x}_i), \mathbf{y}_i)$ for a given training data set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\}$. In contrast, a Bayesian Neural Network (BNN) not only aims to find an optimal ω , but also defines a *posterior distribution* $p(\omega | \mathcal{D})$. Given this posterior, inference on a new test sample $(\mathbf{x}^*, \mathbf{y}^*)$ can be done using the *predictive distribution*

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^* | \mathbf{x}^*, \omega) p(\omega | \mathcal{D}) d\omega, \quad (1)$$

where for classification tasks the likelihood $p(\mathbf{y}^* | \mathbf{x}^*, \omega)$ is usually obtained from the *softmax* of the prediction $f^\omega(\mathbf{x}^*)$. The benefit of using (1) for predictions instead of only using the likelihood is that the model also incorporates the *epistemic* uncertainty, i.e. the one that stems from incorrect model parameters, thereby providing better (less overconfident) uncertainty estimates.

Unfortunately, obtaining the parameter posterior $p(\omega | \mathcal{D})$ is not tractable in all but the simplest cases due to the high dimensionality of the parameter space. Therefore, approximations need to be used, and we investigate two common ones: the CDP and Kronecker-factored LAP.

3.1 Concrete Dropout

Dropout [32] was originally proposed to regularize the training process of Deterministic Neural Network (DNN) to improve their generalization performance, although yet without a formal interpretation. Then, Gal [33] showed that using dropout can be interpreted as sampling from a distribution $q_\theta(\omega)$ that approximates the posterior $p(\omega | X, Y)$ in terms of the KL-divergence

$$KL(q_\theta(\omega) || p(\omega | \mathcal{D})) = - \int q_\theta(\omega) \log \frac{p(\omega | \mathcal{D})}{q_\theta(\omega)}. \quad (2)$$

where $\theta = \{\boldsymbol{\omega}, \mathbf{p}\}$, \mathbf{p} is the vector of dropout rates of layers in which dropout is inserted. Minimizing this is equivalent to minimizing the Evidence Lower Bound (ELBO)

$$\mathcal{L}(\theta) = - \sum_{i=1}^N \int q_{\theta}(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} + \text{KL}(q_{\theta}(\boldsymbol{\omega}) || p(\boldsymbol{\omega})) \quad (3)$$

$$\approx - \sum_{i \in \mathcal{S}} \frac{N}{K} \int q_{\theta}(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} + \text{KL}(q_{\theta}(\boldsymbol{\omega}) || p(\boldsymbol{\omega})), \quad (4)$$

where \mathcal{S} is a mini-batch of size K . To estimate the expected log likelihood in the first term, Monte Carlo integration is used, i.e. samples are generated from $q_{\theta}(\boldsymbol{\omega})$, and the integral is approximated by summing likelihood terms over the samples. The problem here is that using this standard method, this first term can not be derived with respect to θ , which is necessary to minimize $\mathcal{L}(\theta)$. Therefore, the *re-parameterization trick* is used, i.e. a bivariate transformation $g(\theta, \epsilon)$ is used to separate the parameters θ from samples $\epsilon \sim p(\epsilon)$ that are generated from a distribution with fixed parameters. Originally, this could be done only for a Gaussian dropout distribution, later Gal *et al.* [11] showed that for Bernoulli dropout, a *continuous relaxation* of this *discrete* distribution can be found, i.e. a concrete distribution [34], which can then be derived wrt. θ for optimization. This is denoted *concrete dropout*. In our experiments, we use the implementation provided by Gal *et al.* [11].

3.2 Laplace Approximation

The idea within the so-called Laplace approximation is to employ a second-order Taylor expansion at the maximum of the log posterior:

$$\log p(\boldsymbol{\omega} | X, Y) \approx \log p(\boldsymbol{\omega}^* | X, Y) - \frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}^*)^T H(\boldsymbol{\omega} - \boldsymbol{\omega}^*), \quad (5)$$

where $\boldsymbol{\omega}^*$ is the parameter vector that maximizes the log posterior and H is the Hessian of the negative log posterior. Note that the first derivative vanishes at $\boldsymbol{\omega}^*$ and H is p.s.d. because $\boldsymbol{\omega}^*$ is assumed to be a local maximum. After taking the exponential and normalizing we obtain

$$p(\boldsymbol{\omega} | X, Y) \approx \mathcal{N}(\boldsymbol{\omega}^*, H^{-1}). \quad (6)$$

Unfortunately, the dimensionality of this multi-variate normal distribution is in most cases too high to be practical. Also, H needs to be computed on the entire data set, which is also infeasible. Instead, it is approximated by the expected Hessian $\mathbb{E}_{p(X, Y)}[H]$, computed on mini-batches. To reduce the dimensionality, a first step is to assume independence across the layers of the DNN, i.e. H is block-diagonal with L blocks H_i , one for each layer.

Under certain conditions, the Fisher information matrix F , which is the outer product of the first derivatives, is an approximation to the expected Hessian.

Furthermore, in each layer i the block F_i can be approximated by a Kronecker product of two much smaller matrices G_i and A_i , where $G_i = \mathbf{g}_i \mathbf{g}_i^T$ is the outer product of gradients of pre-activation of i -th layer and $A_i = \mathbf{a}_{i-1} \mathbf{a}_{i-1}^T$ is the outer product of activation from the previous layer. This is known as the Kronecker-factored approximate curvature (K-FAC) [35]. If a Gaussian prior is used and F is scaled by the size of the training set N , then the resulting posterior can be written as matrix normal distribution [36]:

$$\mathbf{W}_i \sim \mathcal{MN}(\mathbf{W}_i^*, (\sqrt{N}\mathbb{E}[\mathbf{A}_i] + \sqrt{\tau}\mathbf{I})^{-1}, (\sqrt{N}\mathbb{E}[\mathbf{G}_i] + \sqrt{\tau}\mathbf{I})^{-1}) \quad (7)$$

where τ is the standard deviation of the Gaussian prior. In practice, N and τ can be treated as hyper-parameters as well and tuned on a validation set.

4 Improvements Based on Uncertainty Estimates

In this section, we describe how the uncertainty estimates can be utilized with contextual information within CRF for further improvements. Then, we introduce how to make use of them in adaptive learning for domain adaptation tasks.

4.1 Utilizing Uncertainty Estimates with CRF

While the BNN approach is very useful in providing reliable uncertainty estimates for single object instances, it does not incorporate any *context information* specific for a scene, such that, e.g. more likely object constellations can be accounted for. In order to exploit such contextual information within the classification, we combine the output of the BNN and the relationships between objects within a scene via a CRF (see Fig. 1).

In details, we define a scene as a set of n object instances $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with corresponding class labels $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ represented as one-hot encodings, i.e. $\mathbf{y}_i \in \{0, 1\}^C$ and $\sum_{j=1}^C y_{ij} = 1$, where C is the number of object classes. The CRF models the joint probability $p(\mathbf{y} | \mathbf{x})$ as an undirected graph consisting of cliques of random variables. Here a *pairwise* CRF is used, consisting of nodes \mathcal{V} and edges \mathcal{E} , where the node potentials are modeled as $\phi_u(\mathbf{x}_i, \mathbf{y}_i)$ for individual object instances and the edge potentials $\phi_p(\mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j)$ for pairs of objects $(\mathbf{x}_i, \mathbf{x}_j)$ which are in the scene. Concretely, we define ϕ_u as the predictive probability of each instance (see Eq. (1)) and ϕ_p as the co-occurrence probability of two objects. Co-occurrence probabilities can be obtained from an independent source (as in our household use-case, discussed shortly in Sect. 6). In case the list of expected objects in the scene is known (as in our industrial use-case, evaluated in Subsect. 5.2), the pairwise feature is binary and provided automatically per scene. Thus, the CRF has the following form:

$$p(\mathbf{y} | \mathbf{x}; \theta) = \frac{1}{Z(\mathbf{x}, \theta)} \exp \left(\theta_u \sum_{i \in \mathcal{V}} p(\mathbf{y}_i | \mathbf{x}_i) + \theta_p \sum_{(i,j) \in \mathcal{E}} M(\mathbf{y}_i, \mathbf{y}_j) \right), \quad (8)$$

where $\theta = \{\theta_u, \theta_p\}$ are the node and edge weights respectively, Z is the partition function, and M is a $C \times C$ binary matrix modelling the co-occurrence of two object classes \mathbf{y}_i and \mathbf{y}_j . The training process of the CRF involves minimizing the negative log likelihood, i.e. finding optimal model parameters θ^* such that $\theta^* = \arg \min_{\theta} \{-\log p(\mathbf{y} | \mathbf{x}; \theta)\}$. To do this, we employ Stochastic Gradient Descent (SGD) with momentum, which requires the calculation of gradients and thus an inference step for the likelihood shown in Eq. (8). We use a fully connected CRF, i.e. an exact inference of the likelihood is intractable. Therefore, we apply Loopy Belief Propagation (LBP) for approximate inference. In our implementation, we use the C++ library UPGM++ [37] for this purpose.

4.2 Adaptive Learning for Domain Adaptation

The domain gap between the training and test data distribution deteriorates the performance of most of classifiers. This problem is unavoidable when the classifier is trained on easily obtainable dataset such as a public large-scale or synthetic dataset and then deployed in a real environment.

In this case, the effects of better uncertainty estimates can be presented by adapting the classifier to the test data with as little manual efforts as possible. The proposed flowchart for adaptive learning is visualized in Fig. 2. For this purpose, the classifier should be introspective, that is, to express reliable confidences about its predictions.

At first, the classifier is trained on an easily obtainable or accessible dataset, which can be a large-scale public or synthetic one. Next, in **adaptation phase** the classifier is able to adapt to the test data by fine-tuning itself on the so-called adaptation dataset. In this work, we focus on obtaining this kind of adaptation dataset with as little manual efforts as possible. To this end, the annotations in this dataset are collected in a semi-supervised manner (including both automatic and manual manner). On the one hand, the predictions with high confidence are used for pseudo labels, thus requiring the classifier to provide reliable uncertainty estimation for both correct and false predictions. On the other hand, the classifier would ask people to label a small and random portion of data interactively.

In the end, the adapted classifier is evaluated on the real test data. To note that, if the relationships between objects in the test environment are complementary to the BNN classifier and can be encoded well with pairwise feature, the CRF can be applied to capture them for further improvements.

5 Experiments

In this section, we firstly compared performance on uncertainty estimates of two approximate inference techniques for BNN, which are CDP and LAP on a household objects dataset in terms of comprehensive metrics. Then the one with better performance was applied in the following experiments, which are to evaluate **(1)** the combination with CRF and **(2)** the adaptive learning for domain adaptation respectively.

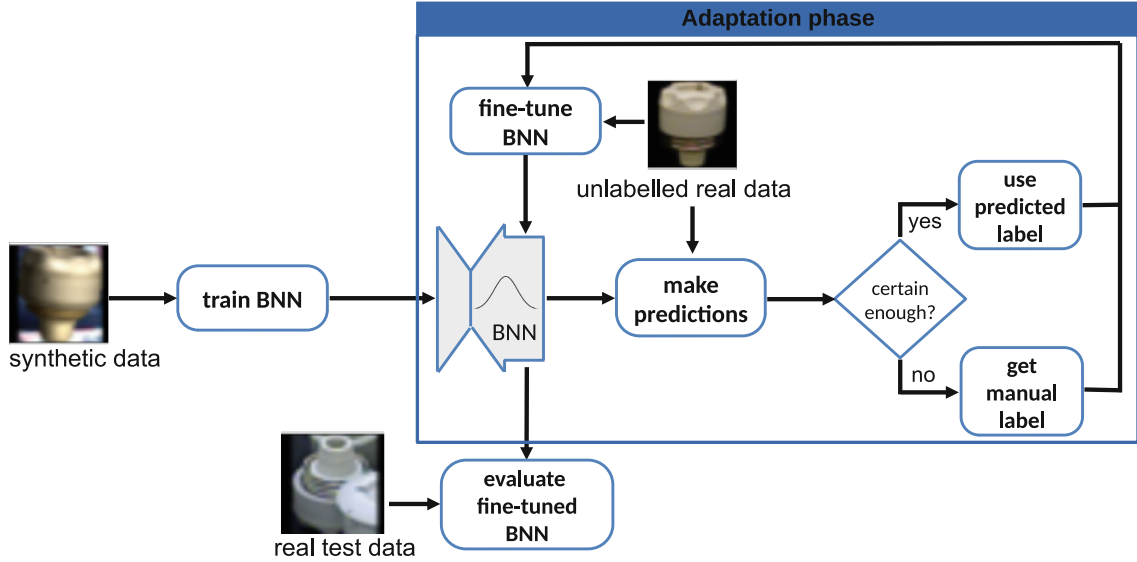


Fig. 2. The flowchart for adaptive learning in domain adaptation. Better uncertainty estimates can help distinguishing certain predictions in automatic labeling during adaptation phase (illustrated on the T-LESS dataset and best viewed in color).

Two types of datasets were employed in our experiments. The first one is the household objects including the RGB-D Dataset from Washington University (WRGB-D) [38] and the UniHB dataset recorded by ourselves trying to mimic the WRGB-D but with only one instance in each category. They contain multi-view images of household objects in 51 classes, with a 15° step in elevation (from 30° to 60°) and 2° step in azimuth (from 0° to 360°). Besides, we have recorded some household objects of novel categories which served as Out-of-distribution (OOD) dataset. The second one is an industrial dataset, T-LESS [39], which has little texture but similar appearance between objects. This dataset contains multi-view images of industrial components objects in 30 classes. The training images depict objects in isolation with a black background, while the test images are from 20 table-top scenes with arbitrarily arranged objects placed on a table (as in a kitting or sorting task). Besides the original T-LESS dataset, we have generated a synthetic dataset trying to mimic the original T-LESS training set. Since there are lots of occlusions in the test scenes, we employed data augmentations both to the original and synthetic T-LESS training set.

As mentioned in Subsect. 4.2, an easily obtainable dataset is used for training in initialization phase. This can be a large-scale public dataset like WRGB-D dataset or synthetic one like the synthetic T-LESS training set we generated. The (independent) adaptation and testing datasets simulate the data that the classifier encounters in the test environment.

5.1 Uncertainty Estimates Evaluation

In this part, we performed extensive experiments to evaluate uncertainty estimates on a household objects dataset. We trained models on the entire WRGB-D dataset and tested them on objects of 30° and 60° in the UniHB dataset.

Different metrics were used for the evaluation. To evaluate calibration performance we used Expected Calibration Error (ECE) and Maximal Calibration Error (MCE) [40]. For summary of both accuracy and calibration we used predictive Negative Log Likelihood (NLL) and brier score, which belong to proper scoring rules [41]. Additionally, we also employed metrics such as area under Receiver Operating Characteristic (ROC) curve and area under Precision Recall (PR) curve to measure the separability between correct predictions and misclassifications as well as OOD predictions. Apart from quantitative metrics, a qualitative (visual) metrics, the histogram (see Fig. 3 and Fig. 4) of uncertainty estimates was employed. For better visualization, we set the normalizer in the histogram as the amount of the corresponding type of prediction. Regarding the uncertainty measure, we evaluated three different ones including confidence (maximum predictive likelihood), predictive entropy and mutual information [33]. The separability metrics list in the Table 1 were chosen based on the uncertainty measure with best performance.

The DNNs and BNNs were implemented in Tensorflow and the optimization was performed using RMSprop with an initial learning rate of $1e^{-5}$ and L2 regularization with coefficient of $3.5e^{-6}$ as well as the dropout regularization with coefficient of $1.0e^{-5}$. Early stopping was applied for model selection, based on the performance on a validation set. During inference, the number of samples drawn from the posterior distribution was set to 50 for both inference methods.

In order to preserve the powerful feature extraction capability of ResNet50 and incorporate the better uncertainty estimation from BNNs, we slightly modify it by appending three fully connected layers with 1024 hidden units before the output layer. CDPs are inserted into the flatten layer and the three new fully connected layers. The weights of these layers were initialized from a Gaussian prior ($\mathcal{N}(0, 0.1)$) and the rest from the model pre-trained on ImageNet [42]. This avoids destroying the pre-trained features and enables the model to possess large enough model capacity which was reduced by inserting dropout [32]. Furthermore, the computation complexity during inference can be reduced by only running the forward pass of the additional layers instead of the whole network. In the following, we show both qualitative and quantitative results in Fig. 3 and Table 1, in which we denote original version of ResNet50 by **ORI** (without additional fully-connected layers), concrete dropout by **CDP**, Laplace approximation by **LAP**. The point estimate model parameters for LAP was model trained with CDP. We set the hyper-parameter N as 1 and τ as 15 in LAP.

As can be seen, BNNs can achieve better performance of uncertainty estimates in terms of all metrics when compared with ORI. At the same time, CDP has better performance than LAP in terms of proper scoring rules and calibration metrics. When OOD predictions were considered along with misclassifications, ECE and MCE decreased significantly. This is because prediction of OOD data is always incorrect and not all predictions of OOD produced high uncertainty correspondingly. If their predictions are highly uncertain, the calibration metrics would have similar values with the ones without OOD data. Both inference methods yield similar results on separability metrics. Based on these experimental results, we used CDP in the following experiments.

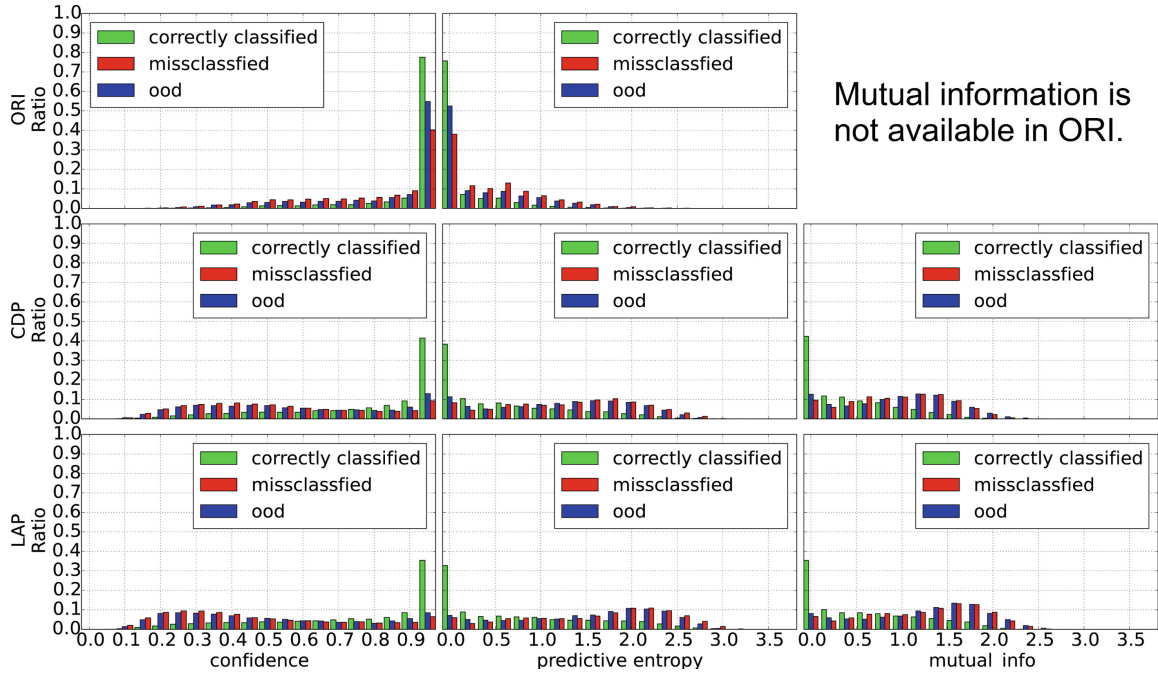


Fig. 3. Histograms of three uncertainty measures including confidence, predictive entropy and mutual information of ORI, CDP and LAP in top-down wise (best viewed in color).

Table 1. Different quantitative results averaged over 3 different random seeds

| | ACC \uparrow | predictive NLL \downarrow | Brier score \downarrow | ECE (w/o. OOD/w. OOD) \downarrow | MCE (w/o. OOD/w. OOD) \downarrow | AUROC (vs. Miss-classified/vs. OOD) \uparrow | AUPR (vs. Miss-classified/vs. OOD) \uparrow |
|-----|--------------------------|-----------------------------|--------------------------|---|---|--|--|
| ORI | 0.568 \pm 0.008 | 3.342 \pm 0.340 | 0.722 \pm 0.019 | 0.304 \pm 0.016/ 0.633 \pm 0.065 | 0.461 \pm 0.027/ 0.362 \pm 0.025 | 0.750 \pm 0.007/ 0.664 \pm 0.011 | 0.802 \pm 0.008/ 0.751 \pm 0.018 |
| CDP | 0.577 \pm 0.008 | 2.088 \pm 0.181 | 0.594 \pm 0.013 | 0.124 \pm 0.023/ 0.288 \pm 0.048 | 0.206 \pm 0.015/ 0.374 \pm 0.018 | 0.775 \pm 0.008/ 0.783 \pm 0.022 | 0.825 \pm 0.007/ 0.850 \pm 0.022 |
| LAP | 0.576 \pm 0.009 | 2.322 \pm 0.350 | 0.602 \pm 0.011 | 0.129 \pm 0.058/ 0.341 \pm 0.157 | 0.235 \pm 0.073/ 0.406 \pm 0.070 | 0.779 \pm 0.004/ 0.782 \pm 0.017 | 0.826 \pm 0.007/ 0.849 \pm 0.016 |

5.2 Combining with CRFs

In this experiment, we will show the results on evaluating the idea introduced in Subsect. 4.1. We use the test set of T-LESS in this part. We split the scenes 2, 3, 5, 8 off for training our CRF and the scenes 1, 4, 6, 7 for testing. These splits were chosen in this way so that as many categories as possible occur in both training and testing (an evaluation on the whole T-LESS test set is shown in the next experiment). The maximum number of iterations during training is 30K, the initial learning rate is $1e^{-4}$, and the size of mini-batch is 16.

In order to see the influence of reliable uncertainty estimates we firstly trained DNNs and BNNs which provide the unary potential in the next step. Preliminary experiments, which are not displayed here, show a significant lower performance of the DNN trained without dropout compared to the BNN. On the other hand, the DNN trained with dropout but turning off MCD during inference (denoted as **NOMCD** in the following) resulted in worse uncertainty estimates but a

better accuracy. Hence, since we want to investigate the effect of uncertainty estimates on the CRFs, we compared the proposed BNN with the **NOMCD**.

Comparing the weights obtained by training the CRF with the uncertainty estimates of NOMCD ($\theta_u = 4.875$; $\theta_p = 6.073$) and BNN ($\theta_u = 8.122$; $\theta_p = 6.59$) a different rating of the provided information can be observed (θ_u vs. θ_p). While in the BNN case the CRF relies more on the classifier, in the NOMCD case the co-occurrence statistics are given a higher importance, reflecting the added usefulness of the correct uncertainty estimates (since the NOMCD and BNN accuracies without smoothing are similar, as seen in Table 2).

Table 2. Results of CRF trained and tested with different unary features

| | Type of unary features in testing | Accuracy with unary potentials | Accuracy with unary and pairwise potentials |
|--|-----------------------------------|--------------------------------|---|
| CRF trained with unary features from NOMCD | NOMCD | 58.48% | 68.6% |
| | BNN | 60.36% | 76.19% |
| CRF trained with unary features from BNN | NOMCD | 58.48% | 68.62% |
| | BNN | 60.36% | 76.36% |

Table 2 shows the much larger performance gain when using the CRFs with better uncertainty estimates, and this is irrespective of the CRF weights used. Besides the performance gain, the CRF is also improving (or at least maintaining) the uncertainty estimates. Figure 4 shows the histogram of confidence of the predictions made by NOMCD and BNN before and after applying LBP inference within CRF. We can see that the uncertainty estimates' quality of NOMCD has been improved and that of BNN has been maintained, which can be helpful for further improvement in the down-stream tasks.

5.3 Adaptive Learning

In this part, a proof-of-concept experiment is performed to evaluate the idea illustrated in Sect. 4.2. To this end, we employed both datasets from two different scenarios for evaluation.

Following the pipeline in Fig. 2, at the beginning we used WRGB-D dataset and the augmented, synthetic T-LESS dataset generated by ourselves for initial training, because they can be obtained more easily. During adaptation phase, objects of 30° and 60° in UniHB dataset (~17.1K) and the original training set of T-LESS (~30K) were used as adaptation dataset. In order to adapt to the test environment, the classifier should be able to collect a dataset for fine-tuning with as little manual efforts as possible. Therefore, this collected dataset can be annotated in two different manners, automatically and manually. The automatically labeled data was selected based on threshold of the uncertainty estimates. In the end, during the deployment phase, the adapted model was evaluated on the test dataset. The 45° objects in UniHB dataset and original test set of T-LESS were treated as data the robot encounters in the test environment.

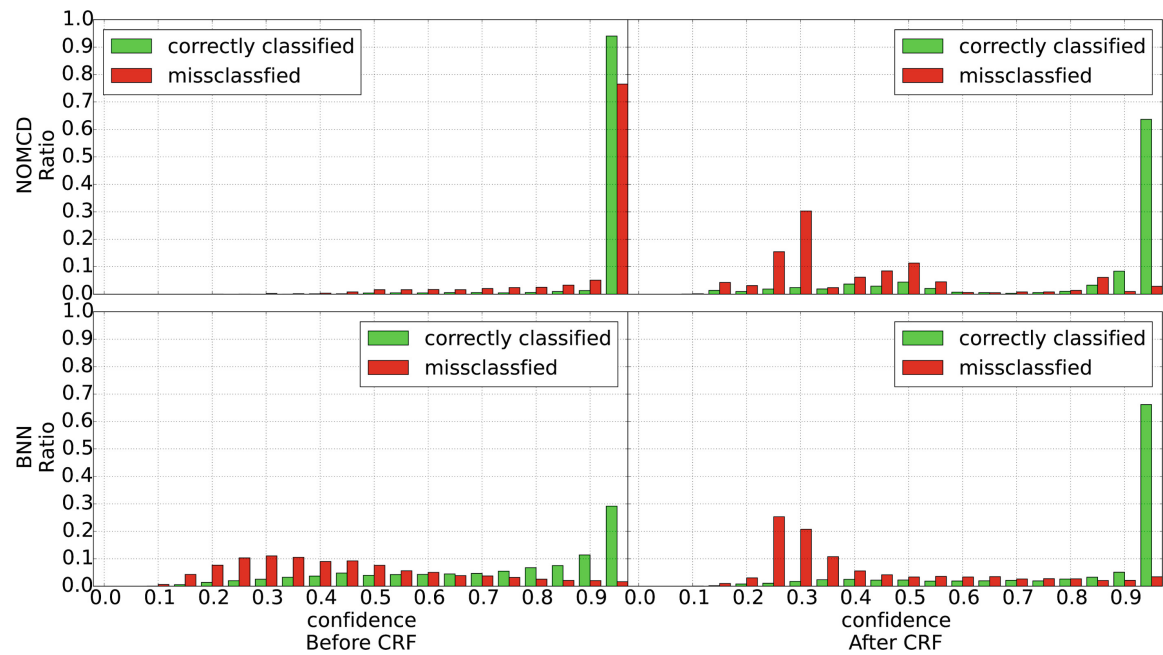


Fig. 4. Histograms of confidence of NOMCD (top row) and BNN (bottom row) before (left column) and after (right column) applying LBP in CRF (best viewed in color).

Household Objects Dataset: We tested different versions of the proposed automatic labeling procedure based on uncertainty estimates, and found that the best results were obtained by setting the confidence threshold s.t. the accuracy of the predictions (estimated on a small manually labeled set) is 95%. The accuracy of automatically labeled data in III, IV is around 96%, matching the 95% estimate.

Our main results are shown in Table 3. As it can be seen, the manual labeling effort can be reduced based on automatic labeling. More detailed testing will be performed on the industrial dataset, based on the insights gained here.

During the experiment, we found that the balance of number of each class on the adaptation dataset plays an important role. The main reason for this should be the different visual domain gap of different objects. The initial model is more familiar with some objects instead of other and thus give lower uncertainty for these familiar ones. Since we selected predictions based on the uncertainty estimates, this would lead to an imbalanced dataset and thus bias the adapted model. Therefore it's important to mitigate this issue. We found that adding manually labeled data and augmentations is useful not only to increase the diversity of the dataset, but to balance the dataset (see III and IV). Other ways of balancing the automatically labeled data (e.g. by selecting the top most confident predictions per class) decreased performance as they resulted in either too few labels or included too many incorrect ones.

Table 3. Results of fine-tuned network on household objects dataset

| Dataset used for fine-tuning | Accuracy (average over 3 random seeds) |
|---|--|
| I: 0% (no fine-tuning) | 66.9% |
| II: 3% manually labeled data, selected randomly (balanced) | 91.7% |
| III: 3% automatically labeled data (imbalanced) | 79.0% |
| IV: 2% automatically labeled data and 1% manually labeled data randomly, augmentation for balance (balanced) | 89.6% |

Industrial Components Dataset: With the same procedure of selecting automatically labeled data, the size of dataset is $\sim 1\text{K}$ with only 93% accuracy using the original ResNet50, but $\sim 1.6\text{K}$ with 96% accuracy using BNN. The summary of the results is shown in Table 4. The performance of the classifier adapted using 3% manually labeled data (VI) is matched by the use of 1% manually labeled data if automatic labeling is employed (V). Moreover, adding the automatic labeling to the 3% manually labeled data can nearly reach the performance of classifier adapted with all available data manually labeled (III vs VII). By incorporating contextual information with CRF, the performance can be increased further (VIII).

Table 4. BNN fine-tuning with different datasets (size of dataset before augmentations is showed in the bracket).

| Dataset used for fine-tuning | Accuracy |
|---|----------|
| I: augmented, synthetic dataset | 34.91% |
| II: fine-tune I with augmented, automatically labeled real dataset ($\sim 1.6\text{K}$) | 53.54% |
| III: entire real dataset, i.e. 100% manually labeled ($\sim 30\text{K}$), augmented | 72.78% |
| IV: fine-tune I with 1% manually labeled real dataset ($\sim 0.3\text{K}$), augmented | 67.4% |
| V: fine-tune I with II and IV ($\sim 1.9\text{K}$), augmented | 68.1% |
| VI: fine-tune I with 3% manually labeled real dataset ($\sim 0.9\text{K}$), augmented | 68.1% |
| VII: fine-tune I with II and VI ($\sim 2.5\text{K}$), augmented | 72.48% |
| VIII: Incorporating contextual information with CRF based on VII | 74.64% |

6 Conclusions

We presented an approach to make robots learning new objects more introspectively, by improving its awareness of possible mistakes, and leveraging this in two ways: first, for better incorporating context information (if available) through smoothing over all object predictions using a CRF, and second, for exploiting this in semi-supervised domain adaptation, where the mostly correct predictions are automatically obtained as adaptation data while asking humans for help with the more uncertain ones.

The improved uncertainty estimation from BNN plays an important role especially in the latter use-case, because it not only provides a reliable uncertainty estimation, but also increases the separability between correct predictions and false predictions, which is more useful in this task. It was found, however that it is very important to ensure that the data is balanced. For manual labeling this can be easily achieved by requesting the human operator to label a more-or-less equal number of instances of each object, e.g. repeatedly selecting random subsets and having to click all occurrences of an object (as in an image CAPTCHA), then switching to the next target object once enough samples were collected. For the automatic labeling, random selection is not a good alternative, as the accuracy penalty would be too large if the overall performance of the initial classifier is too low (as in our cases). It could be, however, incorporated if multiple rounds of adaptation are performed, and the performance is gradually increasing to acceptable levels (around 95% in our tests).

In the former use-case the importance of a clear co-occurrence statistic is highlighted by the fact that the CRF failed to improve results on the household dataset (the pairwise weight was negligible) due to the difficulty of obtaining good co-occurrence statistics in this scenario (which we mined from word co-occurrences in WikiHow articles) and since many household objects have similar appearances and contexts at the same time. In an industrial scenario, e.g. for kitting applications, such a list of parts is available, and the learned CRF weights generalize well over objects and scenes.

Acknowledgments. This work was partially funded by the Big Data Interdisciplinary Project of DLR e.V. under the project number 2464047. Jianxiang Feng is supported by the Munich School for Data Science (MUDS) and Rudolph Triebel is a member of MUDS.

References

1. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1183–1192. JMLR. org (2017)
2. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International Conference on Machine Learning, pp. 1613–1622 (2015)
3. Osband, I., Blundell, C., Pritzel, A., Van Roy, B.: Deep exploration via bootstrapped DQN. In: Advances in Neural Information Processing Systems, pp. 4026–4034 (2016)
4. Gal, Y., McAllister, R., Rasmussen, C.E.: Improving PILCO with Bayesian neural network dynamics models. In: Data-Efficient Machine Learning workshop, ICML (2016)
5. Grimmett, H., Triebel, R., Paul, R., Posner, I.: Introspective classification for robot perception. *Int. J. Robot. Res. (IJRR)* **35**(7), 743–762 (2016)
6. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings (2017). <https://openreview.net/forum?id=Hkg4TI9xl>

7. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint [arXiv:1611.01236](https://arxiv.org/abs/1611.01236) (2016)
8. Balan, A.K., Rathod, V., Murphy, K.P., Welling, M.: Bayesian dark knowledge. In: Advances in Neural Information Processing Systems, pp. 3438–3446 (2015)
9. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059 (2016)
10. Louizos, C., Welling, M.: Structured and efficient variational deep learning with matrix gaussian posteriors. In: International Conference on Machine Learning, pp. 1708–1716 (2016)
11. Gal, Y., Hron, J., Kendall, A.: Concrete dropout. In: Advances in Neural Information Processing Systems, pp. 3581–3590 (2017)
12. Sun, S., Chen, C., Carin, L.: Learning structured weight uncertainty in Bayesian neural networks. In: Artificial Intelligence and Statistics, pp. 1283–1292 (2017)
13. Louizos, C., Welling, M.: Multiplicative normalizing flows for variational Bayesian neural networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2218–2227. JMLR. org (2017)
14. Ritter, H., Botev, A., Barber, D.: A scalable Laplace approximation for neural networks. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=Skdvd2xAZ>
15. Wang, K., Vicol, P., Lucas, J., Gu, L., Grosse, R.B., Zemel, R.S.: Adversarial distillation of Bayesian neural network posteriors. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018, pp. 5177–5186 (2018)
16. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems, pp. 6402–6413 (2017)
17. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIR Press, Cambridge (2009)
18. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems, pp. 1799–1807 (2014)
19. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162–5170 (2015)
20. Wang, H., Yeung, D.Y.: Towards Bayesian deep learning: a survey. arXiv preprint [arXiv:1604.01662](https://arxiv.org/abs/1604.01662) (2016)
21. Johnson, M., Duvenaud, D.K., Wiltschko, A., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Advances in Neural Information Processing Systems, pp. 2946–2954 (2016)
22. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. arXiv preprint [arXiv:1901.09005](https://arxiv.org/abs/1901.09005) (2019)
23. Tang, K., Ramanathan, V., Fei-Fei, L., Koller, D.: Shifting weights: adapting object detectors from image to video. In: Advances in Neural Information Processing Systems, pp. 638–646 (2012)
24. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 289–305 (2018)
25. Xu, J., Xiao, L., Lopez, A.M.: Self-supervised domain adaptation for computer vision tasks. arXiv preprint [arXiv:1907.10915](https://arxiv.org/abs/1907.10915) (2019)

26. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. *IEEE Trans. Circuits Syst. Video Technol.* **27**(12), 2591–2600 (2016)
27. Lin, L., Wang, K., Meng, D., Zuo, W., Zhang, L.: Active self-paced learning for cost-effective and progressive face identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(1), 7–19 (2017)
28. MacKay, D.J.: A practical Bayesian framework for backpropagation networks. *Neural Comput.* **4**(3), 448–472 (1992)
29. Neal, R.M.: *Bayesian Learning For Neural Networks*. LNS, vol. 118. Springer Science & Business Media, New York (2012). <https://doi.org/10.1007/978-1-4612-0745-0>
30. Graves, A.: Practical variational inference for neural networks. In: *Advances in Neural Information Processing Systems*, pp. 2348–2356 (2011)
31. Liu, F., Lin, G., Shen, C.: CRF learning with CNN features for image segmentation. *Pattern Recogn.* **48**(10), 2983–2992 (2015)
32. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
33. Gal, Y.: *Uncertainty in deep learning*. Ph.D. thesis, University of Cambridge (2016)
34. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: a continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* (2016)
35. Martens, J., Grosse, R.: Optimizing neural networks with Kronecker-factored approximate curvature. In: *International Conference on Machine Learning*, pp. 2408–2417 (2015)
36. Gupta, A., Nagar, D.: *Matrix Variate Distributions*, vol. 104. CRC Press, Boca Raton (1999)
37. Ruiz-Sarmiento, J.R., Galindo, C., González-Jiménez, J.: UPGMpp: a software library for contextual object recognition. In: *3rd Workshop on Recognition and Action for Scene Understanding (REACTS)* (2015)
38. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 1817–1824. IEEE (2011)
39. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: an RGB-D dataset for 6D pose estimation of texture-less objects. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2017)
40. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org (2017)
41. Gneiting, T., Balabdaoui, F., Raftery, A.E.: Probabilistic forecasts, calibration and sharpness. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **69**(2), 243–268 (2007)
42. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)

A.2. Publication 2

Jianxiang Feng, Jongseok Lee, Simon Geisler, Stephan Günnemann, Rudolph Triebel (2023): “*Topology-Matching Normalizing Flows for Out-of-Distribution in Robot Learning*”. 7th Annual Conference on Robot Learning (CoRL).

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- <https://proceedings.mlr.press/v229/feng23b.html>

Please refer to the final published version for citation:

```
@inproceedings{
  feng2023topologymatching,
  title={Topology-Matching Normalizing Flows for Out-of-Distribution
    Detection in Robot Learning},
  author={Jianxiang Feng and Jongseok Lee and Simon Geisler and Stephan Gü
    nnemann and Rudolph Triebel},
  booktitle={7th Annual Conference on Robot Learning},
  year={2023},
  url={https://openreview.net/forum?id=BzjLaVvr955}
}
```

Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning

Jianxiang Feng^{*,1}, Jongseok Lee^{2,3}, Simon Geisler¹, Stephan Günnemann¹, Rudolph Triebel^{2,3}

¹ Department of Informatics, Technical University of Munich (TUM)

² Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

³ Department of Informatics, Karlsruhe Institute of Technology (KIT)

jianxiang.feng@tum.de, {jongseok.lee, rudolph.triebel}@dlr.de,
{geisler, guennemann}@in.tum.de

Abstract: To facilitate reliable deployments of autonomous robots in the real world, Out-of-Distribution (OOD) detection capabilities are often required. A powerful approach for OOD detection is based on density estimation with Normalizing Flows (NFs). However, we find that prior work with NFs attempts to match the complex target distribution topologically with naïve base distributions leading to adverse implications. In this work, we circumvent this topological mismatch using an expressive class-conditional base distribution trained with an information-theoretic objective to match the required topology. The proposed method enjoys the merits of wide compatibility with existing learned models without any performance degradation and minimum computation overhead while enhancing OOD detection capabilities. We demonstrate superior results in density estimation and 2D object detection benchmarks in comparison with extensive baselines. Moreover, we showcase the applicability of the method with a real-robot deployment.

Keywords: Normalizing Flows, Out-of-Distribution, Robotic Introspection

1 Introduction

The reliable identification of **Out-of-Distribution (OOD)** data, which is not well represented in the training set, poses a pressing challenge on the path towards trustworthy open-world robotic systems such as self-driving cars [1], delivery drones [2] or healthcare robots [3]. For example, with widespread adoption in the perception pipeline, existing object detectors have been reported to overconfidently misclassify an **OOD** object into a known class, which might obfuscate the decision-making module and eventually cause catastrophic consequences in safety-critical scenarios [1, 4, 5].

Normalizing Flows (NFs) are a popular class of generative models [6, 7, 8, 9] that may be used for **OOD** detection. **NFs** represent complex probability distributions [10] with a learnable series of transformations from a simple base distribution to a complex target distribution. However, **NFs**' expressivity [11, 12, 13] and numerical stability [14, 15] is limited by a fundamental constraint: the supports of the base and target distribution should preserve *similar topological properties* (Definition 3.3.10 in Runde [16]). The topological properties subsume different geometrical characteristics of the target distribution, including its continuity, the number of connected components, or the number of modes. Increasing the capacity of the transformation may mitigate this constraint. Yet, this raises computation and memory demands [11, 17, 12]. An alternative to overcome the topological mismatch is to increase the flexibility of the base distribution, which is surprisingly under-explored in the **OOD** detection literature.

Therefore, we propose to equip **NFs** with efficient but flexible base distributions for **OOD** detection in robot learning. Concretely, we replace the frequently used uni-modal Gaussian base distribution

*: work done when working at DLR.
code: <https://github.com/DLR-RM>

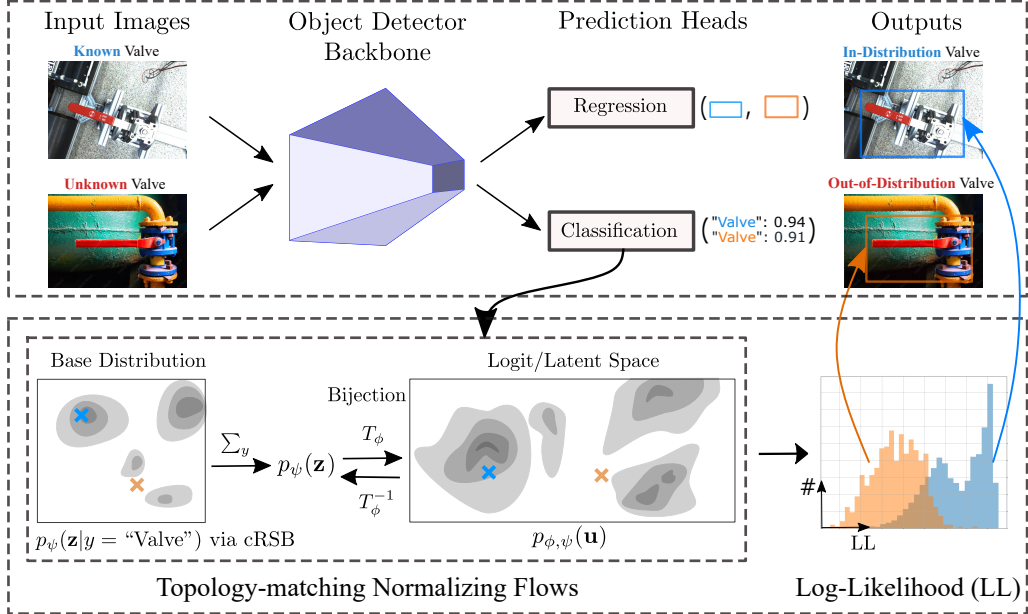


Figure 1: **The proposed architecture.** We overcome the topological mismatch problem in NFs to accurately model **In-Distribution (ID)** density. That is, the **Conditional Resampled Base Distributions (cRSB)** base distribution trained with **Information Bottleneck (IB)** $p_\psi(\mathbf{z}|\mathbf{y})$ can, e.g., adapt the numbers of modes to match target distribution with complex topology. Then we can identify **OOD** objects by low predicted log-likelihoods more reliably (best viewed in color).

with the **cRSB**, a class-conditional version of a learnable base distribution for mitigating the topological problem in NFs – **Resampled Base Distributions (RSB)** [13]. **cRSB** can learn the required topological properties, like adapting the number of modes, to match the unknown topological structure of the latent class-specific target distribution (Figure 1). Moreover, we adapt our **cRSB** with an adapted **IB** objective [18] to balance fusing class-conditional information with the marginalized density estimation capabilities in NFs. **IB** [19] is an information-theoretic objective to incorporate task-specific details e.g. class conditions, which are commonly ignored in pure generative modeling. This delivers a topology in the base distribution that is more accurately aligned to the one in the target distribution (see Figure 3).

Our **OOD** detection approach using topology-matching NFs is powerful and yet resource-efficient for open-set object detection. It is applicable to diverse object detectors (e.g., Faster-RCNN [20] and Yolov7 [21] used in this work) with minor changes and no loss of prediction performance. Moreover, our approach is sampling-free, i.e., only a single forward pass is required for efficient test-time inference while keeping the space memory tractable. As a result, our method is suitable for robotic applications that require a fast and robust perception module. We empirically show the state-of-the-art performance of the proposed idea using synthetic density estimation and 2D object detection tasks against extensive baselines. To further validate the applicability in robotics, we examine an object detector equipped with the proposed method on an exemplary inspection and maintenance aerial robot, showing the practical benefits of negligible memory and run-time overhead.

Contributions. Our main contribution is a NFs-based **OOD** detection method that overcomes the topological constraints while taking class-conditional information into account. We show that training with **IB** yields effective representation with superior **OOD** detection capabilities. We conduct a comprehensive empirical evaluation using both synthetic density estimation and public object detection datasets followed by a real-world robot deployment, which overall shows the effectiveness of the proposed approach.

2 Methodology

Problem Formulation Given an image $\mathbf{x} \in \mathcal{X}$ and a trained object detector F_θ that localizes a set of objects with corresponding bounding box coordinates $\mathbf{b}_i \in \mathcal{R}^4$ as well as class label $y_i \in \mathcal{Y} = \{1, 2, \dots, C\}$, the task is to distinguish if $(\mathbf{x}, \mathbf{b}_i, y_i)$ is **ID**, i.e., drawn from \mathcal{P}_{id} , or **OOD**, i.e., belongs to the unknown distribution \mathcal{P}_{ood} . For conciseness, from now on we omit the suffix i and use y to denote the class label without further notice. As discussed, a powerful **OOD** detection can be obtained via density estimation using **NFs**. This density estimator identifies **OOD** objects with low likelihoods after being trained *only* on data drawn from \mathcal{P}_{id} . Following relevant prior [22, 23], we use the semantically rich logit space (pre-softmax layer) for density estimation. To note that, our method can be readily applied to other (high-dimensional) latent feature spaces.

NFs are known to be universal distribution approximators [10]. That is, they can model a complex target distribution $p(\mathbf{u})$ on a space \mathcal{R}^d by defining \mathbf{u} as a transformation $T_\phi : \mathcal{R}^d \rightarrow \mathcal{R}^d$ from a well-defined base distribution $p_\psi(\mathbf{z})$, where ϕ and ψ are model parameters, respectively:

$$\mathbf{u} = T_\phi(\mathbf{z}) \text{ where } \mathbf{z} \sim p_\psi(\mathbf{z}) \quad (1)$$

where $\mathbf{z} \in \mathcal{R}^d$ and p_ψ is commonly chosen as a uni-modal Gaussian. By designing T_ϕ

to be a *diffeomorphism*, that is, a bijection where both T_ϕ and T_ϕ^{-1} are differentiable, We can compute the likelihood of the input \mathbf{u} *exactly* based on the change-of-variables formula [24]:

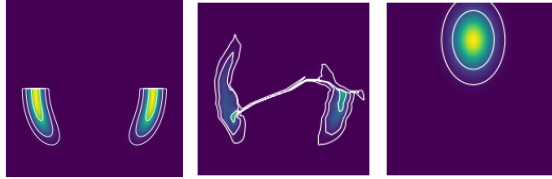
$$p_{\phi,\psi}(\mathbf{u}) = p_\psi(T_\phi^{-1}(\mathbf{u})) |\det(J_{T_\phi^{-1}}(\mathbf{u}))|, \quad (2)$$

where $J_{T_\phi^{-1}}(\mathbf{u}) \in \mathcal{R}^{d \times d}$ is the Jacobian of the inverse T_ϕ^{-1} with respect to \mathbf{u} . When the target distribution is unknown but samples thereof are available, we can estimate the parameter (ϕ, ψ) by minimizing the forward **Kullback-Leibler Divergence (KLD)**, which is equivalent to maximizing the expected **Log-Likelihood (LL)**.

Topological Mismatch However, since the base distribution $p_\psi(\mathbf{z})$ is usually a uni-modal Gaussian (e.g. Figure 2c) and T_ϕ is a diffeomorphism, problems arise for modeling data distribution with different topological properties. These include well-separated multi-modal distributions or distributions with disconnected components (e.g., Figure 2a). For example, one can see that this leads to density filaments between the modes in Figure 2b. Cornish et al. [11] have shown that flows require a bijection with *infinite bi-Lipshitz constant* when modeling a target distribution with disconnected support using a unimodal base distribution. Besides the diminishing modeling performance, this renders the bijection to be numerically "non-invertible", thus, causing optimization instability during training and unreliability of likelihood calculation [14].

2.1 Conditional Resampled Base Distributions

One possible partial mitigation is by enriching the expressiveness of the flows. For example, by (a) increasing the number of layers or parameters, (b) using more complex base distributions, or (c) employing multiple **NFs**, e.g., mixtures of **NFs**. It is important to note that especially (a) and (c) may escalate the computational cost and memory burden. Moreover, scaling the normalizing flow's expressivity, (a) or (c), often does not increase the stability of the optimization [15] or the likelihood calculation. For these reasons, we pursue (b) and attempt to compensate for the complexity of the transformation with the elasticity of the base distribution. In other words, we use a more flexible but efficient base distribution to trade off a costly but sufficiently expressive bijection of the normalizing flow. This way we aim to capture desirable topological properties of the target distribution [17]. Following the prior work [25], to model the fidelitous distribution of data with task-specific conditions, e.g. class labels, we use a class-conditional base distribution. This way we get similar benefits like combining multiple conditional flows (c), however, without having to burden the computational



(a) $p(\mathbf{u}|y=0)$ (b) $p_{\phi,\psi}(\mathbf{u}|y=0)$ (c) $p_\psi(\mathbf{z}|y=0)$
Figure 2: Filament connect modes in the modeled class-conditional distribution (b) if using (trainable) uni-modal base (c) for the multi-modal target (a).

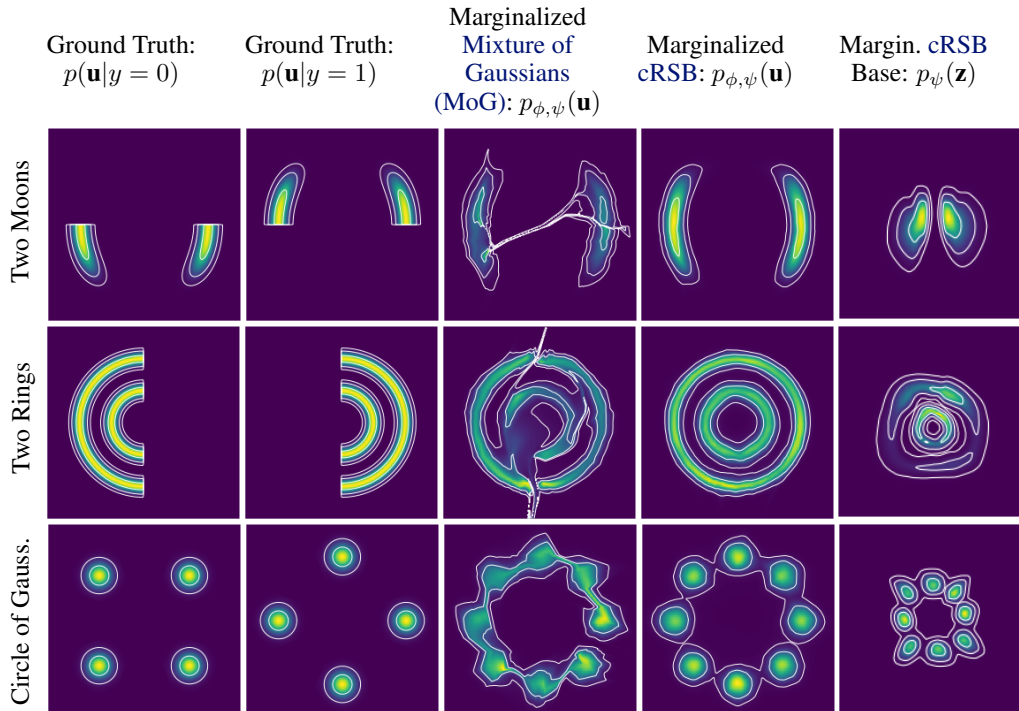


Figure 3: Visualization of density estimation using Real NVP with class conditional MoG, where each class is modeled by a uni-modal Gaussian, and cRSB as well as the class-marginalized density for the base distribution of cRSB.

cost on marginalization over classes. This is because, with (c), this operation requires repeated evaluation of the flows when each flow of the NFs mixture is class-conditional [26]. Even though a class-conditional distribution can specialize on a smaller fraction of the dataset containing similar instances, it will manifest in a multi-modal distribution.

Therefore, we propose to capture the complex topological properties in the target distribution with a more expressive base distribution instead of the uni-model Gaussian. To the end, we introduce cRSB by extending a powerful unconditional base distribution RSB [13] with class-conditional modeling. RSB deforms a uni-modal Gaussian in a learnable manner to obtain more complex distributions via Learned accept/reject sampling (LARS) [27]. LARS iteratively re-weights samples drawn from a proposal distribution $\pi(\mathbf{z})$, e.g. a standard Gaussian, through a learned acceptance function $a_{\psi} : \mathcal{R}^d \rightarrow [0, 1]$. To reduce the computation cost in practice, this process is truncated by accepting the T -th samples if the previous $T - 1$ samples get rejected. To take into account class-conditional information, we conditionalize the learnable acceptance function $a_{\psi}(\mathbf{z}|y)$. As a result, we have the conditional base distribution:

$$p_{\psi}(\mathbf{z}|y) = (1 - \alpha_T) \frac{a_{\psi}(\mathbf{z}|y)\pi(\mathbf{z})}{Z_y} + \alpha_T \pi(\mathbf{z}), \quad (3)$$

where $a_{\psi} : \mathcal{R}^d \rightarrow [0, 1]^C$ and $\alpha_T = (1 - Z_y)^{T-1}$, where $Z_y \in \mathcal{R}$ is the normalization factor for $a_{\psi}(\mathbf{z}|y)\pi(\mathbf{z})$. This factor can be estimated via Monte Carlo Sampling.

In Figure 3, we contrast the density estimation capabilities of NFs with the common MoG [8, 25] base distribution and our cRSB on three tasks with class-conditional structure using an appropriate learning objective (see next section). We find that our cRSB learns appropriate topology-matching base distributions (right outer column) and as a result, the respective NFs do not have adverse effects like filaments between the modes.

2.2 Training with Information Bottleneck

Unfortunately, directly training NFs with a conditional base distribution can lead to underperformance as observed in experiments (see Table 2 and appendix) and reported by Fetaya et al. [25].

We attribute this to the lack of explicit control for the balance between generative and discriminative modeling in the likelihood-based training objective of **NFs**. To alleviate this, we train the normalizing flow with a class-conditional base distribution using the **IB** objective [19]. To abuse the notations, we denote random variables by capital letters such as U, Z, Y , and their realizations by lowercase letters such as $\mathbf{u}, \mathbf{z}, y$. The **IB** minimizes the **Mutual Information (MI)** $I(U, Z)$ between U and Z , while simultaneously maximizing the **MI** $I(Z, Y)$ between Z and Y . Intuitively, the **IB** trades off between the objectives of modeling the class conditional information $p(\mathbf{u}|y)$ with the marginalized density $p(\mathbf{u})$, thus allowing to leverage the class-conditional structure to facilitate more effective density estimation for data characterized with semantic classes.

However, the **IB** is not directly applicable to latent class-conditional distributions in **NFs** since the bijection T_ϕ is lossless by design. Thus, for trading off the class-conditional information with density estimation capabilities, we adapt the approach proposed by Ardizzone et al. [18] for our **cRSB**. Specifically, we inject a small amount of noise ϵ into the input U and hence $Z_\epsilon = T_\phi^{-1}(U + \epsilon)$. Further we define an asymptotically exact version of **MI**, namely the **Mutual Cross-Information (CI)** (more details in appendix):

$$\mathcal{L}_{\text{IBNF}} = CI(U, Z_\epsilon) - \beta CI(Z_\epsilon, Y) \quad (4)$$

$$CI(U, Z_\epsilon) = \mathbb{E}_{p(\mathbf{u}), p(\epsilon)} \left[-\log \sum_{y'} p_\psi(\mathbf{z}_\epsilon | y') - \log |\det(J_{T_\phi^{-1}}(\mathbf{u} + \epsilon))| \right], \quad (5)$$

$$CI(Z_\epsilon, Y) = \mathbb{E}_{p(y)} \left[\log \frac{p_\psi(\mathbf{z}_\epsilon | y)p(y)}{\sum_{y'} p_\psi(\mathbf{z}_\epsilon | y')p(y')} \right], \quad (6)$$

where $\mathbf{z}_\epsilon = T_\phi^{-1}(\mathbf{u} + \epsilon)$, $p(\epsilon) = \mathcal{N}(0, \sigma^2 \mathcal{I}_d)$ is a zero-mean Gaussian with variance σ^2 , and β trades off class information and generative density estimation. With flexible conditional base distributions defined in Eq. 3, we can train the *topology-matching* **NFs** with **IB** by substituting **cRSB** into the conditional base probability $p_\psi(\mathbf{z}|y)$ in Eq. 5 and 6. More noteworthy, we observed that the **IB** is able to regularize the acceptance rate learning for **cRSB** to better assimilate the topological structure of the target distribution, leading to an overall improved performance on accurately approximating the complex target distribution (see Figure 4).

2.3 Detecting OOD Objects

During test time, we detect the **OOD** data based on the predicted **Log-Likelihood (LL)**. To note that, only one forward pass is required to evaluate the acceptance function in **cRSB**. Practically, we use Monte Carlo sampling to estimate the normalization factor Z offline so that no additional computation required for this during inference. We *marginalize* the density over classes for the base distribution defined in Eq. 3 and compute the final **LL** given the logits \mathbf{u}' from the test image:

$$\text{LL}_{\text{test}}(\mathbf{u}') = \log \sum_{y'} (p_\psi(T_\phi^{-1}(\mathbf{u}') | y')) + \log |\det(J_{T_\phi^{-1}}(\mathbf{u}'))|. \quad (7)$$

We then expect **LL** for **ID** objects to be higher than **OOD** ones.

3 Related Work

Normalizing Flows NFs [28] are a popular class of deep generative models. **NFs** have shown applicability in a variety of areas such as image generation [29, 30], uncertainty estimation [31, 32, 33] and **OOD** detection [6, 34, 35]. For **NFs**, one trend has been designing expressive flow-based architectures. Notable examples are affine coupling flows [29, 30], auto-regressive flows [36, 37], invertible ResNet blocks [38] and ODEs-based maps [39]. The major focus of these works is on reducing computing requirements for Jacobian computations while ensuring that each mapping is

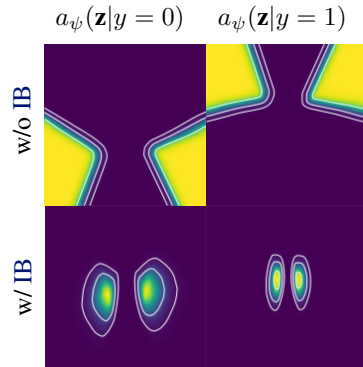


Figure 4: **cRSB** acceptance rate $a_\psi(\mathbf{z})$ w/o and w/ **IB** training for Two Moons.

invertible. Another research direction, currently emerging, is on addressing the topological mismatch [28, 10] of NFs. Targeting this problem, some existing works attempt to increase the learning capacity of the transformation via mixture models [26], latent variable models [11, 40] or injecting carefully specified randomness [41, 12]. These methods may be limited in their applicability to robotics because they either increase memory consumption by expanding the width of transformations or approximate the exact likelihood. Recently, these constraints have been addressed by improving the expressivity of the base distribution [13, 17]. In this paper, we build upon this class of methods since they only add slight computation overheads and thus are well suited for applications in robotics.

Normalizing Flows for OOD Detection NFs have been widely adapted for OOD detection due to its superior density estimation [42]. For example, though with some counter-intuitive observations on raw data space [34], NFs have demonstrated encouraging OOD detection results with additional refinements for raw data [43, 44, 45] or directly based on task-relevant feature embeddings [6, 7, 46, 47]. In this work, we directly apply NFs on the feature space. To note that, another principle direction is to estimate the error bound for this task [48]. Recently hybrid models [49, 7, 50] have shown remarkable performance gain on OOD detection by modeling the joint distribution of both data and its class labels. Such works suggest that class labels can provide useful information. However, directly performing class conditional modeling with NFs for OOD detection results in performance degradation. Tishby et al. [19], Ardizzone et al. [18] mitigate such performance degradation by utilizing IB for training NFs. This explicitly controls the trade-off between generative and discriminative modeling [9]. However, these works on OOD detection utilize NFs without much concern for the fundamental topological problem as the first citizen. Therefore, complementary to these approaches, we examine the problem of topological mismatch of NFs for OOD detection.

OOD Detection in Object Detectors OOD detection research has focused on image classification [42], which may be limited in relevance to robotic vision. In robotics, we may often need both categorization and localization of objects of interest. Therefore, we focus on object detection in open-set conditions here. In this domain, uncertainty estimation [51] has been considered propitious for OOD detection but suffered from computation burdens on runtime [52, 53, 54, 55] or memory costs [56]. To address this, instead of directly applying uncertainty estimation techniques for object detection [54, 2], another popular approach is to explicitly formulate the problem as OOD detection tasks [23, 57, 8, 58, 59]. Amongst them, NFs has been utilized as an expressive density estimator [8, 58]. However, despite the encouraging results, these approaches have not examined the problem of topological mismatch in NFs. As this might prevent additional performance improvements, this work examines the topology-matching NFs for OOD detection in object detectors.

4 Experiments

We next demonstrate the efficacy of our method. First, we evaluate on synthetic density estimation for distributions with distinct topological properties. We then evaluate the OOD detection performance on two object-detection data-sets adapted from their public counterparts [60, 61] for open-set (OS) experiments: Pascal-VOC-OS and MS-COCO-OS based on Glow [30] and a pre-trained Faster-RCNN [20] provided by Miller et al. [23] for a fair comparison. To showcase the practicality, we deploy the one-stage object detector Yolov7 [21] equipped with the proposed method on a real aerial manipulation robot along with the run-time and memory analysis. We empirically found that, to parameterize the acceptance function in LARS, a simple multi-layer perceptron (MLP) (2x128 for density estimation and 3x128 for object detection) is sufficient. We select the hyper-parameters (e.g., $T, \epsilon, \sigma, \beta$) based on the validation set. More details can be found in the supplementary materials.

Datasets and Metrics For density estimation, there are three synthetic datasets: two moons, two rings, and a circle of Gaussians. We employ the KLD between the target and the model distributions to measure the performance. For OOD detection, since existing object detection datasets are not ready for fair evaluation [4], we strictly follow the experimental protocol in [23]. For real robot deployment, we generate $2k$ synthetic images of two objects (a valve and a crawler robot) rendered

based on their CAD models and additionally labeled $2k$ real images. $1k$ synthetic images are used for training and another $1k$ for testing with all real images. We use the **Area Under Receiver Operation Curve (AUROC)** and the **True Positive Rate (TPR)** at different **False Positive Rate (FPR)** (5%, 10%, 20%) as metrics for this task, as they represent the performance of the potential operating points for safety-critical applications, which requires the **FPR** to be sufficiently low.

4.1 Density Estimation

We compare the density estimation performance in Table 1 and provide qualitative results in Figure 3. We find that the **cRSB** base distribution consistently outperforms the class-conditional **Mixture of Gaussians (MoG)**. The performance improvement by **cRSB** can be generalized across two different **NFs** architectures, i.e. Real NVP and NSF.

Table 1: Performance on density estimation for different flow architectures w.r.t. **KLD**, i.e., $D_{\text{KL}}(p(\mathbf{u}, y) \| p_{\phi, \psi}(\mathbf{u}, y))$. Better base distribution is highlighted in bold.

| Flow architecture Base distribution | Real NVP | | NSFs | |
|--|----------|--------------|--------|--------------|
| | MoG_IB | cRSB_IB | MoG_IB | cRSB_IB |
| Two Moons | 1.179 | 1.066 | 0.909 | 0.906 |
| Two Rings | 2.032 | 1.704 | 1.647 | 1.602 |
| Circle of Gaussians | 2.335 | 1.667 | 1.766 | 1.653 |

4.2 OOD Detection in Object Detection

We compare our method (**cRSB_IB**) with both flow-based and non-flow-based approaches. The latter consists of Mahalanobis Distance (MD) [62], Relative Mahalanobis Distance (RMD) [63], GMMDet [23], Softmax, Entropy and, their Deep Ensemble variants with five models [56]. Among flow-based approaches, we have six different base distributions, including unconditional ones (uni-modal Gaussian, **MoG**, **RSB**) and their conditional variants (**MoG_CLS**, **cRSB_CLS**) [25] and **MoG** trained with **IB** (**MoG_IB**) [8, 18]. From Table 2, we can observe that flows with uni-modal Gaussian are able to provide satisfactory performance, i.e., better than most of non flow-based baselines, while flows with more expressive base distributions such as **MoG** and **RSB** can bring more benefits on Pascal-VOC-OS than MS-COCO-OS. When trained with **IB**, the more flexible conditional base distribution (**cRSB_IB**) can mostly have greater performance gains (on both Pascal VOC and COCO) than its strong competitor (**MoG_IB**) (only on COCO) in comparison with their counterparts without **IB** (**MoG_CLS**). These results demonstrate the effectiveness of **cRSB** with **IB** for **OOD** detection in complicated 2D object detection tasks. We further provide the visualization from data before and after the flow transformation with different base distributions in Figure 5, evidencing the ability of matching complex topology of the target data distribution with **cRSB**.

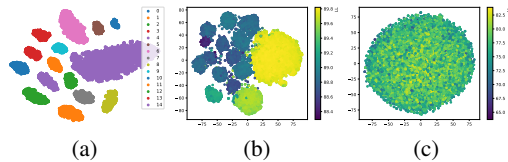


Figure 5: t-SNE visualization for (a) feature embeddings from the object detector (b) latents of the proposed learned base distribution **cRSB** and (c) the uni-modal Gaussian on the training set of Pascal-VOC-OS.

4.3 Real Robot Deployment

Next, we validate the applicability in an application of robotic inspection and maintenance, where it is crucial to avoid false positives of **OOD** objects that appear routinely in outdoor environments. In this experiment, we train Yolov7 with only synthetic images of two objects (a valve and a crawler robot) and deploy on the robot around only real objects. The task is to identify the falsely detected real objects as **OOD** since they are from a distribution different to the synthetic ones. Besides, the performance drop when compared with Table 2 is potentially attributed to the "closer" **OOD** data because the synthetic images are rendered in a highly photorealistic manner. However, our method still outperform other baseline approaches in Figure 6c, where ours can notably achieve higher **TPR** around the low **FPR** region, which are commonly used as operating points for the robot. Computational efficiency is another important requirement. We compare the runtime and space memory consumption against a vanilla Yolov7 using the NVIDIA's embedded GPU called Jetson Orin in Figure 6. The results indicate that the computational overhead of having an **OOD** detector is

Table 2: OOD detection performance on Pascal-VOC-OS and MS-COCO-OS datasets for different methods based on the Faster-RCNN from 3 random runs. The highest values are marked in **bold** and the second highest in *italics*.

| | Pascal-VOC-OS | | | | MS-COCO-OS | | | |
|-----------------------|----------------------|--------------------|--------------------|--------------------|----------------------|-------------------|--------------------|--------------------|
| | AUROC | TPR at | | | AUROC | TPR at | | |
| | | 5%FPR | 10%FPR | 20%FPR | | 5%FPR | 10%FPR | 20%FPR |
| Softmax | 0.901 | 60.1 | 72.8 | 83.1 | 0.882 | 61.3 | 70.6 | 78.1 |
| Entropy | 0.905 | 59.8 | 72.9 | 82.9 | 0.903 | 61.2 | 70.6 | 80.2 |
| MD [62] | 0.9 | 54.1 | 68.8 | 83.3 | 0.902 | 57.2 | 71.4 | 85.5 |
| RMD [63] | 0.838 | 15.2 | 28.4 | 77.4 | 0.531 | 1.7 | 2.6 | 7.1 |
| Ensemble Softmax [56] | 0.885 | 47.8 | 72.6 | 83.1 | 0.898 | 66.2 | 73.5 | 82.3 |
| Ensemble Entropy [56] | 0.887 | 47.8 | 72.5 | 83.1 | 0.906 | 66.2 | 73.5 | 82.3 |
| GMMDet [23] | 0.931 | 70.7 | <i>80.5</i> | <i>89.3</i> | 0.924 | 69.5 | 80.2 | 87.9 |
| Flows Gaussian | 0.915 ± 0.002 | 72.2 ± 0.75 | 77.8 ± 0.89 | 86.1 ± 0.67 | 0.924 ± 0.001 | 68.2 ± 0.73 | 81.2 ± 0.61 | 89.4 ± 0.04 |
| Flows MoG | 0.919 ± 0.002 | 69.0 ± 2.4 | 77.0 ± 2.5 | 86.5 ± 1.2 | 0.925 ± 0.001 | 68.3 ± 0.30 | 80.5 ± 0.50 | 89.6 ± 0.05 |
| Flows RSB [13] | 0.924 ± 0.003 | 72.8 ± 0.88 | 79.3 ± 1.0 | 87.1 ± 0.82 | 0.925 ± 0.001 | 68.6 ± 0.87 | 81.3 ± 0.31 | 89.5 ± 0.34 |
| Flows MoG.CLS [25] | 0.923 ± 0.001 | 69.2 ± 1.5 | 78.2 ± 1.3 | 88.5 ± 0.82 | <i>0.930</i> ± 0.001 | 68.5 ± 0.73 | 82.2 ± 0.31 | 89.7 ± 0.30 |
| Flows MoG.IB [8] | <i>0.934</i> ± 0.002 | 73.1 ± 1.3 | 79.6 ± 0.6 | 87.8 ± 0.2 | 0.924 ± 0.002 | <i>71.1</i> ± 0.9 | 79.6 ± 0.46 | 88.6 ± 0.63 |
| Flows cRSB.CLS | 0.919 ± 0.001 | 72.5 ± 0.37 | 78.8 ± 0.27 | 86.8 ± 0.42 | 0.924 ± 0.001 | 68.3 ± 0.14 | 81.1 ± 0.30 | 89.3 ± 0.18 |
| Flows cRSB.IB (ours) | 0.946 ± 0.003 | 78.5 ± 0.97 | 84.0 ± 0.83 | 90.8 ± 0.76 | 0.934 ± 0.002 | 73.3 ± 2.0 | 84.3 ± 0.40 | 91.3 ± 0.28 |

relatively small when compared to the vanilla YOLOv7. Overall, these experiments validate our claim that our method features efficient runtime inference and cost-effective memory consumption.

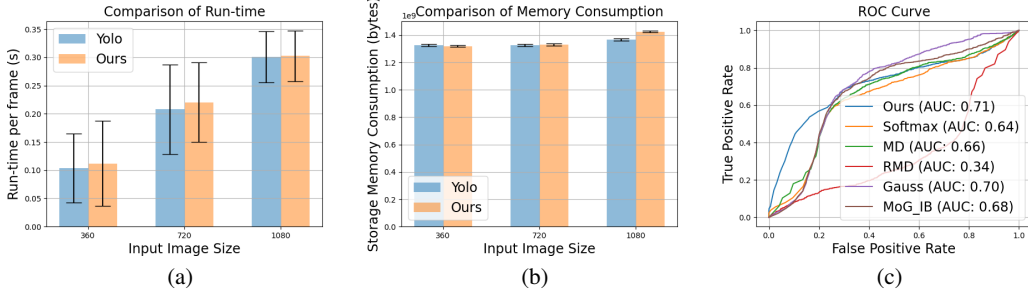


Figure 6: Results from experiments on a real robot. Run-time, memory consumption, and ROC curve are reported. Compared to the vanilla YOLOv7, the proposed method does not yield significant computational costs, while providing performance gains in OOD detection.

5 Limitations

The proposed method is envisioned to work on feature embeddings instead of raw data to counteract the NFs artifacts of assigning higher likelihoods to OOD data [10]. This leads to two limitations. First, it can't directly applied to the tasks/models that could not provide useful feature embeddings extracted from the raw data. Second, its performance is restricted to the quality of features. As reported by previous work [23, 8], learning more compact and centralized features can often lead to increased performance for OOD detection while feature collapse can be harmful to OOD detection. Besides, there are two limitations during deployment. The first is the prolonged initialization time for calculating the normalization factor in LARS based on Monte Carlo sampling. This might not be friendly for applications that require instant response at the beginning. Moreover, the current version of the proposed method does not consider the sequential nature of observations at deployment.

6 Conclusion

To endow robots with introspective awareness against OOD data, we propose the NFs equipped with effective yet lightweight cRSB and train with IB objective. Such NFs are able to mitigate the fundamental topological mismatch problem, facilitating more effective OOD detection capabilities. We present empirical evidence that the proposed method achieves superior performance both quantitatively and qualitatively. To demonstrate the run-time efficiency and minimum memory overheads, we deployed on a real-robot system. Overall, we hope that the results of our work stemming from an enriched base distribution can push forward the direction of NFs-based OOD detection in robot learning.

Acknowledgments

We thank the anonymous reviewers for their thoughtful feedback. Jianxiang Feng and Simon Geisler are supported by the Munich School for Data Science (MUDS). Rudolph Triebel and Stephan Gunnemann are members of MUDS.

References

- [1] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena. Out-of-distribution detection for automotive perception. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2938–2943. IEEE, 2021.
- [2] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel. Trust your robots! predictive uncertainty estimation of neural networks with sparse gaussian processes. In *Conference on Robot Learning*, pages 1168–1179. PMLR, 2022.
- [3] J. Feng, J. Lee, M. Durner, and R. Triebel. Bayesian active learning for sim-to-real robotic perception. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10820–10827. IEEE, 2022.
- [4] A. Dhamija, M. Gunther, J. Ventura, and T. Boulton. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.
- [5] R. Sinha, A. Sharma, S. Banerjee, T. Lew, R. Luo, S. M. Richards, Y. Sun, E. Schmerling, and M. Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.
- [6] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.
- [7] H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 102–117. Springer, 2020.
- [8] R. Li, C. Zhang, H. Zhou, C. Shi, and Y. Luo. Out-of-distribution identification: Let detector tell which i am not sure. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 638–654. Springer, 2022.
- [9] R. Mackowiak, L. Ardizzone, U. Kothe, and C. Rother. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2971–2981, 2021.
- [10] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.
- [11] R. Cornish, A. Caterini, G. Deligiannidis, and A. Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR, 2020.
- [12] H. Wu, J. Köhler, and F. Noé. Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, 33:5933–5944, 2020.
- [13] V. Stimper, B. Schölkopf, and J. M. Hernández-Lobato. Resampling base distributions of normalizing flows. In *International Conference on Artificial Intelligence and Statistics*, pages 4915–4936. PMLR, 2022.

- [14] J. Behrmann, P. Vicol, K.-C. Wang, R. Grosse, and J.-H. Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.
- [15] P. Hagemann and S. Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(8):085002, 2021.
- [16] V. Runde. *A taste of topology*. Springer,, New Delhi:, 2005. Includes bibliographical references and index.
- [17] P. Jaini, I. Kobyzev, Y. Yu, and M. Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR, 2020.
- [18] L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33:7828–7840, 2020.
- [19] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.
- [20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [21] 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*, pages 7464–7475, 2023.
- [22] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating neural network overconfidence with logit normalization. *International Conference on Machine Learning (ICML)*, 2022.
- [23] D. Miller, N. Sünderhauf, M. Milford, and F. Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021.
- [24] V. I. Bogachev and M. A. S. Ruas. *Measure theory*, volume 1. Springer, 2007.
- [25] E. Fetaya, J.-H. Jacobsen, W. Grathwohl, and R. Zemel. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171*, 2019.
- [26] J. Postels, M. Liu, R. Spezialetti, L. Van Gool, and F. Tombari. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. In *2021 International Conference on 3D Vision (3DV)*, pages 1249–1258. IEEE, 2021.
- [27] M. Bauer and A. Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 66–75. PMLR, 2019.
- [28] I. Kobyzev, S. J. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3964–3979, 2020.
- [29] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [30] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.
- [31] B. Charpentier, D. Zügner, and S. Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33:1356–1367, 2020.

- [32] B. Charpentier, O. Borchert, D. Zügner, S. Geisler, and S. Günnemann. Natural posterior network: Deep bayesian uncertainty for exponential family distributions. *arXiv preprint arXiv:2105.04471*, 2021.
- [33] J. Postels, H. Blum, Y. Strümler, C. Cadena, R. Siegwart, L. Van Gool, and F. Tombari. The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*, 2020.
- [34] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.
- [35] S. K. Lind, R. Triebel, L. Nardi, and V. Krueger. Out-of-distribution detection for adaptive computer vision. *arXiv preprint arXiv:2305.09293*, 2023.
- [36] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018.
- [37] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.
- [38] R. T. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.
- [39] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.
- [40] L. Dinh, J. Sohl-Dickstein, H. Larochelle, and R. Pascanu. A rad approach to deep mixture models. *arXiv preprint arXiv:1903.07714*, 2019.
- [41] D. Nielsen, P. Jains, E. Hoogeboom, O. Winther, and M. Welling. Survae flows: Surjections to bridge the gap between vaes and flows. *Advances in Neural Information Processing Systems*, 33:12685–12696, 2020.
- [42] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.
- [43] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.
- [44] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019.
- [45] D. Jiang, S. Sun, and Y. Yu. Revisiting flow generative models for out-of-distribution detection. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=6y2KBh-0Fd9>.
- [46] B. Charpentier, C. Zhang, and S. Günnemann. Training, architecture, and prior for deterministic uncertainty methods. *arXiv preprint arXiv:2303.05796*, 2023.
- [47] J. Feng, M. Atad, I. V. Rodriguez Brena, M. Durner, and R. Triebel. Density-based feasibility learning with normalizing flows for introspective robotic assembly. In *18th Robotics: Science and System 2023 Workshops*, 2023. URL <https://elib.dlr.de/195846/>.
- [48] G. Chou, N. Ozay, and D. Berenson. Safe output feedback motion planning from images via learned perception modules and contraction theory. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 349–367. Springer, 2022.

- [49] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pages 4723–4732. PMLR, 2019.
- [50] S. Cao and Z. Zhang. Deep hybrid models for out-of-distribution detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4723–4733, 2022. doi:10.1109/CVPR52688.2022.00469.
- [51] 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. *Artificial Intelligence Review*, pages 1–77, 2023.
- [52] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.
- [53] J. Lee, M. Humt, J. Feng, and R. Triebel. Estimating model uncertainty of neural networks in sparse information form. In *International Conference on Machine Learning*, pages 5702–5713. PMLR, 2020.
- [54] A. Harakeh, M. Smart, and S. L. Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020.
- [55] J. Feng, M. Durner, Z.-C. Márton, F. Bálint-Benczédi, and R. Triebel. Introspective robot perception using smoothed predictions from bayesian neural networks. In *Robotics Research: The 19th International Symposium ISRR*, pages 660–675. Springer, 2022.
- [56] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- [57] X. Du, G. Gozum, Y. Ming, and Y. Li. SIREN: Shaping representations for detecting out-of-distribution objects. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=8E8tgnYlmN>.
- [58] N. Kumar, S. Šegvić, A. Eslami, and S. Gumhold. Normalizing flow based feature synthesis for outlier-aware object detection. *arXiv preprint arXiv:2302.07106*, 2023.
- [59] X. Du, X. Wang, G. Gozum, and Y. Li. Unknown-aware object detection: Learning what you don’t know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2022.
- [60] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. URL <http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWZ10>.
- [61] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2014. URL <http://arxiv.org/abs/1405.0312>. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
- [62] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.
- [63] J. Ren, S. Fort, J. Liu, A. G. Roy, S. Padhy, and B. Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

A.3. Publication 3

Jianxiang Feng, Jongseok Lee, Maximilian Durner, Rudolph Triebel (2022): “*Bayesian Active Learning for Sim-to-Real Robotic Perception*”. In the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- <https://ieeexplore.ieee.org/abstract/document/9982175/>

Please refer to the final published version for citation:

```
@inproceedings{feng2022bayesian,
  title={Bayesian Active Learning for Sim-to-Real Robotic Perception},
  author={Feng, Jianxiang and Lee, Jongseok and Durner, Maximilian and
    Triebel, Rudolph},
  booktitle={2022 IEEE/RSJ International Conference on Intelligent Robots
    and Systems (IROS)},
  pages={10820--10827},
  year={2022},
  organization={IEEE}
}
```

Bayesian Active Learning for Sim-to-Real Robotic Perception

Jianxiang Feng^{1,2}, Jongseok Lee¹, Maximilian Durner^{1,2} and Rudolph Triebel^{1,2}

Abstract— While learning from synthetic training data has recently gained an increased attention, in real-world robotic applications, there are still performance deficiencies due to the so-called Sim-to-Real gap. In practice, this gap is hard to resolve with only synthetic data. Therefore, we focus on an efficient acquisition of real data within a Sim-to-Real learning pipeline. Concretely, we employ deep Bayesian active learning to minimize manual annotation efforts and devise an autonomous learning paradigm to select the data that is considered useful for the human expert to annotate. To achieve this, a Bayesian Neural Network (BNN) object detector providing reliable uncertainty estimates is adapted to infer the informativeness of the unlabeled data. Furthermore, to cope with misalignments of the label distribution in uncertainty-based sampling, we develop an effective randomized sampling strategy that performs favorably compared to other complex alternatives. In our experiments on object classification and detection, we show benefits of our approach and provide evidence that labeling efforts can be reduced significantly. Finally, we demonstrate the practical effectiveness of this idea in a grasping task on an assistive robot.

I. INTRODUCTION

Over the last years, the performance of computer vision increased sharply, leading to the urge of employing such approaches on robotic vision tasks such as object classification, detection [1], [2] and pose estimation [3]. In this context, the necessity of large amounts of annotated, task-related training data is a main issue, particularly for tasks relying on semantic features such as object classification or detection. Therefore, a compelling solution is to learn from synthetic data. Like this, large amount of annotated data can be obtained from simulation with relatively less time and manual efforts [4]–[6]. With the emergence of open-source image synthesizing pipelines [7], [8], this solution becomes even more accessible in practice. However, although these pipelines continue improving in fidelity and become more photo-realistic, there are subtle but important differences between simulation and real domain. This leads to the so-called *Sim-to-Real gap* which is the main barrier to transfer this technique to real world robotic perception. Several works address this gap by applying techniques such as Domain Randomization (DR) [3], [6] and Domain Adaptation (DA) [4], [9] with certain improvements. Yet, the unpredictable variability of real-world scenes prevents a complete elimination of the reality gap [10].

We encounter similar issues in our real lab environment, when deploying an object detector [1] trained on photo-realistic images on our robotic platform EDAN [11]. From

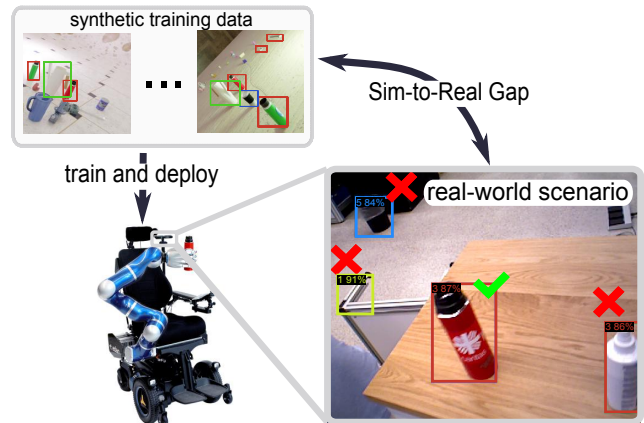


Fig. 1. Illustration of a practical problem. Deploying a detector trained with only synthetic images on real-world scenarios leads to under-performances. These inaccuracies (denoted by red crosses) such as false positives are due to the Sim-to-Real gap and for this, a few informative real images can improve the performance. Therefore, this work investigates the question on how to collect such informative real images via active learning.

our practical experience, variables such as sensor characteristics, illumination, or textures cannot be modeled to precisely match the real environments. Even after a careful tuning of DR, we find that the object detector fails to generalize well in the real-world scenario (e.g., clutter in lab environment see Fig. 1). To overcome this, we applied domain-oriented fine-tuning, by using real data of the underlying robotic application, like [12]. Hence, based on our use-case, we advocate that real data is indispensable for a robot to robustly adapt from simulation to real world.

This however, comes with the requirement of tedious, time consuming manual labeling. In this work, we investigate on the question: *How to bridge the Sim-to-Real gap with minimum annotation efforts?* Having a model trained on synthetic images, we propose an Active Learning (AL) pipeline that can efficiently bridge the still present Sim-to-Real gap. In contrast to our previous work [13], we here aim for autonomous acquisition of as few annotated real images as possible. Based on a deep Bayesian Active Learning (AL) framework [14], [15], we analyze different strategies to select the most informative data samples. Further we devise a simple yet effective strategy to mitigate the lack of diversity in the selected data, caused by the label distribution shift between simulation and real domain [16], [17]. Note, that the latter is important for performance gain in AL under domain shift (simulation vs. reality domain in our case) [18], [19]. Moreover, for the more challenging 2D object detection task, we suggest to incorporate regression uncertainty into the selection process due to its multi-task characteristic

¹ Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. email: jianxiang.feng@dlr.de
² Technical University of Munich (TUM), 80333 Munich, Germany.

(including both classification (*cls*) and regression (*reg*)).

Concretely, we train a BNN with synthetic images, which can be obtained from another task-relevant data set or generated by photo-realistic image synthesizers. In a second step, a pool of task-specific real images are forwarded to the model. According to the scores from the acquisition function and a sampling strategy, a small subset of samples is selected and solicited for human annotations. The labeled data is then used to adapt the model. The aforementioned procedures can be repeated iteratively until the desired performance is achieved.

Besides the empirical validation of the proposed idea on a classification task, we then conduct evaluation on two more challenging 2D object detection data sets, one with large Sim-to-Real domain shift and another with less to show that the proposed idea can help bridge the gap in a cost-effective way, significantly better than the random baseline and competitive against the state-of-the-art approaches. In addition, we provide a failure case on a third object detection data set to help identify the working scenarios of the proposed idea. To demonstrate the practicality and effectiveness, we further deploy the pipeline on our real robot and show a significantly positive impact of the visual perception within grasping as downstream task.

In summary, the main contributions of this work are:

- we propose to actively and efficiently close the Sim-to-Real gap by applying a BNN in an Active Learning (AL) framework.
- we introduce a simple yet effective sampling strategy to mitigate the label distribution shift in Bayesian AL under domain shift.
- we conduct experiments to empirically show the positive impact of the proposed pipeline on both object classification and 2D object detection tasks, clearly outperforming the random baseline and closely competing against the state-of-the-art approaches
- we demonstrate the applicability in reducing labeling efforts on a real robotic system.

Importantly, the accompanying video provides qualitative results including the demonstration on an assistive robot. The code of the implementation will be publicly available¹.

II. RELATED WORK

a) Sim-to-Real Transfer: Sim-to-Real transfer is mainly tackled with DR and DA. The former treats the real test scenario as one instance of many synthetic ones generated by randomizing the parameters of the synthesizer such as materials, lightening, backgrounds, and plausible geometric configurations [20], [21]. In contrast, DA focuses on learning domain-invariant representations across the different domains (e.g. synthetic and real domain in this context) by sometimes including data of the target domain [4]. Though DA has achieved impressive performance, as mentioned by different researchers, when only relying on unlabeled data, the domain gap is hard to diminish both in theory [9] and in practice [22], [23]. Considering this issue, the paradigm of active

learning is appealing to address the reality gap by utilizing annotated real data in an efficient way. In pool-set based active learning [24], the aim is to reach certain level of performance with as less data as possible. In case of supervised learning, the data is selected based on their informativeness, which can be measured by different quantities such as the output uncertainty, the disagreement of a committee, or the expected model change [25], [26]. We also stress that active learning is complementary to the aforementioned techniques. While recent works such as [18], [19] argue for the fusion of DA and active learning to obtain better performance, we additionally use DR in this work. Nevertheless, none of them considers employing BNNs for this purpose and most of them focus on classification tasks, which are less relevant for the robots in the real world. Wen *et al.* [27] apply BNNs for DA, but they only focus on conventional passive learning paradigm and classification tasks. We aim to study the active learning paradigm for Sim-to-Real transfer on a more challenging real-world object detection task, which is arguably more relevant for various use-cases of the robots.

b) Active Learning for Object Detection: In the context of active learning for object detection, specific metrics related to characteristics of the underlying network can be applied [28]. While in [29] the margin of the bounding box scores in different layers is used, Kao *et al.* [30] consider the localization tightness and stability. Meanwhile, uncertainty based approaches [19], [25], [31] are also able to achieve competitive performances in the field of object detection. Most of uncertainty based approaches are built on BNNs [14], [32] which can produce more reliable uncertainty estimates. Along with its theoretic soundness, the task-agnostic characteristic of these approaches can facilitate wider applicability for different fields. While some only exploit the classification branch for the uncertainty estimation [33], [34], others [15] consider both classification and regression branches. Yet, they rely on larger amount of annotated real world data to initialize the training of the model and update the model in each iteration, while we assume relatively small amount of real data.

III. PROBLEM FORMULATION AND OVERVIEW

We consider two domains: the simulation domain S and the real domain R . In S , we assume the availability of annotated data set, i.e., given the synthetic data \mathbf{x}^S and annotated labels \mathbf{y}^S , we denote the synthetic data set as $\mathcal{D}_S = \{(\mathbf{x}_i^S, \mathbf{y}_i^S)\}_{i=1}^{N_S}$ where N_S is the number of data points. In contrary, R contains an unlabeled data set $\mathcal{D}_T = \{(\mathbf{x}_i^R)\}_{i=1}^{N_R}$ which constitutes of N_R number of real images \mathbf{x}^R . We further extend the notations to define an object detection task including classification (*cls*) and regression (*reg*) tasks. Given the space of inputs \mathcal{X} (both synthetic and real images) and outputs \mathcal{Y} (sets of object classes \mathbf{c} and their 2D location as bounding boxes \mathbf{b}), we define the object detector as a function $\mathcal{M}_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters θ . Naturally, our objective is to obtain an object detector in the real domain R , for which synthetic data \mathcal{D}_S can be exploited.

¹<https://github.com/DLR-RM>

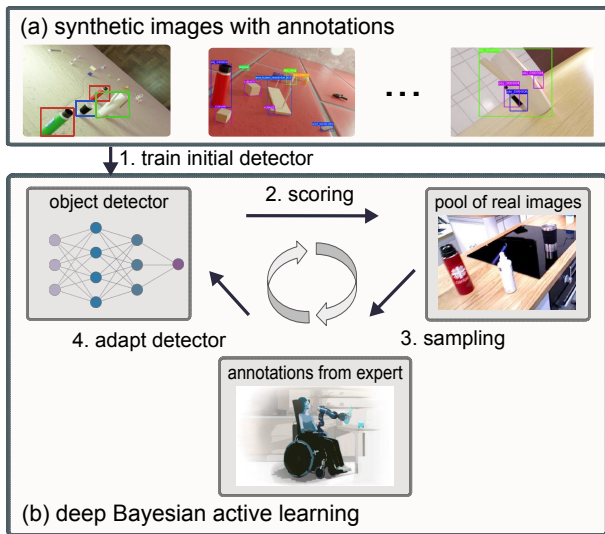


Fig. 2. **The proposed Sim-to-Real pipeline.** Using labeled synthetic images, we first train an initial BNN object detector. Then, we rely on deep Bayesian AL to select the most informative images from a pool of unlabeled real images. The scoring of all the images in the pool is obtained via an acquisition function, while sampling is applied to deal with the foreground class imbalance problem. Based on the selected images, the human expert performs the annotation and the detector is adapted via fine-tuning. The process is repeated to close for Sim-to-Real transfer.

To achieve this goal, the proposed pipeline (depicted in Fig. 2) relies on deep Bayesian AL. What motivates our approach is that in practice, this so-called Sim-to-Real transfer can be achieved by combining (a) the large amounts of annotated synthetic data, and (b) a few but the most informative real images with annotations from human expert. Importantly, we conjecture that the real images can bridge the reality gap in a simple and effective manner, and thus, this work focuses on reducing the amounts of needed real images. For this, as shown in Fig. 2, (i) we use \mathcal{D}_S to train an initial model with domain randomization. (ii) Then, treating the unlabeled real data \mathcal{D}_T as a pool set \mathcal{D}_{pool} , we rank the informativeness of each images with an acquisition function $\mathcal{A}(\cdot)$ and then (iii) apply a sampling strategy to create the subset. (iv) The labels of this subset is queried to a human expert for manual annotation. This process can be repeated for multiple times until the reality gap is diminished. Next, we describe and motivate these steps in detail.

IV. THE PROPOSED PIPELINE

This section describes our pipeline of Sim-to-Real transfer for 2D object detection. The main components are a BNN object detector for uncertainty quantification (Sec. IV-A), and deep Bayesian AL framework (Sec. IV-B).

A. Bayesian Neural Networks for Object Detection

We choose to model the object detector \mathcal{M}_θ as a BNN, in order to obtain its uncertainty estimates. BNNs achieve this by reasoning about the model uncertainty, which indicates *what the model does not know*. Reasoning about the model uncertainty, the AL framework can later leverage this information to label the most uncertainty data to the model itself.

To do so, given the training data \mathcal{D}_{train} and a test data sample \mathbf{x}^* , BNNs produce the output distribution $p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}_{train})$ by marginalizing over the models' distribution:

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}_{train}) = \int p(\mathbf{y}^* | \mathbf{x}^*, \theta) p(\theta | \mathcal{D}_{train}) d\theta. \quad (1)$$

In (1), $p(\mathbf{y}^* | \mathbf{x}^*, \theta)$ is the observation likelihood, and $p(\theta | \mathcal{D}_{train})$ is the distribution over the weights θ . As a closed form solution to the integral in (1) does not exist, the Monte-Carlo integration is often used for a numerical solution [35]. As a note, our AL pipeline uses both the synthetic and the annotated real images as the training set \mathcal{D}_{train} , and the new images \mathbf{x}^* are samples from the pool set \mathcal{D}_{pool} .

However, applying BNNs to the existing anchor-based detectors such as Retinanet [1] requires several adaptations [15], [33]. This is due to their post-processing steps, i.e., (i) *miss-correspondence between the anchor predictions and final outputs*, and (ii) *hard cut-off behavior in non-maximum suppression (NMS) step*. For these, the BayesOD framework [15] can be used, which performs Monte-Carlo sampling for each anchor prediction before NMS steps, and relies on Bayesian inference to infer the output distributions. Intuitively, BayesOD clusters outputs in anchor level using spatial affinity. To explain, assume that such cluster contains M anchors and consider the highest classification score as center of this cluster (indexed by 1). The other outputs are considered as measurements to provide information for the center, denoted by $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{b}}_i$. By further denoting the final predictive distributions for *cls* and *reg* of this cluster as $p_{[\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_M]}(\mathbf{c} | \mathbf{x}^*, \mathcal{D}_{train})$ and as $p_{[\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_M]}(\mathbf{b} | \mathbf{x}^*, \mathcal{D}_{train})$ respectively, the final output distributions are computed as:

$$p_{[\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_M]}(\mathbf{c} | \mathbf{x}^*, \mathcal{D}_{train}) \propto p_{\hat{\mathbf{c}}_1}(\mathbf{c} | \mathbf{x}^*, \mathcal{D}_{train}) \prod_{i=2}^M p(\hat{\mathbf{c}}_i | \mathbf{c}, \mathbf{x}^*, \mathcal{D}_{train}),$$

$$p_{[\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_M]}(\mathbf{b} | \mathbf{x}^*, \mathcal{D}_{train}) \propto p_{\hat{\mathbf{b}}_1}(\mathbf{b} | \mathbf{x}^*, \mathcal{D}_{train}) \prod_{i=2}^M p(\hat{\mathbf{b}}_i | \mathbf{b}, \mathbf{x}^*, \mathcal{D}_{train}).$$

Here, $p_{\hat{\mathbf{c}}_1}(\mathbf{c} | \mathbf{x}^*, \mathcal{D}_{train})$ represents the per-anchor predictive distribution of the cluster center, while $\prod_{i=2}^M p(\hat{\mathbf{b}}_i | \mathbf{b}, \mathbf{x}^*, \mathcal{D}_{train})$ is the likelihood of each cluster member given the output. When we choose the Gaussian and Categorical distributions for *cls* and *reg* tasks respectively, the sufficient statistics of them such as mean and covariance matrix can be computed analytically. We refer more details in [15] and next, we discuss the AL framework that relies on the BayesOD framework.

B. Bayesian Active Learning for Sim-to-Real

With the uncertainty estimates of an object detector, the AL pipeline needs to choose the images for annotation. This selection of images is done via an acquisition function. Moreover, due to the domain shift between S and R , a sampling strategy also needs to be devised to mitigate the bias in the selected data set. We describe below these components and our design choices.

1) *Acquisition Function*: We define the acquisition function based on the uncertainty estimates from the BNN detector. In this step, the acquisition function is used to obtain the informativeness scores for each detected instance on one image, and then *aggregated* into one final score to represent the informativeness of the entire image. Once the scores are obtained for all the images in the pool set \mathcal{D}_{pool} , we sample a subset of them for annotation (IV-B.2) in order to adapt the model. Specifically, we consider uncertainty from both *category classification* and *bounding box regression*, which are referred to as semantic and spatial uncertainty respectively [36]. For the semantic uncertainty of the j -th detection instance on an image, given the Shannon Entropy measure $\mathcal{H}(\cdot)$, the *cls* acquisition function $\mathcal{U}_{j,cls}$ is modeled with a Bernoulli distribution as:

$$\begin{aligned} \mathcal{U}_{j,cls} &= \sum_{i=1}^{|\mathcal{C}|} \mathcal{H}(p(c_i|\mathbf{x}^*, \mathcal{D}_{train})), \\ &= \sum_{i=1}^{|\mathcal{C}|} [-p(c_i|\mathbf{x}^*, \mathcal{D}_{train}) \log p(c_i|\mathbf{x}^*, \mathcal{D}_{train}) \\ &\quad - (1 - p(c_i|\mathbf{x}^*, \mathcal{D}_{train})) \log (1 - p(c_i|\mathbf{x}^*, \mathcal{D}_{train}))]. \end{aligned} \quad (2)$$

In (2), the steps follows from the definition of the entropy, and optimizing the given measure is equivalent to maximizing the information gain [37] or information content.

The uncertainty from regression is defined as differential entropy of $p(\mathbf{b}|\mathbf{x}^*, \mathcal{D}_{train})$ which is approximated by a multivariate Gaussian with covariance matrix \mathbf{C}_b calculated from the samples of predicted bounding boxes:

$$\begin{aligned} \mathcal{U}_{j,reg} &= \mathcal{H}(p(\mathbf{b}|\mathbf{x}^*, \mathcal{D}_{train})) \\ &= \frac{k}{2} + \frac{k}{2} \ln(2\pi) + \frac{1}{2} \ln(|\mathbf{C}_b|), \end{aligned} \quad (3)$$

where k is the dimensionality of random variable \mathbf{b} . Again, this regression acquisition function $\mathcal{U}_{j,reg}$ follows from the definition of entropy for Gaussian distributions, and represents the information content of an image.

We choose to exploit these two quantities by a combination function $comb(\cdot)$, in order to produce the uncertainty score for each of N_k detected instance on k -th image. Then, the acquisition function for k -th image \mathcal{A} is defined by aggregating scores with a function $agg(\cdot)$ denoted by:

$$\mathcal{A}(\mathbf{x}_k) = agg_{j \in N_k}(comb(\mathcal{U}_{j,cls}, \mathcal{U}_{j,reg})), \quad (4)$$

The combination function $comb(\cdot)$ can be a weighted sum (*sum*) or maximum (*max*) operation [31]. The aggregation function $agg(\cdot)$ can be a maximum (*max*), summation (*sum*) or average (*avg*) operation [29]. What motivates this is the problem itself, i.e. object detection involves both *cls* and *reg* tasks and multiple instances in one image.

2) *Sampling Strategy*: One problem in the naive TopN sampling motivates us to combine the TopN sampling with the popular sub-sampling technique [38]. The problem is the violation of the assumption that the simulation domain S and the real one R are the same, which does not hold in fact. This will further lead to a performance degradation for both AL and object detection training [39], [40]. More specific, to

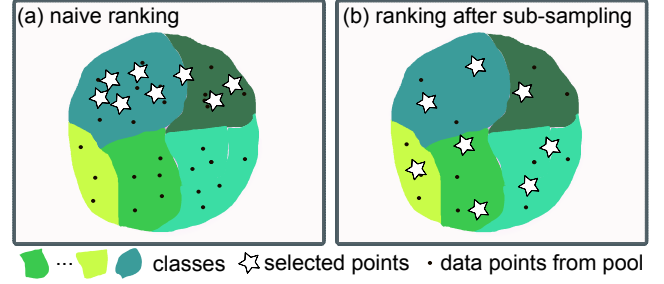


Fig. 3. **Sub-sampling Strategy**. We illustrate the ranking after sub-sampling strategy. A naive ranking selects the most informative points from a few classes of pool data, while the ranking after sub-sampling enables to evenly select the most informative points across the variety of classes. This mitigates the class imbalance problem of AL for object detection, and introduced diversity can improve the performance.

select the B most informative images scored based on the model trained on S will result in an imbalance problem in the selected data set. Since the algorithm queries only images from real domain R , we attribute the under-performance during AL to the label distribution shift [16]. To explain, we denote the distribution followed by sub-sampling as $P_{ss}(\mathbf{c}, \mathbf{b})$ and the distribution followed by uncertainty sampling as $P_{unc}(\mathcal{A}(\mathbf{c}, \mathbf{b}))$, which can be a product of delta distribution with probability mass placed at the top B scored predictions. Therefore, the selected data during AL follow the a label distribution $P_{ss}P_{unc}$. Additionally, we use $P_r(\mathbf{c}, \mathbf{b})$ for the real label distribution, which is assumed to be uniform. The goal is to adapt the model with data points drawn from P_r , which is unavailable for unlabeled data. Instead we adapt the model with data points drawn from $P_{ss}P_{unc}$, which ideally should be aligned with P_r . Unlike classification case, in which the label distribution lies in a discrete finite space and importance weighting correction [17] can be easily adapted, the label space for object detection is more complex when there is an additional regression task involved. The trade-off between alleviation of label distribution shift and utilization of information contained in the uncertainty estimates is thus determined by the distribution form of P_{ss} and the amount of data to be sub-sampled. Intuitively, by assuming there is certain degree of redundancy in the data set, we select the uniform distribution for P_{ss} , which works empirically well, shown in the experiments. In practice, the pool set data is filtered by P_{ss} first, and then with P_{unc} , the learner thus can choose by considering the informativeness in the sub-sampled data. An illustrative explanation on the class imbalance problem, one instance of label distribution shift, is shown in Fig. 3.

V. EXPERIMENT

In this section, we first validate the proposed sampling strategy on a classification task, in which the model is transferred from MNIST [41] to MSNIST-M [42]. Then we move on to two *more challenging but task-relevant* self-collected data-sets on *2D object detection*. To note that, we employ two data-sets with different magnitudes of Sim-to-Real gap (one is large and the other small) to demonstrate that the proposed pipeline can efficiently bridge the gap for

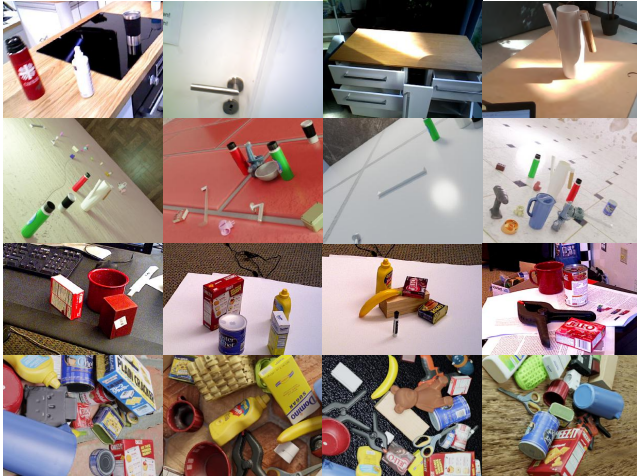


Fig. 4. **The real and synthetic data.** Exemplary images from real (1st,3rd row) and synthetic domain (2nd, 4th row) of EDAN (1-2 rows) and YCBV (3-4 rows) data sets.

both cases. In all experiments, we instantiate the *Sim-to-Real gap* by subtracting the performance of the corresponding models trained on purely the real and simulated data-set. Nevertheless, we address the limitation of the proposed idea by including one *failure case* on the public YCBV data set [43] to further identify the operational scenario. In the end, we show the practical effectiveness of our idea by deploying the model on an assistive robot within a grasping task. The implementation details and parameter settings of the proposed pipeline are then provided, which is followed by results and discussions.

a) *Data sets:* (1) **Digits** include MNIST and MNIST-M digit data sets with 10 classes. MNIST-M contains digits from MNIST but blended with random color patches. We can treat MNIST-M as MNIST digits in real-world in this case and perform Sim-to-Real transfer for them. (2) **EDAN** includes 5 classes: ikea bottle, watering can, door handle, drawer handle and grey mug. With simple textures and geometry of the objects and the *indoors lab environments* (see Fig. 4), the domain gap on this data set is small. (3) **SAM** [44] includes 3 classes: cage, pipe and hook. With more complex textures and geometry of the objects and different weather conditions in *outdoor environments*, the domain gap on this data set is much larger than EDAN. (4) **YCBV** contains images of 21 classes from common objects such as pitcher, sugar box and so on. Basic information of the aforementioned data sets is summarized in Table I and the synthetic data sets except for the one of 1 are generated by BlenderProc [7] with domain randomization applied.

b) *Baselines:* In order to validate the proposed idea, we compare with the following baselines. (1) **Random:** an approach to randomly select data points for query in each iteration. (2) **Batch-bald** [26]: an approach to query a batch of data with *jointly maximum mutual information* instead of individually. (3) **Clue** [19]: an approach for active domain adaptation that considers both *diversity* and *uncertainty* in the acquisition function. (4) **Coreset** [45]: a *diversity-oriented*

TABLE I
BASIC INFORMATION AND TRAINING HYPER-PARAMETERS ON FOUR DATA SETS

| Data set (size of sim, real-pool, real-val, real-test set, number of class) | Query Size (image) | Maximum Training Period during AL (epoch) | Learning Rate | Network Architecture |
|---|--------------------|---|--------------------------------------|----------------------|
| Digits data set (60k, 55k, 5k, 10k, 10) | 20 | 50 | linearly from $1e^{-5}$ to $1e^{-3}$ | the same CNN in [26] |
| EDAN (10k, 0.5k, 0.1k, 1k, 5) | 20 | 10 | $1e^{-4}$ | RetinaNet [1] |
| SAM (2.5k, 2k, 0.1k, 0.5k, 3) | 80 | 10 | $1e^{-4}$ | RetinaNet [1] |
| YCBV (50k, 1.4k, 0.1k, 0.5k, 21) | 50 | 10 | $1e^{-3}$ | RetinaNet [1] |

TABLE II
COMPARISON OF DIFFERENT ACQUISITION FUNCTIONS: MEAN MAP OVER 10 ITERATIONS FOR DIFFERENT AGGREGATION AND COMBINATION FUNCTIONS WITH AND WITHOUT SUB-SAMPLING STRATEGY AND WITH 10 AND 30 SAMPLES ON EDAN DATA SET.

| Agg. | Comb. | 10 samples | | 30 samples | |
|------|-------|------------|---------------|------------|---------------|
| | | w.o. sub. | w. sub. | w.o. sub. | w. sub. |
| Avg | Max | 74.73% | 76.51% | 74.47% | 76.76% |
| | Sum | 74.77% | 77.09% | 75.17% | 77.19% |
| Sum | Max | 75.80% | 74.54% | 76.08% | 76.76% |
| | Sum | 71.83% | 75.35% | 74.31% | 76.67% |
| Max | Max | 73.67% | 72.67% | 73.02% | 74.74% |
| | Sum | 75.36% | 76.89% | 74.98% | 77.49% |

approach for AL, whose greedy version is a k-center algorithm. For *clue²* and *batch-batch³*, we use the open-sourced implementation and only apply to 1 with *max* aggregation function due to their iterative calculation characteristic. For efficiency within *coreset* and *clue*, we use the logits layer as latent features.

c) *Implementation details:* Training hyper-parameters are summarized in Table I. Within the *sum* combination function, we set the weight of 1 to 1 on all data sets. For regression, we select 0.01 for EDAN and SAM, 0.001 for YCBV. The percentage of sub-sampling is set to 1% for digits data sets and 50% for the others based on the performance on validation set. We set dropout rate to 0.1 in BayesOD and apply Bayesian inference *only for bounding box regression* instead of both heads to avoid under-performance observed in preliminary experiments. We use 100 Monte-Carlo samples to approximate the joint distribution in batch-bald. Regarding the evaluation metric, following the convention in [19], we employ the mean accuracies for classification and mean MAP for object detection over AL iterations.

A. Results and Analysis

a) *Design choices:* We firstly conduct an initial empirical study on the effects of aggregation and combination functions in Eq. (4), number of samples to approximate Eq. (1) on EDAN data set (Table II). We can observe: 1. More weight posterior samples can lead to slightly better results;

²<https://github.com/virajprabhu/CLUE>

³<https://github.com/BlackHC/BatchBALD>

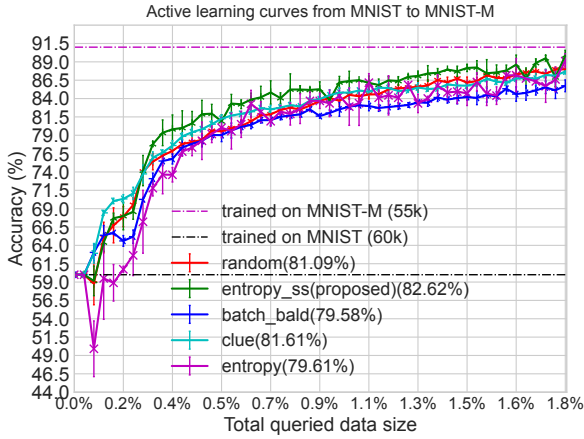


Fig. 5. **Results on digits data sets (MNIST \rightarrow MNIST-M).** Active learning curves of 3 random runs (with 50 iterations and 20 images queried in each iteration). The black and purple dotted lines represent the performance (on MNIST-M test set) of model trained on MNIST and MNIST-M training set with size in the parentheses, respectively. The compared methods include the proposed one (*entropy_{ss}*) and other baselines. Values in the parentheses are mean accuracies over 50 iterations.

TABLE III

RESULTS SUMMARY FOR OBJECT DETECTION DATA SETS. VALUES IN THIS TABLE ARE 1. PERCENTAGE OF ANNOTATED IMAGES REQUIRED TO BRIDGE SIM-TO-REAL GAP (LOWER THE BETTER) AND 2. MEAN MAP OVER 10 ITERATIONS WITHIN AL (HIGHER THE BETTER).

| | Random | Proposed | Coreset | Clue | Batch-bald |
|------|---------------------------|---------------------------|---------------------------|---------------------------|---------------|
| EDAN | > 40% / 75.7% | 36% / 77.1% | > 40% / 75.0% | > 40% / 75.7% | > 40% / 72.9% |
| SAM | > 40% / 81.4% | 32% / 82.2% | 20% / 85.6% | 32% / 85.0% | > 40% / 82.0% |
| YCBV | 40% / 65.2% | > 40% / 63.5% | > 40% / 61.1% | 40% / 65.2% | > 40% / 64.8% |

2. The sub-sampling strategy can improve performance most of the cases; 3. When using *avg* and *max* to aggregate uncertainties of detections on the image, the *sum* combination function yields better results; Only within *sum* aggregation function, the *max* operation outperforms.

In general, the setting pairs of *max* + *sum* and *avg* + *sum* provide the best results. As this ablates our design choices, we use this insight and mainly focus on these two settings with 10 samples and report only the one with better results.

b) *Results on digits data sets:* In Fig. 5, we can see that the domain gap can be bridged with $\sim 2\%$ data by the proposed sub-sampling strategy (*entropy_{ss}*), faster and better than the *random* and *clue* baseline. In contrast, the naive *entropy* and *batch_bald* perform worse than *random* along with large variations. This shows that the proposed sampling strategy for mitigating distribution shift in AL is able to provide greater performance gain than the one considering trade-off between *uncertainty* and *diversity*.

c) *Results on EDAN data set:* In Fig. 6 and Table III, we can learn that the gap can be eliminated by the proposed method with *avg* to aggregate detections and *sum* to combine *cls* and *reg* uncertainties (*avg_{sum_{ss}}*) with only 36% data, outperforming both the strong baseline *random* and *clue*. In

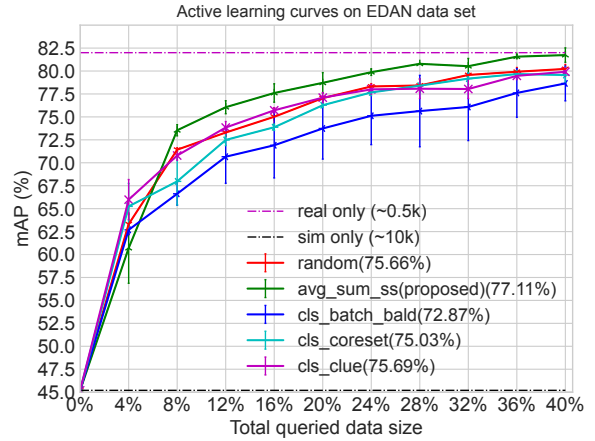


Fig. 6. **Results on the EDAN data set.** Active learning curves of 3 random runs (with 10 iterations and 20 images queried in each iteration). The black and purple dotted lines represent the performance of model trained on sim and real data sets with size in the parentheses, respectively. The compared methods include the proposed one (*avg_{sum_{ss}}*) and other baselines. Values in the parentheses are mean mAP over 10 iterations.

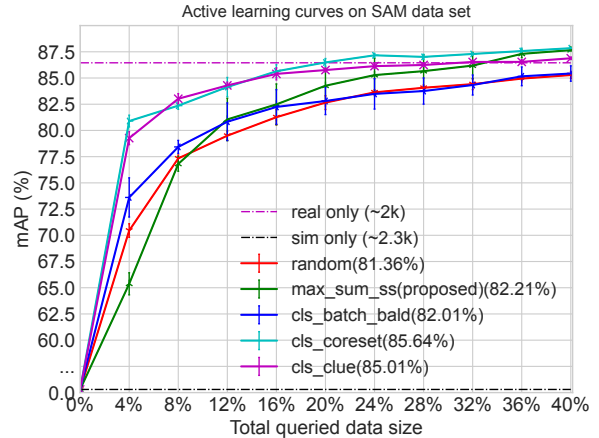


Fig. 7. **Results on the SAM data set.** Active learning curves of 3 random runs (with 10 iterations and 80 images queried in each iteration). The black and purple dotted lines represent the performance of model trained on sim and real data sets with size in the parentheses, respectively. The compared methods include the proposed one (*max_{sum_{ss}}*) and other baselines. Values in the parentheses are mean mAP over 10 iterations.

contrary, while *clue* is on a pair with *random* and slightly better than *coreset*, *batch_bald* has the lowest mean mAP. This demonstrates that utilization of information from both *cls* and *reg* with sub-sampling is advantageous in the case of data set with moderate distribution shift like EDAN.

d) *Results on SAM data set:* The final detector on this data set can achieve a quite decent mAP ($> 90\%$), therefore in this experiment we aim to bridge the gap up to a sufficient level, which is 95% of gap. In Fig. 7 and Table III, the proposed method with *max* as aggregation and *sum* as combination function followed by sub-sampling strategy (*max_{sum_{ss}}*) is still able to beat the strong *random* baseline as well as *batch_bald* and diminish the gap. Nevertheless, *clue* and *coreset* perform better than *max_{sum_{ss}}* probably

due to the larger domain gap on this data set. It could attribute to the reason that our proposed sampling strategy aims to compensate the shift in label distribution, thereby less effective for a large shift in the input distribution.

e) Limitation Analysis: In this sub-section, we show a failure case on YCBV data set to demonstrate the limitation of the proposed idea. With this, we aim to identify the operational scenarios of AL for Sim-to-Real transfer and highlight the characteristic of this problem with the hope of providing some enlightening thoughts for the community.

In the last row of Table III, we see that all approaches are on a par with (*clue*) or worse than (*avg_sum_ss*, *coreset*, *batch_bald*) the *random* baseline. To investigate the reason behind, inspired by [39], we compute the average inter class variations over AL iterations in Table IV. The inter class variation is defined as the $\sigma \times C$, where C is the number of class and σ is the standard deviation of number of instances for all classes. The lower this value is, less variations and more balance the object category distribution possesses. We can quantitatively observe that variations of YCBV are significantly larger than the others due to greater number of class, which might pose greater difficulty on decreasing the label distribution shift. Further from row-wise comparison, there is an obvious inversely proportional relation between inter class variations and the performance on YCBV, which is obscure on EDAN and SAM. Therefore, we infer that the impact of label distribution shift is more severe on data sets with greater number of class, thus impeding the effective utilization of uncertainty estimates. Considering this, we suggest that it is more effective to employ the proposed pipeline for bridging the reality gap when the class imbalance problem, one instance of label distribution shift is at a small scale.

TABLE IV
INTER CLASS VARIATIONS FOR THE SELECTED DATA SET IN EACH ITERATION DURING ACTIVE LEARNING. LOWER THE BETTER.

| | Random | Sub-sampling | Core-set | Clue | Batch-bald |
|------|------------|--------------|-----------|------------|-------------|
| EDAN | 90 | 152 | 78 | 114 | 126 |
| SAM | 81.3 | 39.9 | 71.8 | 75.3 | 12.6 |
| YCBV | 268 | 316 | 305 | 268 | 318 |

B. Deployment on EDAN

On account of the working scenarios (e.g. care-giving) for an assistive robot [11], a variety of objects need to be detected and the manual efforts required for adaptation must be kept as minimum as possible. Therefore, we show the effectiveness of the proposed idea in a shared-control grasping task on EDAN (Fig. 8), where a user such as people with motor disability sitting on the chair intends to control the robot arm for tasks like pouring by using an input device (EMG signal sensors or a spacemouse (used in the demo))with lower degrees of freedom (DoFs) than that of the end effector (3 vs. 6). The mis-correspondence of DoFs between the input device and the manipulator demands that the user needs to tediously switch input mapping between

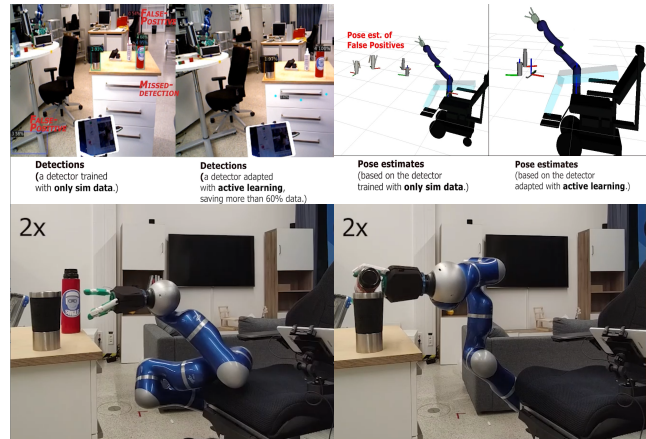


Fig. 8. Exemplary screenshots of a pouring task via shared control on an assistive robot. The two screenshots on the top show the performance of the detector and the corresponding pose estimates (visualized in Rviz) before (left in each column) and after (right in each column) adaptation via the proposed pipeline. The two screenshots at the bottom show the sequence of a grasping and pouring task execution with shared-control [11].

them for task completion in a pure manual control mode. In order to ease task execution, we employ shared-control templates [46], which require robust and precise 2D object detection and pose estimation⁶⁵. For more details on how to incorporate the perception pipeline into the shared-control module, we refer readers to the original work [46].

In this demo, we integrate the adapted detector trained with a similar setting introduced in the previous section and further use the Augmented autoencoder (AAE) and Iterative Closest Point (ICP) pipeline [3] for accurate pose estimation. This perception pipeline can be deployed on an embedding system such as a NVIDIA Jetson TX2 or a workstation PC (used in demo), the predictions (i.e. pose estimates of the detected objects) are then sent to the shared-control module via Links and Nodes (LN) middle-ware. Based on this, the user is able to control the manipulator to perform a series of common daily tasks such as pouring and drinking with much less cognitive workload. We also provide a video to showcase the deployment.

VI. CONCLUSION

This paper presents an active Sim-to-Real pipeline for 2D object detection, in which, a model is initially learned from synthetic data. Having observed the sub-optimal performance of learning only from simulation, we propose to efficiently use real annotated data via exploiting deep Bayesian active learning. Empirically, we demonstrate the encouraging impact of the proposed pipeline on classification and 2D object detection data sets, further address the limitation of the proposed pipeline and show its applicability on a real robotic system. In particular, our experiments indicate that the proposed sampling strategy can alleviate the label distribution shift which can have a vital impact on the success of our pipeline. More importantly, our work provides an empirical evidence that the real annotated images can efficiently reduce the reality gap.

VII. ACKNOWLEDGMENTS

We thank the anonymous reviewers and area chairs for their thoughtful feedback. Sincere thanks to the re-enabling robot (EDAN) team at DLR, especially the help on real robot deployment from Annette Hagenruber and Gabriel Quere. Jianxiang Feng is supported by the Munich School for Data Science (MUDS) and Rudolph Triebel is a member of MUDS.

REFERENCES

- [1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Int. Conf. on Computer Vision*, 2017.
- [2] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z.-C. Márton, and R. Triebel, “Unknown Object Segmentation from Stereo Images,” in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021.
- [3] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, “Augmented autoencoders: Implicit 3d orientation learning for 6d object detection,” *Int. Journal of Computer Vision*, vol. 128, no. 3, 2020.
- [4] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, “Using simulation and domain adaptation to improve efficiency of deep robotic grasping,” in *ICRA*, 2018.
- [5] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, “Synthesizing training data for object detection in indoor scenes,” *arXiv:1702.07836*, 2017.
- [6] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,” *IROS*, 2017.
- [7] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, “Blenderproc,” *arXiv:1911.01911*, 2019.
- [8] M. G. Müller, M. Durner, A. Gawel, W. Stürzl, R. Triebel, and R. Siegwart, “A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments,” in *IROS*, 2021.
- [9] A. K. Tanwani, “Domain invariant representation learning for sim-to-real transfer,” in *CoRL*, 2020.
- [10] M. Ranaweera and Q. H. Mahmoud, “Virtual to real-world transfer learning: A systematic review,” *Electronics*, vol. 10, no. 12, 2021.
- [11] J. Vogel, A. Hagenruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Höppner, D. Leidner, and A. Albuschäffer, “Edan: An emg-controlled daily assistant to help people with physical disabilities,” in *IROS*, 2020.
- [12] M. Durner, S. Kriegel, S. Riedel, M. Brucker, Z. Márton, F. Bálint-Benczédi, and R. Triebel, “Experience-based optimization of robotic perception,” in *Int. Conf. on Advanced Robotics (ICAR)*, 2017.
- [13] J. Feng, M. Durner, Z.-C. Marton, F. Balint-Benczedi, and R. Triebel, “Introspective robot perception using smoothed predictions from bayesian neural networks,” *Robotics Research. ISRR 2019*.
- [14] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Int. Conf. on Machine Learning (ICML)*, 2016.
- [15] A. Harakeh, M. Smart, and S. L. Waslander, “Bayesod: A bayesian approach for uncertainty estimation in deep object detectors,” in *Int. Conf. on Robotics and Automation (ICRA)*, 2020.
- [16] A. Prabhu, C. Dognin, and M. Singh, “Sampling bias in deep active classification: An empirical study,” *arXiv:1909.09389*, 2019.
- [17] E. Zhao, A. Liu, A. Anandkumar, and Y. Yue, “Active learning under label shift,” in *Int. Conf. on Artificial Intelligence and Statistics*, 2021.
- [18] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, “Active adversarial domain adaptation,” in *Winter Conf. on Applications of Computer Vision (WACV)*, 2020.
- [19] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, “Active domain adaptation via clustering uncertainty-weighted embeddings,” in *CVPR*, 2021.
- [20] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, “On pre-trained image features and synthetic images for deep learning,” in *Europ. Conf. on Computer Vision (ECCV) Workshops*, 2018.
- [21] T. Hodan, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter, “Photorealistic Image Synthesis for Object Instance Detection,” *arXiv*, 2019.
- [22] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, “Adapting object detectors via selective cross-domain alignment,” in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [23] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, “Domain adaptive faster r-cnn for object detection in the wild,” in *CVPR*, 2018.
- [24] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *Journal of Artificial Intelligence Research*, vol. 4, 1996.
- [25] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, “Deep active learning for efficient training of a lidar 3d object detector,” in *Intelligent Vehicles Symposium (IV)*, 2019.
- [26] A. Kirsch, J. Van Amersfoort, and Y. Gal, “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [27] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, “Bayesian uncertainty matching for unsupervised domain adaptation,” *arXiv:1906.09693*, 2019.
- [28] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. López, “Active learning for deep detection neural networks,” in *Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 3672–3680.
- [29] S. Roy, A. Unmesh, and V. P. Namboodiri, “Deep active learning for object detection,” in *British Machine Vision Conf. (BMCV)*, vol. 362, 2018.
- [30] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu, “Localization-aware active learning for object detection,” in *ACCV*, 2018.
- [31] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, “Active learning for deep object detection via probabilistic modeling,” *ICCV*, 2021.
- [32] J. Lee, M. Humt, J. Feng, and R. Triebel, “Estimating model uncertainty of neural networks in sparse information form,” in *ICML*, 2020.
- [33] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, “Dropout sampling for robust object detection in open-set conditions,” in *ICRA*, 2018.
- [34] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel, “Trust your robots! predictive uncertainty estimation of neural networks with sparse gaussian processes,” in *5th Annual Conference on Robot Learning (CoRL)*, 2021.
- [35] 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,” *arXiv:2107.03342*, 2021.
- [36] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, “Probabilistic object detection: Definition and evaluation,” in *WACV*, 2020.
- [37] D. J. MacKay, “Information-based objective functions for active data selection,” *Neural Computation*, vol. 4, no. 4, 1992.
- [38] Y. Yang, G. Ma, *et al.*, “Ensemble-based active learning for class imbalance problem,” *Journal of Biomedical Science and Engineering*, vol. 3, no. 10, 2010.
- [39] U. Aggarwal, A. Popescu, and C. Hudelot, “Active learning for imbalanced datasets,” in *WACV*, 2020.
- [40] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review,” *Trans. on Pattern Analysis and Machine Intelligence*, 2020.
- [41] Y. LeCun and C. Cortes, “MNIST handwritten digit database.” 2010.
- [42] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *Journal of Machine Learning Research*, vol. 17, no. 1, 2016.
- [43] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” *Robotics: Science and Systems (RSS)*, 2018.
- [44] J. Lee, R. Balachandran, Y. S. Sarkisov, M. De Stefano, A. Coelho, K. Shinde, M. J. Kim, R. Triebel, and K. Kondak, “Visual-inertial telepresence for aerial manipulation,” in *ICRA*, 2018.
- [45] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” *Int. Conf. on Learning Representations (ICLR)*, 2018.
- [46] G. Quere, A. Hagenruber, M. S. Z. Iskandar, S. Bustamante Gomez, D. Leidner, F. Stulp, and J. Vogel, “Shared control templates for assistive robotics,” in *ICRA*, 2020.

A.4. Publication 4

Matan Atad*, **Jianxiang Feng***, Ismael Rodríguez, Maximilian Durner, Rudolph Triebel (2023): “*Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning*”. In the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- <https://elib.dlr.de/195845/>

Please refer to the final published version for citation:

```
@inproceedings{atad2023,  
  author = {Atad, Matan and Feng, Jianxiang and Rodriguez Brena, Ismael  
    Valentin and Durner, Maximilian and Triebel, Rudolph},  
  publisher = {IEEE},  
  booktitle = {2023 IEEE/RSJ International Conference on Intelligent  
    Robots and Systems, IROS 2023},  
  year = {2023},  
  title = {Efficient and Feasible Robotic Assembly Sequence Planning via  
    Graph Representation Learning},  
  url = {https://elib.dlr.de/195845/},  
  keywords = {Graph Neural Networks, Robotic Assembly Sequence Planning}  
}
```

Efficient and Feasible Robotic Assembly Sequence Planning via Graph Representation Learning

Matan Atad^{*2}, Jianxiang Feng^{*1,2}, Ismael Rodríguez^{1,2}, Maximilian Durner^{1,2} and Rudolph Triebel^{1,2}

Abstract— Automatic Robotic Assembly Sequence Planning (RASP) can significantly improve productivity and resilience in modern manufacturing along with the growing need for greater product customization. One of the main challenges in realizing such automation resides in efficiently finding solutions from a growing number of potential sequences for increasingly complex assemblies. Besides, costly feasibility checks are always required for the robotic system. To address this, we propose a holistic graphical approach including a graph representation called Assembly Graph for product assemblies and a policy architecture, Graph Assembly Processing Network, dubbed GRACE for assembly sequence generation. With GRACE, we are able to extract meaningful information from the graph input and predict assembly sequences in a step-by-step manner. In experiments, we show that our approach can predict feasible assembly sequences across product variants of aluminum profiles based on data collected in simulation of a dual-armed robotic system. We further demonstrate that our method is capable of detecting infeasible assemblies, substantially alleviating the undesirable impacts from false predictions, and hence facilitating real-world deployment soon. Code and training data are available at <https://github.com/DLR-RM/GRACE>.

I. INTRODUCTION

Aiming for high flexibility, manufacturers around the globe are introducing automation for *Robotic Assembly Sequence Planning* (RASP) at a greater pace to respond to rapid changes in market needs for customization of novel product variants [1]. These changes cause often modifications in assembly lines, requiring time-consuming and resource-intensive re-planning, because of the NP-hard combinatorial characteristic [2] of *Assembly Sequence Planning* (ASP), where the number of possible solutions grows with the factorial of the amount of parts involved. Also, to check whether a certain assembly sequence can actually be executed on a specific robotic system is computationally expensive. For example in [3], 11 minutes were required for the assembly motion planning of an IKEA chair. This is more time than the actual execution of the plan, not to mention the case of product variants whose assembly sequence space itself must be explored, easily leading to a search of hours or days instead of minutes.

Several existing works attempt to improve the tedious ASP process by predicting feasible assembly sequences [4], [5] or inferring the underlying rules guiding their creation [6], [7]. Although those works already facilitate the assembly planning, they still lack desirable attributes such as generalization

* Equal Contribution.

¹ Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82110 Wessling, Germany. <first>. <second>@dlr.de

² Department of Informatics, Technical University of Munich, 85748 Garching, Germany. <first>. <second>@tum.de

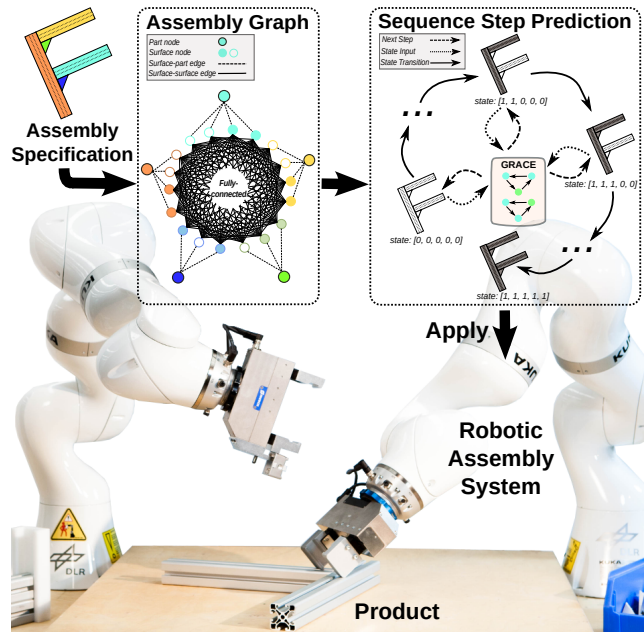


Fig. 1: **Workflow of our proposed graphical ASP method** on a dual-armed robotic system (simulated in our setting): an aluminum assembly specification is first represented as an Assembly Graph and then fed into our policy network GRACE, designed to flexibly and efficiently generate assembly sequences in a *step-by-step* manner that can be executed by the robot. Best viewed in color.

across varying product types and sizes as well as run-time efficiency. In this work, we address these problems with a graphical learning-based approach, that is able to automatically generate sequences for *unknown* assembly variants in an *efficient* way.

In a nutshell of our main idea, inspired by [8], we formulate RASP as a sequential decision-making problem with a *Markov Decision Process* (MDP), in order to break the restriction of combinatorial complexity wrt. the number of parts and thus, boost generalization performance. Hence the sequence is generated step-by-step based on the current assembly state. Meanwhile we exploit the idea of distilling previous knowledge acquired for assembling products to predict the next feasible actions with a designed policy architecture. This architecture is optimized to imitate the demonstration sequences collected in simulation which are interpreted as expert demonstrations.

Specifically, to put the aforementioned ideas into practice,

we propose to use a graphical representation to faithfully describe the spatial structure of assemblies. Our so-called Assembly Graph is adapted from and more fine-grained than the one in [7] by representing the assembly as a heterogeneous graph whose edges denote geometrical relations between the assembly part surfaces. Based on this, we further develop a policy architecture based on *Graph Neural Networks* (GNNs), called **GR**aph Assembly proCessing nEtworks, for short GRACE, to extract useful information from the Assembly Graph and predict actions determining which parts should be assembled next. Apart from this, false predicted sequences and infeasible assemblies pose a severe problem for efficiency of learning-based assembly robots, e.g., an incorrect sequence might require the robot to perform time-consuming re-planning. Therefore, it would be beneficial to detect these beforehand, e.g., being introspective against false predictions [9], hence we further develop and analyze various schemes to enhance the performance of feasibility prediction.

It is worthwhile to note that there are several advantages for the proposed graphical representation and the policy architecture, GRACE: (1) Invariance to number of parts: contrary to previous works such as [7] restricted by a fixed number of parts, ours is free from this limitation, as GRACE is capable of handling varying number of input graph nodes. (2) Memory efficient learning: GRACE employs shared weights across all nodes in the graph, further alleviating the burden of the aforementioned complexity. (3) Generalization: GRACE trained on assemblies of one size is able to generalize to those of smaller sizes (see results in Tab. III). (4) Multiple solutions: GRACE predicts several feasible sequences (in contrast to [8]), allowing greater flexibility and resilience during execution.

We validate the proposed method with comprehensive experiments based on a dataset of assemblies made of different numbers of aluminum parts created in simulation of a dual-armed robotic system. This setting can be mapped to various tasks in the industry [7] as it allows for construction of numerous product variations. Moreover, as shown in Fig. 3, it requires a deeper understanding of several complex relations (e.g., distances between parts, physical part characteristics). The results show that our approach is able to efficiently predict feasible assembly sequences across product variants (with few millisecond to predict the next step (V-A.3)).

To summarize, our contribution is three-fold:

- We introduce Assembly Graph, a heterogeneous graphical representation for the RASP task, which is a more fine-grained and flexible representation than our previous one in [7] by including part surfaces and parts in the same graph.
- We develop a policy architecture GRACE to process the Assembly Graph and predict feasible assembly sequences in a step-by-step manner as well as the feasibility for a given assembly specification.
- We conduct comprehensive experiments in simulation to validate the proposed approach including failure analysis and ablation studies on design choices.

II. RELATED WORK

a) Assembly Sequence Planning: A popular assembly graph representation for ASP is the AND/OR Graph [11], a formalism to encode the space of feasible assembly sequences, which can be created with the Disassembly For Assembly strategy [12]–[15]. However, these approaches are restricted on time to find a solution efficiently due to the feasibility checks. While graph search methods are impractical for larger assemblies because of the combinatorial explosion problem, heuristic intelligent search methods provide another alternative. They reject infeasible sequences and search for feasible ones close to the optimal based on manually designed termination criteria [16], [17], learned [18], [19] or hand-crafted [20] energy functions. More recently, Zhao *et al.* [4] and Watanabe *et al.* [5] applied deep *Reinforcement Learning* (RL) for ASP. Different to us, they do not have a graph representation to take into account relations between parts. Targeting at RASP, Rodriguez *et al.* [6], [7] suggested inferring assembly rules (e.g., a specific part should be assembled before another), which can be transferred from previous identified sub-assemblies to those of larger sizes to prune the search space, thus reducing planning time. Their approach only produces rules, from which the final assembly sequences need to be derived additionally. It also requires further re-training when adapting to other product variants. Enlightened by them, we refine their graph representation to a more fine-grained level and adapt their idea with a learning-based approach, aiming to mitigate these issues. Similar to us, Ma *et al.* [10] used GNN for ASP of LEGO structures. However, they differ from us in two aspects, first they do not consider assembly robots in the loop and second they model assemblies only with a coarser graph representation whose edges only consider connections among parts instead of part surfaces. To clearly show different characteristics among relevant works, we provide a concise comparison in Tab. I.

b) Graph Representation Learning in Task Planning: In this setting, graphs commonly incorporate nodes for manipulated objects [21]–[23], their target positions [8], [24] and the robot gripper [25]. Edges can represent high-level relations between objects [21], [23]. With the graph representation, Zhu *et al.* [23] and Ye *et al.* [25] generated feasible candidate paths by sampling, and trained a network that predicts a sequence of feasible actions in backward and forward search, respectively. Nguyen *et al.* [21] performed sampling to find action sequences that transform the source to target graph and then used optimization to eliminate invalid sequences subject to the environment constraints. Besides, some researchers resorted to RL methods such as [22], [24], and [26], who used GNNs for task planning. Recently, Lin *et al.* [8] utilized *Imitation Learning* (IL) to train two GNNs, one for selecting objects in the scene and another picking a suitable goal state from a set of possible goal positions for long-horizon manipulation tasks. Inspired by them, we train our GNNs for RASP task by leveraging IL for ease and efficiency in training.

TABLE I: Comparison between our proposed method and other relevant works.

| | Efficient generalization across assembly sizes? | Robotic constraints involved? | Direct sequences generation? | Fine-grained graph representation? |
|---------------------|--|----------------------------------|---------------------------------|---------------------------------------|
| ASPW-DRL [4] | ✗ | ✗ | ✓ | ✗ |
| LEGO-GRAPH [10] | ✓ | ✗ | ✓ | ✗ |
| KT-RASP [7] | ✗ | ✓ | ✗ | ✗ |
| Our proposed method | ✓ | ✓ | ✓ | ✓ |

III. BACKGROUND

In this section, we briefly recap the concept of GNNs and heterogeneous graphs, which are the base for our method.

a) *Graph Neural Networks*: A GNN operates on an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes \mathcal{V} and edges \mathcal{E} , where every node $v \in \mathcal{V}$ is assigned with a feature vector $\phi(v)$. It updates node features by exchanging information between neighboring nodes. This is done with multiple Message Passing layers [27]. For each layer l , let $\mathbf{h}_i^0 = \phi(v_i)$ be the input features of node v_i and \mathcal{N}_i its set of neighboring nodes. Then we can define a three-step process to update these features:

- 1) *Gather* feature from neighboring nodes: $\{\mathbf{h}_j^{l-1}\}_{j \in \mathcal{N}_i}$.
- 2) *Aggregate* messages from the neighboring nodes: $\mathbf{m}_i^l = g_\omega(\{\mathbf{h}_j^{l-1}\}_{j \in \mathcal{N}_i})$.
- 3) *Update* features of node v_i : $\mathbf{h}_i^l = f_\phi(\mathbf{h}_i^{l-1}, \mathbf{m}_i^l)$.

The function g_ω can be either constant (e.g. sum) or learned during training. The term f_ϕ is a *Neural Network* (NN) parameterized by ϕ . Both, f_ϕ and g_ω , are shared across all nodes in the graph, making GNNs efficient and independent of the number of nodes in the graph.

In our proposed method we apply a Graph Attention Network (GAT) [28], a popular variant of GNNs, that defines g_ω as attention:

$$\mathbf{m}_i^l = \sum_{j \in \mathcal{N}_i} (\alpha_{i,j} \cdot \mathbf{h}_j^{l-1}), \quad (1)$$

$$\mathbf{h}_i^l = \mathbf{W}_1 \cdot \alpha_{i,j} \mathbf{h}_i^{l-1} + \mathbf{W}_1 \cdot \mathbf{m}_i^l, \quad (2)$$

$$\alpha_{i,j} = \frac{\exp(\mathbf{a} \cdot \sigma(\mathbf{W}_2[\mathbf{h}_i^{l-1} \parallel \mathbf{h}_j^{l-1} \parallel e_{i,j}]))}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp(\mathbf{a} \cdot \sigma(\mathbf{W}_2[\mathbf{h}_i^{l-1} \parallel \mathbf{h}_k^{l-1} \parallel e_{i,k}]))}, \quad (3)$$

where \mathbf{W}_1 , \mathbf{W}_2 , and \mathbf{a} are learned, σ is a Leaky ReLU activation function, and $[a \parallel b]$ is a concatenation operator between a and b .

b) *Heterogeneous Graph*: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generalizes graphs to multiple types of nodes and edges [?]. Each node $v \in \mathcal{V}$ belongs to one particular node type $\psi_n(v)$ and analogously each edge $e \in \mathcal{E}$ to an edge type $\psi_e(e)$. In [29], the authors extend *Graph Attention Network* (GAT)s to a heterogeneous graph setting. This is accomplished by obtaining for each node a different updated feature vector per group of specific neighboring source node and edge types, and aggregating the features to obtain a single result, for instance using a sum. This formulation is essential, as every type of neighboring node may have a different feature dimension.

IV. METHOD

In this section, RASP is formulated as a sequential decision-making problem with a MDP and then we present our graph representation to depict assemblies. Based on this, we elaborate the proposed network GRACE, and demonstrate the assembly sequence generation.

A. Problem Formulation

We describe the sequence prediction task for an assembly with N parts as a MDP [30] with a discrete state space \mathcal{S} and a high-level discrete action space \mathcal{A} .

Starting from state \mathbf{s}_t at time step t , executing action a_t produces a reward r_t and switches to state $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} | \mathbf{s}_t, a_t)$ with a transition function p . State $\mathbf{s}_t \in \{0, 1\}^N$ is a binary vector indicating which parts are already placed in their target position by 1 (i.e. assembled) otherwise by 0. Action $a_t \in \{1, \dots, N\}$ represents the next part placement among the unplaced ones. For *feasible* assemblies, there are multiple different sequences leading to the final state, in which all N parts are placed correctly. For *infeasible* assemblies, no sequence exists, due to constraints of different aspects spanning from part geometries to kinematic and dynamics regarding the robotic system. Our objective is to learn a policy network $\pi_\theta(\mathbf{s}_t) = a_t$ parameterized by θ , which is optimized to imitate the assembly demonstrations $\tau_i = \{\mathbf{s}_{i,1}, a_{i,1}^{exp}, \dots, \mathbf{s}_{i,T}, a_{i,T}^{exp}\}$ in a dataset of M sequences $\mathcal{D} = \{\tau_i\}_{i=1}^M$ and generalize across variants of different types and sizes at test time. In practice, our network predicts a set of multiple possible actions e.g. $K_t = \{a_{t,k}\}_{k=1}^{K_t}$ based on a tunable threshold to control the prediction quality.

B. Assembly Graphs

We represent the overall structure of an assembly with a heterogeneous graph. To make this representation agnostic to the rotation and mirroring of the assembly structure, we employ only relative distances instead of absolute positions for the features of edges between surfaces. More formally, given an assembly A (Fig. 2) at state \mathbf{s}_t it is modeled as a graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ containing two types of nodes: part nodes \mathcal{V}^p and surface nodes \mathcal{V}^s , and two types of edges: \mathcal{E}^{s-to-s} , connecting all surface nodes, and \mathcal{E}^{s-to-p} , connecting each surface node to its respective part. We detail each component as follows:

1) *Part Nodes*: Responsible for encoding the current state of the assembly. A part node $v_i^p \in \mathcal{V}^p$ is associated with a feature vector $\phi(v_i^p) = [assembled-flag \in \{0, 1\}, part-type \in$

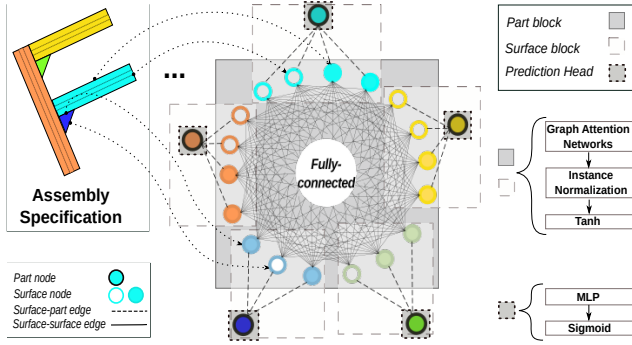


Fig. 2: **Illustration of Assembly Graph and GRACE.** Assembly Graph consists of edges connecting parts and their surfaces and edges among all part surfaces. In GRACE, the Part Block is shared for sub-graphs of part surfaces and the attached part, while Surface Block is for the sub-graph of all part surfaces. To predict scores for parts to be assembled next, we apply a prediction head on each spare part.

\mathbb{N} , $part-id \in \mathbb{R}^d$]. There are three atomic part types: *long profile*, *short profile* and *angle bracket*.

2) *Surface Nodes*: Different to the one in [7], we associate each surface node $v_i^s \in \mathcal{V}^s$ with the features $\phi(v_i^s) = [surface-type \in \mathbb{N}, surface-id \in \mathbb{R}^d]$. There are two surface types (*long* and *short*) for profiles and one (*lateral*) for brackets. Both the *part-id* and *surface-id* fields are encoded with a d -dimensional Sinusoidal Positional Encoding [31].

3) *Surface-to-Surface Edges*: We design a fully-connected graph for all surface nodes \mathcal{V}^s to capture the relation between untouched surfaces, which is more fine-grained than those in [7] with only connects between touched surfaces. These edges are assigned with a feature $\phi(e_i) \in \mathbb{R}$, indicating the *relation* between the two surfaces: $\phi(e_i) = relative\ distance$ (parallel); 1 (belong to the same part); -1 (orthogonal); 0 (same-surface loop).

4) *Surface-to-Part Edges*: These connect each surface and part node pair $(v_i^s, v_j^p) \in \mathcal{V}^s \times \mathcal{V}^p$, where surface v_i^s belongs to the part v_j^p . This type of edges is not associated with any feature vector.

C. Graph Assembly Processing Networks (GRACE)

Based on the formulation of a *step-by-step* sequential decision-making process per each part in the assembly in IV-A, we introduce **GR**aph **A**ssembly **pr**o**C**essing **n**et**w**orks, for short GRACE, $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$, where $a_i = \{y_i | y_i \geq \lambda\}_{i=1}^N$, to extract useful information from the Assembly Graph and predict the next action given the current state of an assembly of N parts. $\lambda \geq 0$ is a threshold used to control the quality of predicted sequences. GRACE outputs a score per part $y_i \in [0, 1], i \in \{1, \dots, N\}$, reflecting the probability of placing the i -th part next. We further articulate the main components of this network (Fig. 2), describe the algorithm for predicting the entire sequence of length N by traversing predicted steps and the way we infer the feasibility of a given assembly.

1) *Surface and Part Blocks*: The architecture is made of identical blocks, which are applied sequentially to obtain updated node features. Each block is made of a GAT [28], an Instance Normalization layer [32] and a Tanh function. We choose GAT as it allows to utilize the rich semantics of edge features for updating node features in our graph representation. Surface Blocks are applied on surface nodes \mathcal{V}^s and surface-to-surface edges \mathcal{E}^{s-to-s} for updating surface node features $\phi(v_i^s)$, while Part Blocks are applied on surface nodes \mathcal{V}^s , part nodes \mathcal{V}^p and surface-to-part edges \mathcal{E}^{s-to-p} to update part node features $\phi(v_j^p)$.

2) *Prediction Head and Loss Function*: To obtain a score per part, a fully-connected layer followed by a Sigmoid function is applied on each part node. During training, we minimize the loss between the network outputs and the ground-truth sequence steps from a dataset of assembly sequences (see IV-A) using binary cross-entropy. To note that, we apply this loss function for each part node separately. Our objective function (4) includes an additional regularization term (5), aiming at encouraging the network not to predict already placed parts:

$$L_\theta = \sum_{i=1}^M \sum_{j=1}^{N_i} (\hat{y}_{ij} \cdot \log(y_{ij}) + (1 - \hat{y}_{ij}) \log(1 - y_{ij})) + \delta L_{reg}, \quad (4)$$

$$L_{reg} = \sum_{i=1}^M \sum_{j=1}^{N_i} f_{ij} \cdot y_{ij}, \quad (5)$$

where M is the number of data examples in the dataset, N_i is the number of nodes in the i -th graph. Abusing the notations, we denote y_{ij} and \hat{y}_{ij} the output score of the model π_θ and the ground-truth step in a sequence for the j -th node in the i -th graph respectively. δ is a weighing coefficient and f_{ij} the value of the *assembled-flag* in the input features.

3) *Predicting Sequences*: As described, GRACE predicts a set of possible next steps based on the current state of an assembly. In order to generate a complete sequence (i.e. of length N), we repeatedly apply GRACE based on the current predicted state of the Assembly Graph. We devise an algorithm (Algo. 1) to traverse the assembly state tree using *Depth First Search* (DFS):

Starting with the graph in its initial state \mathcal{G}_0 – for all part nodes, *assembled-flags* are set to zero, the algorithm performs the following steps recursively: First, it checks the exit condition of the recursion – if all parts are already in place. Next, it predicts the probability for each part node y_i and picks those larger than the threshold λ , controlling the trade-off between precision and recall. Each of those nodes spawns a new branch individually. Therefore, we set the *assembled-flag* and call the recursion on the altered graph to retrieve possible sequences starting with the chosen node. Finally, we add the chosen nodes to the head of each returned sequence and return.

4) *Feasibility Prediction*: To address the issue from infeasible assemblies, we develop two schemes to infer the feasibility (defined in IV-A) of a given assembly: (1) We use the number of predicted complete sequences (output

Algorithm 1 Assembly State Tree Traversal

```
function TRAVERSE-TREE(Model  $M$ , Assembly Graph  $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ , Threshold  $\lambda$ )  
   $S \leftarrow \text{list}()$   
  if ( $\forall v \in \mathcal{V} : v.\text{assembled-flag} == 1$ ) then  
    return  $S$   $\triangleright$  Exit: all parts assembled  
  end if  
   $\mathbf{y} \leftarrow M(\mathcal{G}_t)$   
  for  $i \leftarrow 1$  to  $|\mathcal{V}|$  do  
    if  $\mathbf{y}[i] < \lambda$  then  
      continue  
    end if  
     $\mathcal{G}_{t+1} \leftarrow \text{copy}(\mathcal{G}_t)$   
     $[\mathcal{V}_{t+1}]_i.\text{assembled-flag} \leftarrow 1$   $\triangleright$  assembled node  $i$   
     $S_* \leftarrow \text{TRAVERSE-TREE}(M, \mathcal{G}_{t+1}, \lambda)$   
    for  $s$  in  $S_*$  do  
       $s_* \leftarrow [i] + s$   $\triangleright$  Add current part to the sequence  
       $S.\text{append}(s_*)$   
    end for  
  end for  
  return  $S$   
end function
```

by Algo. 1) as an indicator for the feasibility of a given assembly. If no sequences were retrieved, the assembly is predicted as infeasible. (2) We aggregate the features of all part nodes from a pre-trained GRACE with a *mean-pooling* operation, creating a feature vector for the entire assembly graph. This feature vector is then used to train a binary classifier for feasibility prediction, where we analyze several classifiers i.e. Support Vector Machines (SVMs), Multi-layer Perceptrons (MLPs) and Nearest Neighbor.

V. EXPERIMENT

In this section, we first describe the experimental setup such as our dataset, evaluation metrics and implementation details. To note that we use the term *size* to describe the number of parts of an assembly without prior notice. We evaluate the Sequence Prediction under two experimental protocols with 4-fold cross-validation: (1) **intra-sized**: the assemblies in training and test set share *the same* sizes; (2) **inter-sized**: the assemblies in training and test set have *different* sizes, where there are two sub-protocols: Many-to-one and One-to-Many (detailed in V-B.2) The results on Feasibility Prediction are presented before the failure analysis and ablation study.

A. Experimental Setup

1) *Dataset*: We applied our in-house simulation software MediView to randomly generate data of synthetic aluminium assemblies whose sizes range from 3 to 7 (denoted by A_i , where i is the size). The simulation software was tasked with putting together the structures by brute-forcing all part orders, while considering the restrictions of part geometries or those imposed by the capabilities of a dual-armed robotic system *KUKA LBR Med* (Fig. 1). More restrictions could

be added to this environment in a future work, e.g. taking into account grasp planning. An illustration of this process is given in Fig. 3. The resulting data consists of the following amount per size: $A_3 : 5717$, $A_4 : 2464$, $A_5 : 6036$, $A_6 : 2865$, $A_7 : 431$.

We post-processed the simulation output to obtain the *Placement Action* (required during training) – the next possible placement actions given a state of an assembly in a feasible sequence. In addition, we derive the *Feasibility* of each assembly based on the number of ground truth sequences e.g. 0 indicates an infeasible assembly.

2) *Metrics*: We use the following metrics for the sequence prediction task: (1) **Step-by-Step AUC** examines our method’s predictive performance to infer the parts that should be assembled next given the current state by comparing the ground truth binary labels with the predicted step scores. For this purpose we use the common Precision-Recall curve w.r.t. λ and finally deriving an *Area Under Curve* (AUC) score. (2) **Complete-Sequence AUC** evaluates the ability to infer the entire set of ground truth sequences, since a step-by-step evaluation only partially displays our application¹. We use Information Retrieval (IR) Precision-Recall [33], devised for set prediction evaluation, computed as $\text{IR-Precision} = |\text{RET} \cap \text{REL}|/|\text{RET}|$, $\text{IR-Recall} = |\text{RET} \cap \text{REL}|/|\text{REL}|$, where *RET* are the retrieved sequences and *REL* are the relevant sequences (i.e. ones in the ground truth set). Here, again, we plot an IR Precision-Recall curve and derive an AUC score. (3) **Precision@k (P@k)**: since in practice we only consider the highest scored predicted sequences, we also compute IR-Precision while taking into account only the top- k ones [34].

For feasibility prediction, we use common binary classification metrics *False Positive Rate FPR* and *True Positive Rate TPR*. In this setting, a positive instance is a feasible assembly and a negative an infeasible one.

3) *Implementation Details*: We use PyTorch Geometric (PyG) [35] to build the model which consists of 3 surface blocks and 1 part block with a latent-dimensionality of 94. For training, we choose a batch size of 256 and a learning rate of 0.0022 with Adam optimizer based on validation performance on 15% of the training samples during hyperparameter search. Besides, we set the regularization weight in Eq. 4 to 0.3 and length of positional encoding for node features to 16. For the training and evaluation of the feasibility classifiers a balanced dataset is used. Our model includes 51.7K trainable parameters and requires 4.06 ± 0.15 ms to infer the next feasible sequence step². More details are referred to our open-sourced code.

B. Results

1) *Sequence Prediction for Intra-sized Assemblies*: The results are shown separately per assembly size (Tab. II), including the step-by-step and complete-sequence AUC, P@k

¹Consider a method that predicts the first 97 steps correctly and fails in the 98-th step for a 100-parts-assembly.

²Measured on NVIDIA GeForce GTX 1080.

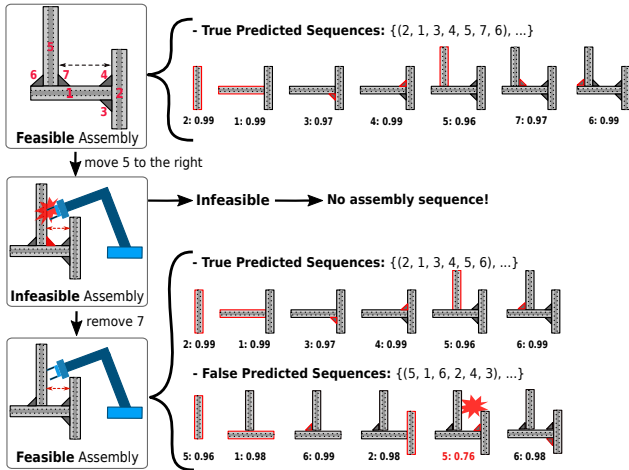


Fig. 3: **Examples of ASP for Aluminum Assemblies.** We demonstrate the complexity of our task through the predictions for three different assemblies: starting from the top, there is a feasible assembly sequence predicted based on the assembly on the left. By decreasing the distance between part 5 and 4, it becomes infeasible due to limited space for the robot arm. Further, by removing part 7, with certain sequences such as the one shown in the figure, it is feasible. But this does not work with another, i.e. the false predicted sequence, because this one would cause collision.

scores for $k \in \{1, 2, 3\}$ with a threshold of 0.5. GRACE is able to perform perfectly on step prediction for all sizes. More relevant to our goal and more challenging than step prediction, our method can reach 1.0 for small sizes (e.g. 3 and 4 parts) on the task of complete sequence prediction. However, we can observe a slight drop for larger sizes (e.g. 5, 6 and 7 parts), implying the greater complexity for large assemblies. Hence, GRACE can effectively learn an useful inductive bias from our proposed graph representation when trained with similar sizes. To note that, this performance has already reached that of the approach in [7] and GRACE is able to generalize to larger sizes which the previous one is incapable of (see Tab. I for a qualitative comparison).

2) Sequence Prediction for Inter-sized Assemblies:

To comprehensively evaluate the *generalization* ability of GRACE across different sizes, a distinct limitation of previous works [7], [36], we further design two more *challenging* sub-protocols under the inter-sized protocol.

- **Many-to-one:** GRACE is trained on assemblies of mixed sizes but i , i.e. $A_{\forall j \neq i}$, and tested on A_i .
- **One-to-many:** GRACE is trained on a single-sized dataset A_i and tested on all the other, i.e. $A_{\forall j \neq i}$.

1. *Many-to-one:* This setting is similar to the intra-sized one except that we excluded assemblies of the size evaluated at test time from the training set. When comparing the results in this setting to the intra-sized ones (Tab. II), we observe a slight performance decrease in AUC on step and sequence prediction. However, note that $P@1$ and $P@2$ can still reach ~ 1.0 for small sizes (3 and 4 parts) and ~ 0.9 for large

sizes, indicating that GRACE is capable on generalizing to assembly variants with different sizes that have not been seen before.

2. *One-to-many:* This setting is an inverse version of the previous one, which is more challenging, since the amount and diversity of the training set are much lower than before³. The results (lower triangular block in Tab. III) provide a clear pattern that GRACE is able to obtain comparably better results for assemblies with less parts. For instance, trained with only A_5 , GRACE preforms well for A_3 and A_4 which is reasonable as the constraints guiding smaller assembly structures are *contained* in larger ones. This shows the *generalization* capability and *sample efficient* learning ability (trained on single size and worked on smaller sizes) of our method. Nevertheless, the performance drops for larger assemblies (see the upper triangular block in Tab. III). We hypothesize that an increasing amount of items introduces new constraints that are not covered by the training data. Thus, there might be a critical number of items containing all possible constraints that if included in the training data can lead to an overall generalizing model.

3) *Feasibility Prediction:* In this setting, we examine the ability of our approach to detect infeasible assemblies. For this experiment we consider GRACE trained on multiple sizes and test on the A_5 set. As mentioned in IV-C, we compare the implicit approach via the number of predicted sequences (Algo. 1) and alternative schemes exploiting the graph representation of the pre-trained GRACE. Therefore, we explore several binary classifiers i.e. SVMs, a Multi-layer Perceptron (MLP) and Nearest Neighbor. As seen in Fig. 4, GRACE (*#sequences*) is able to detect infeasible assemblies (AUC of 0.97). However, training our method exclusively with feasible assemblies (*#sequences, feasible only*) results in a poor detection performance. We hypothesize that by missing infeasible structures during training the method learns to always assemble an item leading to overconfidence. Exploiting an additional scheme by adding one of the classifiers (except SVM with RBF kernel) maintains or even slightly improves the performance.

C. Failure Analysis

To better understand the limitations of our method, we conduct an analysis of falsely predicted assembly sequences by our baseline model for A_5 and A_6 . Each of these false predicted sequences includes a *false step*, i.e. action from which the sequence deviates from the corresponding ground truth sequences. Fig. 5 depicts the histogram of false steps binned by their predicted probability. One can observe a large amount false steps performed in the beginning of the sequence (steps 1 and 2) with a high confidence. On the other hand, wrong step predictions at the end of an assembly (steps 4 and 5) exhibit lower confidence scores. We hypothesize that this bias is a result of an inherit imbalance in our training setting. Our dataset samples could be thought of as nodes in

³We do not perform this experiment on A_7 , as there are relatively small amount of assemblies with 7 parts in the dataset.

TABLE II: Sequence Prediction Results for intra-sized and inter-sized (many-to-one) assemblies.

| Metrics | Intra-sized | | | | | Inter-sized | | | | |
|--------------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | A_3 | A_4 | A_5 | A_6 | A_7 | A_3 | A_4 | A_5 | A_6 | A_7 |
| Step-by-Step AUC (\uparrow) | 1.00 ± 0.00 | 1.00 ± 0.00 | 1.00 ± 0.00 | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.99 ± 0.02 | 1.00 ± 0.00 | 0.98 ± 0.10 | 0.98 ± 0.00 | 1.00 ± 0.00 |
| Complete Sequence AUC (\uparrow) | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.96 ± 0.02 | 0.93 ± 0.03 | 0.97 ± 0.02 | 0.97 ± 0.03 | 1.00 ± 0.10 | 0.87 ± 0.03 | 0.90 ± 0.04 | 0.95 ± 0.07 |
| $P@1$ (\uparrow) | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.95 ± 0.04 | 0.96 ± 0.06 | 0.99 ± 0.01 | 0.99 ± 0.01 | 0.98 ± 0.03 | 0.90 ± 0.09 | 0.88 ± 0.07 | 0.96 ± 0.05 |
| $P@2$ (\uparrow) | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.94 ± 0.04 | 0.95 ± 0.07 | 0.99 ± 0.01 | 0.99 ± 0.01 | 0.97 ± 0.03 | 0.87 ± 0.12 | 0.87 ± 0.07 | 0.96 ± 0.04 |
| $P@3$ (\uparrow) | - | 1.00 ± 0.00 | 0.99 ± 0.01 | 0.95 ± 0.08 | 0.99 ± 0.02 | - | 0.95 ± 0.06 | 0.93 ± 0.10 | 0.85 ± 0.08 | 0.96 ± 0.04 |

TABLE III: Sequence Prediction Results for inter-sized assemblies in one-to-many setting.

| Training Set | Step-by-Step AUC (\uparrow) on assemblies of various sizes | | | | | Complete Sequence AUC (\uparrow) on assemblies of various sizes | | | | |
|--------------|--|-----------------|-----------------|-----------------|-----------------|---|-----------------|-----------------|-----------------|-----------------|
| | A_3 | A_4 | A_5 | A_6 | A_7 | A_3 | A_4 | A_5 | A_6 | A_7 |
| A_4 | 0.92 ± 0.11 | - | 0.48 ± 0.12 | 0.41 ± 0.11 | 0.43 ± 0.12 | 0.93 ± 0.09 | - | 0.28 ± 0.12 | 0.25 ± 0.15 | 0.25 ± 0.15 |
| A_5 | 0.93 ± 0.06 | 0.89 ± 0.07 | - | 0.78 ± 0.14 | 0.59 ± 0.16 | 0.83 ± 0.07 | 0.70 ± 0.09 | - | 0.36 ± 0.12 | 0.24 ± 0.07 |
| A_6 | 0.90 ± 0.10 | 0.89 ± 0.11 | 0.93 ± 0.04 | - | 0.59 ± 0.16 | 0.73 ± 0.16 | 0.68 ± 0.24 | 0.71 ± 0.13 | - | 0.24 ± 0.07 |

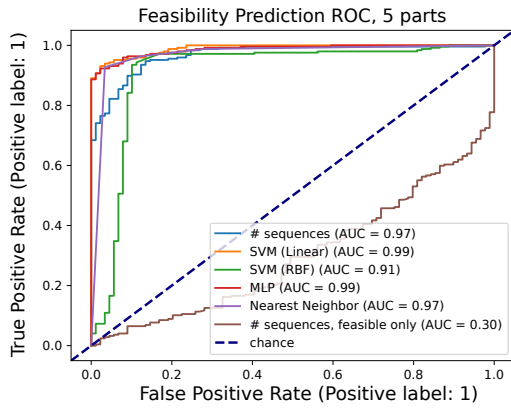


Fig. 4: Results of Feasibility Prediction. Comparison of different proposed schemes for feasibility classification and an ablation study in which infeasible ones are unavailable based on assemblies with 5 parts.

a state tree, where earlier steps share state nodes closer to the root and later ones have independent nodes towards the leaves. As each of these nodes is represented only once, there are fewer samples in the dataset attributed to earlier steps. This problem could be solved by balancing the training set based on the sequence step.

D. Ablation Study

In Assembly Graph (IV-B), both the part and surface node embeddings contain a $16d$ sinusoidal positional encoding [31]. We conducted an ablation study to investigate the impacts from the values and permutation order thereof based on A_5 . (1) **Values**: Initializing the positional encoding with random values dramatically decrease the performance of our method (Tab. IV). We hypothesize that these positions introduce *geometrical bias*, which is helpful in our task. (2) **Permutation Order**: We number assembly parts and surfaces in a constant order. Parts are counted beginning from the one closest to the environment origin. Surfaces, on the other hand, are always numbered clockwise, starting from the respective part top. Permuting both part and surface

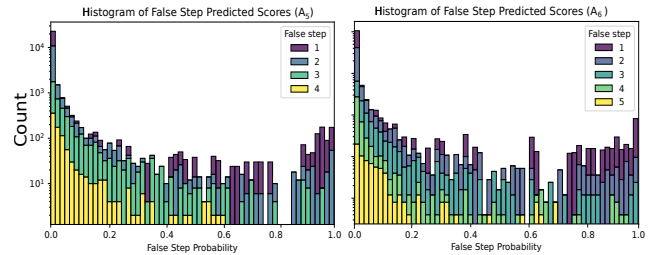


Fig. 5: Failure Analysis: false predicted sequences. Histogram of predicted probability (for A_5 and A_6) in false steps reveals a drift in which GRACE is overconfident in mistakes performed early. This is an evidence for an inherit bias in our training setting.

TABLE IV: Ablation study into the contribution of positional encodings to our method.

| Positional Encoding | A_5 AUC (\uparrow) |
|---|--------------------------|
| Baseline, sinusoidal encoding [31] | 0.94 |
| Random values | 0.37 |
| No encodings | 0.07 |
| Part permutations (test time) | 0.60 |
| Surface permutations (test time) | 0.27 |
| Part permutations (training and test time) | 0.97 |
| Surface permutations (training and test time) | 0.10 |

orders *only* at test time causes severe performance degradation, indicating constant numbering during training harms the model’s ability to generalize (Tab. IV). Interestingly, allowing permutations for only part order can boost the performance while this is not the case for surface permutations. This demonstrates the importance of these features for the network to extract information from the parts’ geometrical structure.

VI. CONCLUSION

In this work, we addressed the RASP problem with a learning-based framework. Concretely, we propose a graph representation, called Assembly Graphs for the aluminum profile assemblies, which is flexible to represent different

2d structures and meanwhile agnostic to rotation and mirroring. Based on this, a novel policy network – GRACE is introduced to extract meaningful information for assembly sequence prediction. Extensive experiments in simulation verify the capability of transferring knowledge between different assembly tasks, on which previous methods fall short. Further, our method can generalize knowledge gained on larger assemblies and then apply it to smaller ones. Last but not least, it is worth to mention, though only validated in simulation, our method should address the challenges during the real-world deployment like not finding a valid motion or a feasible grasping point if these cases are enclosed in the training data and learned to reject by GRACE. Meanwhile encouraged by the superior results on objects with simple geometries, our holistic graphical method lays a solid basis for handling complex 3d objects like curve blocks in the future.

ACKNOWLEDGMENTS

The paper received partial funding by the DLR internal project Factory of the Future Extended (FoF-X). Jianxiang Feng is supported by the Munich School for Data Science (MUDS). Rudolph Triebel is a member of MUDS.

REFERENCES

- [1] W. C. Shih, “Global supply chains in a post-pandemic world,” *Harvard Business review*, 98(5), 82-89, 2020.
- [2] M. F. F. Rashid, W. Hutabarat, and A. Tiwari., “A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches.” *International Journal of Advanced Manufacturing Technology*, vol. 59, pp. 335–349, 2011.
- [3] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham, “Can robots assemble an ikea chair?” *Science Robotics*, vol. 3, no. 17, p. eaat6385, 2018.
- [4] M. Zhao, X. Guo, X. Zhang, Y. Fang, and Y. Ou, “Aspw-drl: assembly sequence planning for workpieces via a deep reinforcement learning approach,” *Assembly Automation*, 2019.
- [5] K. Watanabe and S. Inada, “Search algorithm of the assembly sequence of products by using past learning results,” *International Journal of Production Economics*, vol. 226, p. 107615, 2020.
- [6] I. Rodríguez, K. Nottensteiner, D. Leidner, M. Kaßecker, F. Stulp, and A. Albu-Schäffer, “Iteratively refined feasibility checks in robotic assembly sequence planning,” *IEEE Robotics and Automation Letters (RAL)*, vol. 4, no. 2, pp. 1416–1423, 2019.
- [7] I. Rodríguez, K. Nottensteiner, D. Leidner, M. Durner, F. Stulp, and A. Albu-Schäffer, “Pattern recognition for knowledge transfer in robotic assembly sequence planning,” *IEEE RAL*, vol. 5, no. 2, pp. 3666–3673, 2020.
- [8] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, “Efficient and interpretable robot manipulation with graph neural networks,” *IEEE RAL*, vol. 7, no. 2, pp. 2740–2747, 2022.
- [9] J. Feng, M. Durner, Z.-C. Márton, F. Bálint-Benczédi, and R. Triebel, “Introspective robot perception using smoothed predictions from bayesian neural networks,” in *Robotics Research: The 19th International Symposium ISRR*. Springer, 2022, pp. 660–675.
- [10] L. Ma, J. Gong, H. Xu, H. Chen, H. Zhao, W. Huang, and G. Zhou, “Planning assembly sequence with graph transformer,” *arXiv preprint arXiv:2210.05236*, 2022.
- [11] L. Homem de Mello and A. Sanderson, “And/or graph representation of assembly plans,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–199, 1990.
- [12] T. De Fazio and D. Whitney, “Simplified generation of all mechanical assembly sequences,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 640–658, 1987.
- [13] U. Thomas, M. Barrenscheen, and F. Wahl, “Efficient assembly sequence planning using stereographical projections of c-space obstacles,” in *Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2003.*, 2003, pp. 96–102.
- [14] K. Nottensteiner, T. Bodenmueller, M. Kassecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas, “A complete automated chain for flexible assembly using recognition, planning and sensor-based execution,” in *Proceedings of ISR 2016: 47st International Symposium on Robotics*, 2016, pp. 1–8.
- [15] U. Thomas, T. Stouraitis, and M. A. Roa, “Flexible assembly through integrated assembly sequence planning and grasp planning,” in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 586–592.
- [16] B. Li, Y. Wu, H. Sun, Z. Cheng, and J. Liu, “Unity 3d-based simulation data driven robotic assembly sequence planning using genetic algorithm,” in *2022 14th International Conference on Computer and Automation Engineering (ICCAE)*, 2022, pp. 1–7.
- [17] Iwankowicz and R.R., “An efficient evolutionary method of assembly sequence planning for shipbuilding industry,” *Assembly Automation*, vol. 36, no. 1, pp. 60–71, 2016.
- [18] W.-C. Chen, P.-H. Tai, W.-J. Deng, and L.-F. Hsieh, “A three-stage integrated approach for assembly sequence planning using neural networks,” *Expert Systems with Applications*, vol. 34, no. 3, pp. 1777–1786, 2008.
- [19] C. Sinanoğlu and H. R. Börklü, “An assembly sequence-planning system for mechanical parts using neural network,” *Assembly Automation*, 2005.
- [20] A. Rashid and M.F.F., “A hybrid ant-wolf algorithm to optimize assembly sequence planning problem,” *Assembly Automation*, vol. 37, no. 2, pp. 238–248, 2017.
- [21] S. Nguyen, O. S. Oguz, V. N. Hartmann, and M. Toussaint, “Self-supervised learning of scene-graph representations for robotic sequential manipulation planning,” in *Conference on Robot Learning (CoRL)*, 2020, pp. 2104–2119.
- [22] V. Bapst, A. Sanchez-Gonzalez, C. Doersch, K. Stachenfeld, P. Kohli, P. Battaglia, and J. Hamrick, “Structured agents for physical construction,” in *ICML*. PMLR, 2019, pp. 464–474.
- [23] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, “Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.
- [24] N. Funk, G. Chalvatzaki, B. Belousov, and J. Peters, “Learn2assemble with structured representations and search for robotic architectural construction,” in *CoRL*. PMLR, 2022, pp. 1401–1411.
- [25] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, “Object-centric forward modeling for model predictive control,” in *CoRL*. PMLR, 2020, pp. 100–109.
- [26] R. Li, A. Jabri, T. Darrell, and P. Agrawal, “Towards practical multi-object manipulation using relational reinforcement learning,” in *2020 IEEE ICRA*. IEEE, 2020, pp. 4051–4058.
- [27] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *ICML*. PMLR, 2017, pp. 1263–1272.
- [28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *Stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [29] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *The world wide web conference*, 2019, pp. 2022–2032.
- [30] R. Bellman, “A markovian decision process,” *Journal of mathematics and mechanics*, pp. 679–684, 1957.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [32] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
- [33] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010, vol. 520.
- [34] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
- [35] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” 5 2019. [Online]. Available: https://github.com/pyg-team/pytorch_geometric
- [36] A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, “Learning feasibility for task and motion planning in tabletop environments,” *IEEE RAL*, vol. 4, no. 2, pp. 1255–1262, 2019.

A.5. Publication 5

Jianxiang Feng*, Matan Atad*, Ismael Rodríguez, Maximilian Durner, Stephan Günemann, Rudolph Triebel (2023): “*Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly*”. In Workshop on Robotics and AI: The Future of Industrial Assembly Tasks , Robotics: Science and Systems (RSS) 2023.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- <https://arxiv.org/pdf/2307.01317.pdf>

Please refer to the final published version for citation:

```
@inproceedings{feng2023density,
  title = {Density-based Feasibility Learning with Normalizing Flows for
    Introspective Robotic Assembly},
  booktitle = {18th Robotics: Science and System 2023 Workshops},
  year = {2023},
  author = {Feng, Jianxiang and Atad, Matan and Rodriguez Brena, Ismael
    Valentin and Durner, Maximilian and Triebel, Rudolph},
  keywords = {Feasibility learning; Normalizing Flows;},
  url = {https://elib.dlr.de/195846/},
}
```

Density-based Feasibility Learning with Normalizing Flows for Introspective Robotic Assembly

Jianxiang Feng^{*†}, Matan Atad^{*‡}, Ismael Rodríguez[†], Maximilian Durner[†],
Stephan Günnemann[‡] and Rudolph Triebel[†]

[†]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82110 Wessling, Germany

[‡]Department of Informatics, Technical University of Munich, 85748 Garching, Germany

^{*}Equal contribution. jianxiang.feng@dlr.de, matan.atad@tum.de

Abstract—*Machine Learning (ML) models in Robotic Assembly Sequence Planning (RASP) need to be introspective on the predicted solutions, i.e. whether they are feasible or not, to circumvent potential efficiency degradation. Previous works need both feasible and infeasible examples during training. However, the infeasible ones are hard to collect sufficiently when re-training is required for swift adaptation to new product variants. In this work, we propose a density-based feasibility learning method that requires only feasible examples. Concretely, we formulate the feasibility learning problem as Out-of-Distribution (OOD) detection with Normalizing Flows (NF), which are powerful generative models for estimating complex probability distributions. Empirically, the proposed method is demonstrated on robotic assembly use cases and outperforms other single-class baselines in detecting infeasible assemblies. We further investigate the internal working mechanism of our method and show that a large memory saving can be obtained based on an advanced variant of NF.*

I. INTRODUCTION

To embrace the trend of shorter product life cycles and greater customization, RASP empowered with ML models for productivity enhancement has received more attention over the past years [2, 13, 11, 23]. However, *data-driven* models are reported to behave unreliably with inputs differing from the training distribution, e.g., assemblies with distinct customization [16]. In other words, the assembly robot is *unaware* of the predicted solution’s feasibility, which requires an intrinsic understanding of the geometry of assemblies and the capability of the robotic system [12]. This introspective capability is essential for learning-enabled robots to adapt their knowledge and avoid catastrophic consequences [7]. The lack of introspection in RASP can lead to prolonged planning time induced by re-planning after failed execution of an infeasible plan. To address this issue, feasibility learning has been studied [19, 6, 20, 21, 2] based on a setting with *infeasible assemblies included*. We argue that this setting is undesirable in practice because of the risk of incomplete coverage of all possible infeasible cases and high time costs for generating sufficient infeasible training cases. These aggravate the situation when flexible and efficient adaptation across different product variants is required.

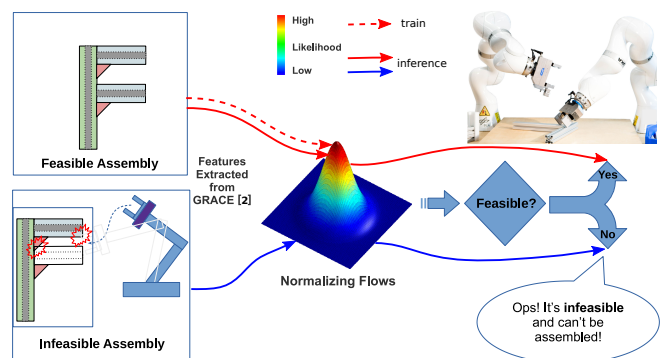


Fig. 1: **Overview of the proposed method** on an assembly scenario with a dual-armed robotic system (used in our setting). The distribution of feasible assemblies is modeled during training with NF. In test time, infeasible assemblies are identified by their low-likelihood.

To establish introspection for assembly robots with *only feasible assemblies* in mind, we seek to model the feasibility of an assembly with NF, which are a powerful class of generative models excelling at density estimation [5]. Concretely, we train the NF model with *Maximum Likelihood Estimation* (MLE) based on *feasible assemblies alone* to estimate the density of *In-Distribution* (ID) data, i.e. feasible assemblies. Hence, infeasible assemblies can be detected via a lower predicted likelihood as *Out-of-Distribution* (OOD).

We examine the proposed idea in a robotic assembly use case, in which different types of aluminum profiles are assembled with a dual-armed robot to create target structures (see Fig. 1). We collected assembly data in simulation and trained the NF on features of *only feasible* assemblies extracted from the *Graph Assembly Processing Networks* (GRACE) proposed in [2]. The NF model is then used to predict the likelihood of test data which includes both feasible and infeasible assemblies. As we learn the feasibility by estimating the density of feasible cases, the predicted outputs from NF represent how likely the given assemblies are feasible. Based on a threshold selected on a validation set, we can then detect infeasible

assemblies. Empirically, we demonstrate better results with the proposed method against other baselines on detecting infeasible assemblies in terms of *Area Under the Receiver Operating Characteristic Curve* (AUROC) in the setting where only feasible assemblies are available. We further investigate the major contributing factors of NF and significantly decrease the memory costs (i.e., number of network layers) by employing a more elaborate base distribution [17].

II. RELATED WORK

A. Feasibility Learning

The major body of work on feasibility learning is concentrated on plan or action feasibility learning in TAMP, while our goal is to learn the feasibility of assemblies directly by distilling the knowledge of assembly geometry and capability of the robot system. Wells et al. [19] trained a feature-based SVM model to directly predict the feasibility of an action sequence based on experience, which is hard to scale to scenarios with different numbers and types of objects. Driess et al. [6] and a recent follow-up [20] predict if a mixed-integer program can find a feasible motion for a required action based on visual input. Besides, Yang et al. [21] predict a plan’s feasibility with a transformer-based architecture using multi-model input embeddings. Atad et al. [2] introduced GRACE, a graph-based feature extractor for assemblies, capable of identifying infeasible assemblies when trained with both feasible and infeasible cases. Different from us, these methods work in a two-class setting, requiring failing action sequences to be included in the training set and then use binary feasibility classifiers.

B. Normalizing Flows for Out-of-Distribution Detection

NF [4] are a family of deep generative models with expressive modeling capability for complex data distributions where both sampling and density evaluation can be efficient and exact. Among a diverse set of flow architectures, Affine Coupling Flows [5] have gained huge popularity for their scalability to big data with high dimensionality and efficiency for both forward and inverse evaluation. These merits make NF more practically advantageous for OOD detection [9] when compared with other more principled but run-time inefficient uncertainty estimation methods [10]. In the context of task-relevant OOD detection, the practice of PostNet [3] of operating on feature embeddings, provides a more reasonable modeling ability. The potentials of NF for OOD detection have been demonstrated in other domains [14, 22], inspiring us to use them for feasibility learning.

III. METHOD

A. Problem Setting

Our goal is to predict the feasibility of assemblies relying only on feasible ones by formulating the problem as an OOD detection. Given a data-set \mathcal{D} of N feature embeddings of feasible assemblies $\{\mathbf{a}_i\}_{i=1}^N$, where $\mathbf{a}_i \in \mathcal{R}^h$ is drawn from an unknown distribution $P_{feasible}$ with *Probability Density Function* (PDF) p_f , a density estimator, denoted by $q_\theta : \mathcal{R}^h \rightarrow \mathcal{R}$,

approximates the true p_f with MLE for its parameters θ based on \mathcal{D} . During inference, given a threshold $\delta \in \mathcal{R}$, the feature of a test assembly $\hat{\mathbf{a}}_i$ is classified as OOD, i.e. infeasible, if $q_\theta(\hat{\mathbf{a}}_i) < \delta$, otherwise as ID, i.e. feasible.

B. Density-based Learning with NF

In this work, NF are used to estimate the density of feasible assemblies. NF, denoted by $f_\theta : \mathcal{R}^h \rightarrow \mathcal{R}^h$, are defined by a chain of *diffeomorphisms* (invertible and differentiable mappings) that transform a base distribution $p(\mathbf{z})$, $\mathbf{z} \in \mathcal{R}^h$ (e.g. an isotropic Gaussian) to the data distribution q_θ (in our case p_f). Based on the Change-of-Variables formula, the likelihood of an embedding of an assembly is obtained by

$$q_\theta(\mathbf{a}) = p(f_\theta^{-1}(\mathbf{a})) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \quad (1)$$

θ is optimized with MLE based on feasible data only, where the log likelihood is defined as:

$$\log q_\theta(\mathbf{a}) = \log p(f_\theta^{-1}(\mathbf{a})) + \log \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{a})}{\partial \mathbf{a}} \right) \right| \quad (2)$$

To this end, the inverse flow f^{-1} and the log determinant of the Jacobian need to be tractable and efficient. We employ the Real-NVP [5] that is composed of multiple layers of affine coupling flows. As the input to the NF, a data-set of feature embeddings for feasible assemblies \mathcal{D} is extracted from a pre-trained GRACE [2], which represents each assembly structure as a graph of its parts and their respective surfaces. To create a single feature embedding per assembly, a channel-wise mean pooling is applied on the graph’s part nodes. Different to previous works, the dimension of this embedding is independent of the number of assembly parts.

During inference, given a test assembly embedding, the trained NF q_θ predicts a log-likelihood score and determines its feasibility based on a pre-defined threshold δ , which was selected with a validation set.

IV. EXPERIMENTS

A. Data-set

We applied our in-house simulation software MediView to randomly generate synthetic assemblies, each with 5 or 6 aluminum parts. The software was tasked with putting together these structures with brute-force search while considering geometry restrictions and those imposed by the capabilities of the dual-armed robotic system *KUKA LBR Med* (seen in Fig. 1). We label structures that were successfully assembled as feasible and ones for which the software failed as infeasible. The resulting data-set consists of 6036 5-parts and 2865 6-parts assemblies. For the training set, we used feasible-labeled assemblies alone. The validation and testing sets were balanced with both feasible and infeasible assemblies¹.

¹This is still a single-class training setting since the validation set is only used for model selection.

| Classifier | AUROC (\uparrow) | |
|---|----------------------|-------------|
| | 5-parts | 6-parts |
| GRACE + NF, Gaussian dist., 749 layers (ours) | 0.85 | 0.83 |
| GRACE + NF, Resampling dist., 109 layers (ours) | 0.83 | - |
| OC-SVM [15] | 0.74 | 0.59 |
| GRACE [2], feasible-only setting | 0.61 | 0.57 |

TABLE I: Feasibility classifiers AUROC score on balanced test sets of 5- and 6-part assemblies.

B. Implementation Details

We pre-trained GRACE [2] with its default parameters to retrieve a 94-dimensions embedding per assembly. We implemented the NF model using [18] and experimented with Gaussian and Resampling [17] base distributions². For training the NF, we chose a batch size of 32 and a learning rate of $1e - 5$ with Adam optimizer. The number of coupling flows was chosen with hyper-parameter search on a validation set. Each affine coupling flow contained 4 layers with 94 hidden channels per layer.

We measure the separation between the feasibility classes with the binary classification metrics *False Positive Rate* (FPR) and *True Positive Rate* (TPR) to derive an AUROC score. In this setting, a positive instance is a feasible assembly and a negative an infeasible one.

C. Results

In Table I, we compare our method to baselines on predicting the feasibility of 5- and 6-part assemblies. The NF model with Gaussian base distribution achieves the highest score with a deep 749-layered network, outperforming the *One-class SVM* (OC-SVM) [15] and the naive GRACE [2]. In this setting, GRACE, trained on *feasible assemblies only*, predicts an assembly sequence for a test instance and infer the assembly’s feasibility based on the success of its sequencing process. More practically relevant, the NF variant with the more expressive Resampling base distribution [17] can reach comparably good results with a much smaller network (109 vs. 749 layers). This benefit of memory efficiency is highly relevant for robotic systems with only restricted computation resources (e.g., mobile manipulators). Contrary to GRACE’s sequencing process, we only require a single-pass through the feature-extraction pipeline, independent of the size of the assembly, and could therefore determine the feasibility of multiple batched assemblies at once.

D. Discussion

For an insight into how NF works on feasibility learning, we study the impacts of the flow transformations from the perspectives of two quantities: 1. likelihoods; 2. sample coordinates. While the former represents the density estimation ability of NF, the latter provides us a hint on how NF shifts the samples from the flow input space into its latent space.

²Code and training data are available at <https://github.com/DLR-RM/GRACE>.

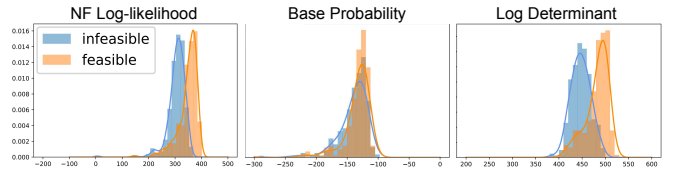


Fig. 2: **NF log-likelihoods for feasible and infeasible assemblies** with 5-parts (left), is a sum of the base probability (middle) and the transformation matrices log determinant (right). Best viewed in color.

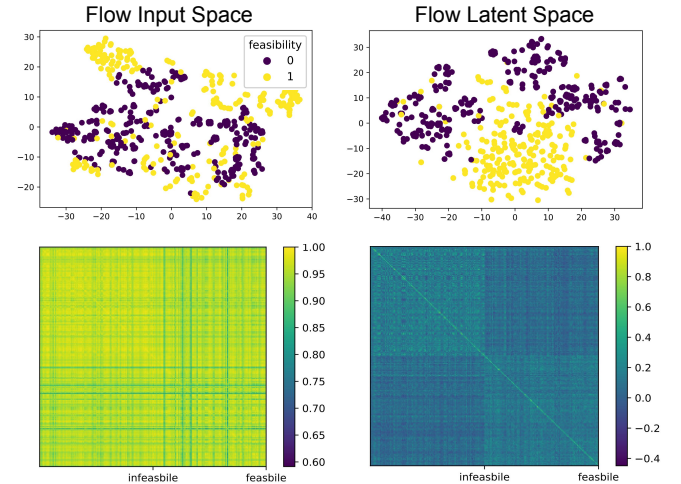


Fig. 3: **Samples visualization in NF input (left) and latent (right) spaces.** At the top, *t-distributed Stochastic Neighbor Embedding* (t-SNE) shows that samples mapped by NF are “normalized”, pulled together to a compact cluster. At the bottom, *Cosine Similarities* between feasible and infeasible assemblies are more distinct after the transformation, verifying the “normalization”. Best viewed in color.

a) *Likelihoods Ablation:* The NF log-likelihood estimation in Eq. 2 is a sum of two terms: the density of the base distribution and the log-determinant of the Jacobian of the flow transformation. To understand the contribution of each of these to the model’s estimation, we plot their values separately for the model with Gaussian base distribution in Fig. 2. As expected, the determinants are the main contributing factor to the final scores, whereas the values produced by the base distribution act as a normalization term.

b) *Samples Visualization:* We visualize the coordinates of the embeddings in the input space (as created by the GRACE feature extractor) and in the NF latent space with t-SNE and similarity matrices (Fig. 3). As shown in the t-SNE visualization, the samples of feasible assemblies are pulled together and hence clustered more compactly when compared to those in the input space before the flow transformation. This is verified again in the similarity matrices at the bottom, where the distances between feasible samples are smaller than those of infeasible ones after. These results show us that the flow transformation indeed “normalizes” the inputs in terms

of both likelihood computation and geometrical coordinates. This observation also confirms the finding of better OOD detection performance in the flow latent space [8], which is worth exploring for more effective feasibility learning algorithms, which we leave for future work. Besides, a further improvement could be archived by encouraging the feature extractor GRACE to grasp semantics that are more closely related to the feasibility task, as suggested by [9].

V. CONCLUSION

In this work, we seek to address feasibility prediction for data-driven methods in RASP with NF relying only on feasible examples. With the formulation of density-based OOD detection, we develop an effective feasibility prediction algorithm based on feature embeddings from a pre-trained processing network. The empirical experiments on detecting infeasible assemblies in simulation present promising results, which outperform the baselines. We further dug into the internal working mechanism of NF for this use case and found insightful observations, which can provide more understanding to inspire other researchers for further improvements in this direction. For future research, we suggest introducing explainability into this setting with a gradient map in respect to the input, which can guide the user in altering the structure and enable its assembly, i.e., counter-factual explanation [1].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their thoughtful feedback. Jianxiang Feng is supported by the Munich School for Data Science (MUDS). Rudolph Triebel and Stephan Günnemann are members of MUDS.

REFERENCES

- [1] Matan Atad, Vitalii Dmytrenko, Yitong Li, Xinyue Zhang, Matthias Keicher, Jan Kirschke, Bene Wiestler, Ashkan Khakzar, and Nassir Navab. Chexplaining in style: Counterfactual explanations for chest x-rays using stylegan. *arXiv preprint arXiv:2207.07553*, 2022.
- [2] Matan Atad, Jianxiang Feng, Ismael Rodríguez, Maximilian Durner, and Rudolph Triebel. Efficient and feasible robotic assembly sequence planning via graph representation learning. *arXiv preprint arXiv:2303.10135*, 2023.
- [3] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33: 1356–1367, 2020.
- [4] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- [5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [6] Danny Driess, Ozgur Oguz, Jung-Su Ha, and Marc Toussaint. Deep visual heuristics: Learning feasibility of mixed-integer programs for manipulation planning. In *2020 IEEE ICRA*, pages 9563–9569. IEEE, 2020.
- [7] Jianxiang Feng, Maximilian Durner, Zoltán-Csaba Márton, Ferenc Bálint-Benczédi, and Rudolph Triebel. Introspective robot perception using smoothed predictions from bayesian neural networks. In *Robotics Research: The 19th International Symposium ISRR*, pages 660–675. Springer, 2022.
- [8] Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *ICLR*, 2022.
- [9] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.
- [10] Jongseok Lee, Matthias Humt, Jianxiang Feng, and Rudolph Triebel. Estimating model uncertainty of neural networks in sparse information form. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 5702–5713. PMLR, 13–18 Jul 2020.
- [11] Lin Ma, Jiangtao Gong, Hao Xu, Hao Chen, Hao Zhao, Wenbing Huang, and Guyue Zhou. Planning assembly sequence with graph transformer. *arXiv preprint arXiv:2210.05236*, 2022.
- [12] Ismael Rodríguez, Korbinian Nottensteiner, Daniel Leidner, Michael Kaßecker, Freck Stulp, and Alin Albu-Schäffer. Iteratively refined feasibility checks in robotic assembly sequence planning. *IEEE RAL*, 4(2):1416–1423, 2019.
- [13] Ismael Rodríguez, Korbinian Nottensteiner, Daniel Leidner, Maximilian Durner, Freck Stulp, and Alin Albu-Schäffer. Pattern recognition for knowledge transfer in robotic assembly sequence planning. *IEEE RAL*, 5(2): 3666–3673, 2020.
- [14] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but different: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF WACV*, pages 1907–1916, 2021.
- [15] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.
- [16] Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.
- [17] Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Resampling base distributions of normalizing flows. In *AISTATS*, pages 4915–4936. PMLR, 2022.
- [18] Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A PyTorch Package for Normalizing Flows. *arXiv preprint*

arXiv:2302.12014, 2023.

- [19] Andrew M Wells, Neil T Dantam, Anshumali Shrivastava, and Lydia E Kavraki. Learning feasibility for task and motion planning in tabletop environments. *IEEE RAL*, 4(2):1255–1262, 2019.
- [20] Lei Xu, Tianyu Ren, Georgia Chalvatzaki, and Jan Peters. Accelerating integrated task and motion planning with neural feasibility checking. *arXiv preprint arXiv:2203.10568*, 2022.
- [21] Zhutian Yang, Caelan Reed Garrett, and Dieter Fox. Sequence-based plan feasibility prediction for efficient task and motion planning. *arXiv preprint arXiv:2211.01576*, 2022.
- [22] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *Computer Vision—ECCV 2020, Proceedings, Part III 16*, pages 102–117. Springer, 2020.
- [23] M. Zhao, X. Guo, X. and Zhang, Y. Fang, and Y. Ou. Asp-w-drl: assembly sequence planning for workpieces via a deep reinforcement learning approach. *Assembly Automation*, 40:65–75, 2020. ISSN 0144-5154.