CommonRoad-CARLA Interface: Bridging the Gap between Motion Planning and 3D Simulation

Sebastian Maierhofer and Matthias Althoff

***Abstract*— Motion planning algorithms should be tested on a large, diverse, and realistic set of scenarios before deploying them in real vehicles. However, existing 3D simulators usually focus on perception and end-to-end learning, lacking specific interfaces for motion planning. We present an interface for the CARLA simulator focusing on motion planning, e.g., to create configurable test scenarios and execute motion planners in interactive environments. Additionally, we introduce a converter from lanelet-based maps to OpenDRIVE, making it possible to use CommonRoad and Lanelet2 maps in CARLA. Our evaluation shows that our interface is easy to use, creates new scenarios efficiently, and can successfully integrate motion planners to solve CommonRoad scenarios. Our tool is published as an open-source toolbox at commonroad.in.tum.de.**

I. Introduction

Motion planning algorithms for autonomous vehicles require intensive virtual testing before evaluating them on test tracks or public roads. However, the motion planning research community lacks an easy-to-use 3D environment with a simple and well-supported motion planning integration. One of the most used 3D simulation environments in research is CARLA [1], which offers many features, e.g., sensor simulation, different weather conditions, multi-agent simulation, and a traffic manager. A widely used motion planning environment is CommonRoad [2], providing benchmark scenarios and many supporting tools, e.g., for drivability checking [3], manual scenario designing and converting maps between different formats [4], verifying and repairing maps [5], evaluating different criticality measures on traffic participants [6], reachability analysis [7], and reinforcement learning [8]. Some CommonRoad tools are already used by the CARLA community, e.g., the converter from OpenDRIVE[1] to lanelet-based maps [4], [9]. In this paper, we present a framework for coupling CommonRoad with CARLA, enabling the motion planning community to benefit from the combined features of CARLA and CommonRoad.

A. Related Work

We separate the literature review into a) open-source autonomous driving simulators, and b) the usage of CARLA in research.

a) Simulators: A comparison of several simulators for testing autonomous vehicles is provided in [10], where CARLA and LGSVL[2] [11] are the best simulators based on the authors' evaluation criteria. CARLA also has extensions, e.g., for crowded traffic environments [12] and integrating external real-world datasets [13]. SUMO [14] is a simulator for microscopic traffic simulation, which can also be utilized as a backend to create interactive scenarios for motion planning [15]. The Waymo and nuPlan datasets [16], [17] provide simulators to realize interactive simulations [17], [18] based on recorded datasets. Many simulators are specialized in working with OpenScenario[3], e.g., OpenPASS[4] and Esmini[5]. Other simulators released in the last years are SimVC [19], which focuses on perception tasks for different robotic applications, MetaDrive [20], which focuses on reinforcement learning and can replay different datasets, and OpenTrafficSim[6], which combines micro-simulation, macro-simulation, and meta-simulation in one tool. Our simulator review emphasizes that currently only CARLA provides photo-realistic environments, partially also considers motion planning, and supports different autonomous driving standards, e.g., OpenDRIVE, ROS[7], and OpenScenario.

b) Usage of CARLA: CARLA is one of the most used simulators in autonomous driving research. Due to the large number of publications related to CARLA, the referenced papers are only examples. CARLA can be used in different research areas, e.g., end-to-end learning [21], [22] or perception [23], [24]. Since CARLA provides photo-realistic environments and supports control via an external steering wheel, it can be used for user studies with a focus on motion planning [25] and human factors [26], [27]. For safe motion planning, CARLA has the Responsibility Safety Standard (RSS) [28] [29] integrated. Several authors use CARLA to show the usability of their motion planning approach in realistic environments, e.g., to evaluate traffic rule compliance [30]–[33], reinforcement learning [34], highway planning [35], and planning with a focus on critical situations [36]. To interact with CARLA, one needs to develop custom interfaces, which limits the exchange and comparison of research results and requires much work by researchers and developers.

B. Contributions

We present the first interface between CommonRoad and CARLA containing the following features:

- Conversion from lanelet-based maps to OpenDRIVE;

All authors are with the School of Computation, Information and Technology, Technical University of Munich, 85748 Garching, Germany.
{sebastian.maierhofer, althoff}@tum.de

[1] asam.net/standards/detail/opendrive
[2] LGSVL is not maintained anymore
[3] asam.net/standards/detail/openscenario-xml/
[4] openpass.eclipse.org
[5] esmini.github.io
[6] opentrafficsim.org
[7] ros.org

- An open-source tool for the efficient 3D visualization of CommonRoad scenarios with custom information;
- Interactive testing of motion planning algorithms;
- User-friendly creation of new motion planning benchmarks.

The remainder of this paper is structured as follows: Sec. II introduces CARLA, CommonRoad, and OpenDRIVE. The CommonRoad-CARLA interface and the CommonRoad to OpenDRIVE map conversion are presented in Sec. III. In Sec. IV, we present examples of map conversions, scenario generation, and motion planning. Sec. V concludes the paper.

II. Preliminaries

Subsequently, we introduce relevant information about CARLA, OpenDRIVE, and CommonRoad. We denote the vehicle controlled by a motion planner or human as the ego vehicle.

A. CARLA

CARLA is an open-source simulator for autonomous driving research. It is based on the Unreal Engine[8] and is implemented as a client-server system, where the server takes care of simulating and rendering the photo-realistic environment. The client is implemented in Python, controls agents, and receives sensor information. CARLA supports vehicle and walker actors. Subsequently, we use the terms walker and pedestrian interchangeably. CARLA also offers an OpenDRIVE standalone mode, which allows one to simulate traffic on maps without additional assets representing the 3D environment – however, this mode has a limited visual appeal.

B. OpenDRIVE

OpenDRIVE is a map format that allows one to model detailed road environments. For instance, it is possible to model the road surface and elevation, which is essential for driving simulators. OpenDRIVE[9] represents roads based on reference lines which are defined by several geometries, e.g., clothoid/Euler spiral (linear change of curvature), arcs (constant curvature), lines (zero curvature), and parametric cubic curves. A road can be constructed by the combination of several geometries. Fig. 1 shows a reference line consisting of a line, a clothoid, and an arc. Based on the reference line, lanes are defined in the lateral direction, where a third-order polynomial specifies their width. Fig. 2 visualizes an OpenDRIVE road consisting of several lane segments in both driving directions. For a more detailed format definition, we refer the interested reader to the OpenDRIVE documentation[10].

[8] `unrealengine.com`
[9] OpenDRIVE version 1.6
[10] `asam.net/standards/detail/opendrive`

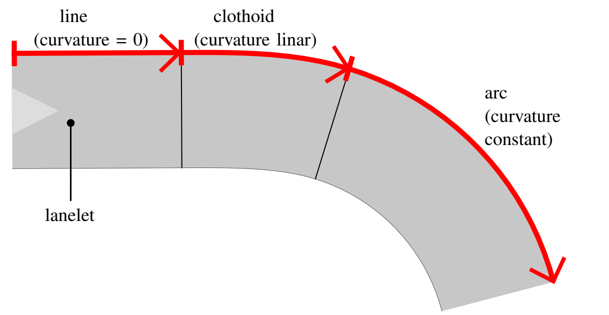


Fig. 1: OpenDRIVE reference line defined by a line, a cloithoid, and an arc. We also illustrate a corresponding CommonRoad lanelet.

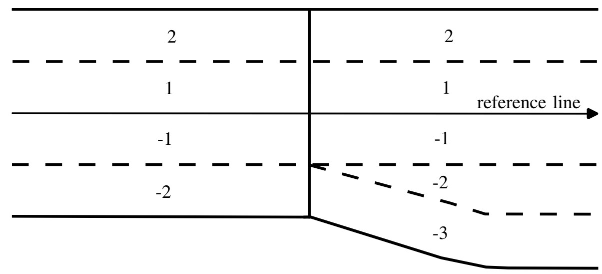


Fig. 2: OpenDRIVE road consisting of lanes in both driving directions (driving direction is indicated by positive/negative lane segment numbers).

C. CommonRoad Scenario Format

CommonRoad offers a scenario format for motion planning benchmarks. A CommonRoad scenario consists of dynamic elements, a map, and planning problems. CommonRoad maps are based on lanelets [37] (cf. Fig. 1), which are small drivable road segments specified by a left and right boundary. Roads are constructed by lanelets as illustrated in Fig. 5. The map format supports regulatory, semantic, and intersection elements. A formal description of the CommonRoad map elements is provided in [5]. Dynamic elements can be dynamic obstacles, digital traffic signs, or traffic lights. A dynamic obstacle consists of an initial state, shape, and prediction, e.g., a trajectory-based, set-based, or stochastic prediction. We refer to the CommonRoad documentation[11] for more details about the format.

III. CommonRoad-CARLA Interface

The CommonRoad-CARLA interface consists of two components: a) an interface for interacting with CARLA and b) a map conversion from CommonRoad to OpenDRIVE (and vice versa). The CommonRoad-CARLA interface is designed to achieve the following properties:

- Support for different applications: interactive motion planning, scenario generation, user studies, and replay of scenarios from different datasets in 3D (e.g., from CommonRoad [2], MONA [38], inD [39], or INTERACTION [40] datasets);
- Modularity: Everything can be configured in one common place;

[11] `commonroad.in.tum.de`

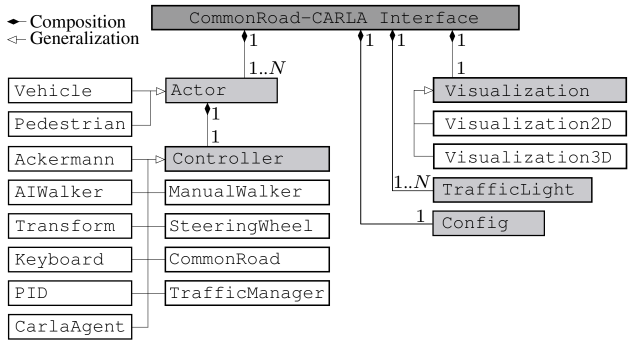


Fig. 3: UML class diagram of the CommonRoad-CARLA interface.

- Simplicity: Predefined interfaces and examples for the applications in focus;
- Extensibility: The code structure facilitates adding new features.

These requirements can also be seen in the unified modeling language (UML) class diagram representing the main parts of the CommonRoad-CARLA interface architecture (cf. Fig. 3). The *CommonRoad-CARLA interface* class is the central module, e.g., it takes care of initiating the connection to the CARLA server, controlling the CARLA agents, and executing different operating modes. The abstract *Actor* class represents CARLA actors and maps them to CommonRoad obstacles. The *Controller* classes specify the movement of CARLA actors. Separate classes take care of traffic light cycles, visualization, and configuration. We utilize the Python interface of CARLA to control obstacles, receive environmental information, and configure the simulation. We also support CARLA features for non-real-time-capable user code, e.g., synchronous simulations and off-screen mode. Subsequently, we introduce the different control mechanisms, map conversion, visualization customization, and scenario generation.

A. Controlling Dynamic Obstacles

We support different ways to control CARLA vehicles from CommonRoad:

1) Transform: We translate and rotate the actor as defined in an obstacle state.
2) Keyboard: Manual actor control via keyboard buttons.
3) TrafficManager: Control via CARLA's traffic manager.
4) CarlaAgent: Control via CARLA's example agent.
5) Ackermann: Applying Ackermann control based on an obstacle state.
6) SteeringWheel: Manual obstacle control via steering wheel and pedals.
7) PID: The obstacle state is provided to a proportional–integral–derivative (PID) controller, which controls the CARLA agent via steering angle and acceleration.

We also provide a simple and extendable motion planner interface, which allows one to use arbitrary motion planners and prediction algorithms. The interface requires an additional low-level controller propagating the planned states,

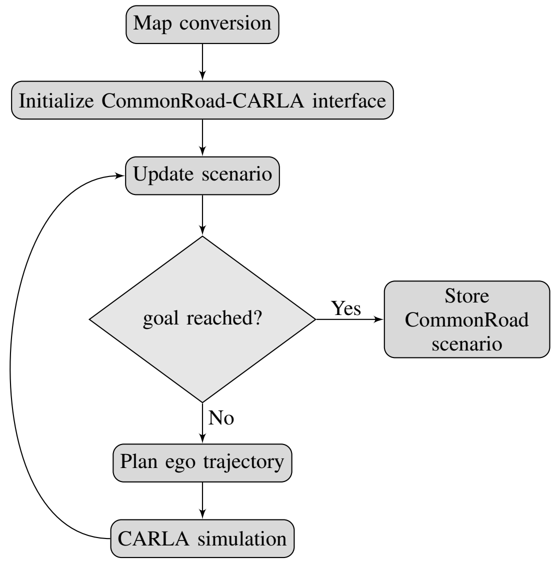


Fig. 4: Possible flow chart for using the CommonRoad-CARLA interface with a motion planner.

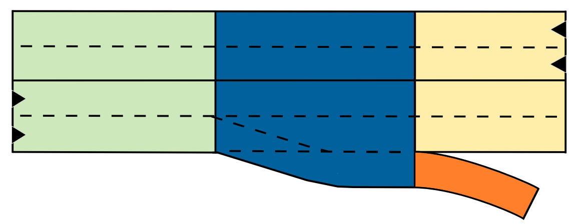


Fig. 5: Mapping of CommonRoad lanelets to OpenDRIVE roads. All lanelets with the same color are combined to an OpenDRIVE road.

i.e., method 1), 5), or 7). Fig. 4 illustrates a possible program flow when using the CommonRoad-CARLA interface with a motion planner, where the concrete program flow may vary depending on the user configuration. We automatically assign CommonRoad obstacles to CARLA vehicle models by comparing the available CARLA vehicle shapes and selecting the best match. Additionally, the following methods can control pedestrians:

1) AIWalker: The pedestrian walks automatically to a goal destination.
2) ManualWalker: The walker moves according to the direction and velocity defined in an obstacle state.

B. Map

To use CommonRoad maps in CARLA, we convert them to OpenDRIVE. Our conversion algorithm supports the arbitrary combination of clothoids, arcs, and lines. We create a) roads, b) junctions, c) static obstacles, and d) regulatory elements and combine them to an OpenDRIVE map. We construct the roads in a breadth-first search fashion, where a randomly selected lanelet is used as the starting lanelet. Iteratively, a lanelet is extended to the left and right to collect all lanelets belonging to an OpenDRIVE road. Fig. 5 shows an exemplary road network with lanelets belonging to different OpenDRIVE roads. The road is constructed

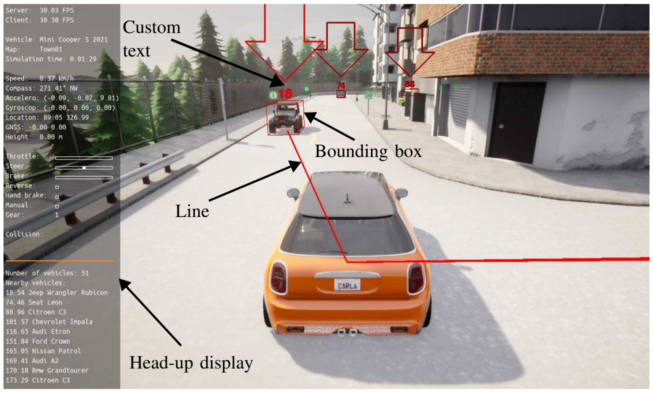


Fig. 6: Visualization of custom information.

by the following steps: 1) selecting a boundary polyline between different driving directions as a reference line; 2) matching the mentioned geometries to the polyline based on the curvature and heading of the reference line; 3) computing the remaining parameters, e.g., lateral offset and width.

To use maps provided by CARLA in CommonRoad, we convert them from OpenDRIVE to CommonRoad using our approach presented in [9]. Both conversion directions are available via our CommonRoad Scenario Designer [4]. We also support converting from Lanelet2 to OpenDRIVE and vice versa.

C. Visualization

For debugging and user studies, we provide an interface to customize visualizations. The interface supports the subsequent features (each with custom coloring):

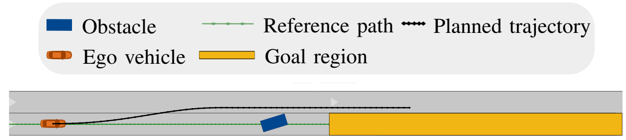
- Bounding boxes for objects (cars, pedestrians, traffic signs);
- Lines between the ego vehicle and selected other obstacles;
- Custom text above other objects, e.g., distance, object ID, or velocity.

Additionally, the head-up display provided by the CARLA examples is configurable. Fig. 6 highlights the supported features. We provide configurations for several screens. Therefore, our interface enables the usage of CARLA in user studies without the need for costly driving simulators. The user can also save images and videos of the ego vehicle cameras. For 2D visualizations, we support the birds-eye view visualization of CARLA.

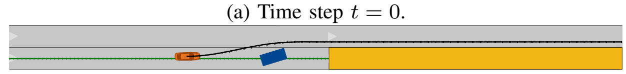
D. Scenario Generation

The CommonRoad-CARLA interface can create CommonRoad scenarios based on the traffic manager of CARLA, which considers more realistic dynamics compared to microscopic traffic simulators like SUMO, e.g., a kinematic single-track model. Moreover, CARLA allows us to specify general high-level behavior, e.g., the traffic light violation rate. We support the following configuration parameters of the CARLA traffic manager:

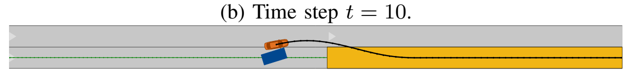
1) Minimum distance to the leading vehicle;



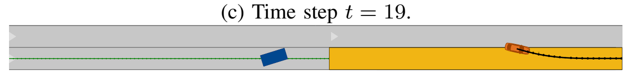
(a) Time step $t = 0$.



(b) Time step $t = 10$.



(c) Time step $t = 19$.

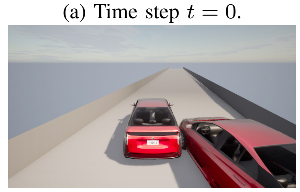


(d) Time step $t = 38$.

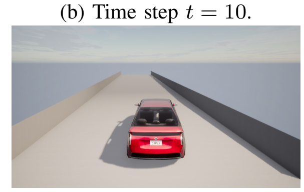
Fig. 7: CommonRoad motion planning example with scenario ZAM-CARLA-1_1_T-1.

(a) Time step $t = 0$.

(b) Time step $t = 10$.



(c) Time step $t = 19$.



(d) Time step $t = 38$.

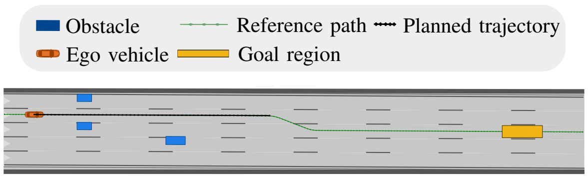
Fig. 8: CARLA visualization of the CommonRoad motion planning example of Fig. 7.

2) Difference to the speed limit;
3) Ignore a) stop signs and b) traffic lights;
4) Following keep right rule;
5) Ignoring collisions with a) pedestrians and b) vehicles;
6) Number of a) running and b) crossing pedestrians;
7) Lane offset from the center line;
8) Number of lane changes to a) left and b) right.

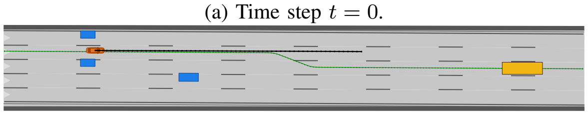
IV. Evaluation

Subsequently, we evaluate the motion planner interface, map conversion, and scenario generation. The scripts to reproduce the results are part of our open-source tool at commonroad.in.tum.de. For all evaluations, we use a time step size of $\Delta t = 0.1\mathrm{s}$. We successfully evaluated a sampling-based planner [41] together with a constant velocity prediction algorithm in CARLA on a CommonRoad scenario with a standing obstacle on the path of the ego vehicle (cf. Fig. 7 and 8) and a CARLA map with random traffic (cf. Fig. 9 and 10). Furthermore, we successfully converted CommonRoad maps to OpenDRIVE. Exemplary visualizations of CommonRoad and OpenDRIVE maps can be seen in Fig. 11-13.

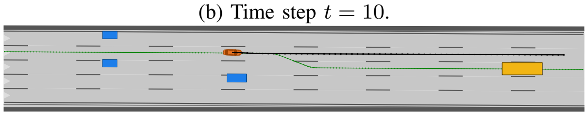
Lastly, we generated scenarios using five different CARLA maps. For parameter 1) we use $1\mathrm{m}$ and for parameters 1) -



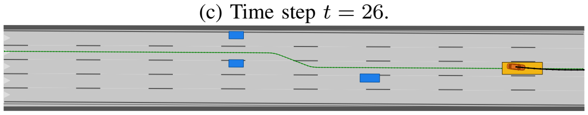
(a) Time step $t = 0$.



(b) Time step $t = 10$.



(c) Time step $t = 26$.

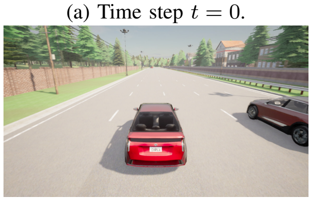


(d) Time step $t = 66$.

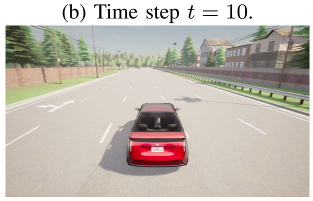
Fig. 9: Planned trajectories by a motion planner in CARLA's Town 6 world with random traffic.

(a) Time step $t = 0$.

(b) Time step $t = 10$.



(c) Time step $t = 26$.



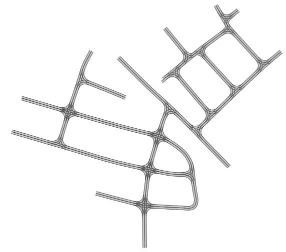
(d) Time step $t = 66$.

Fig. 10: CARLA visualization of the CommonRoad motion planning example of Fig. 9.

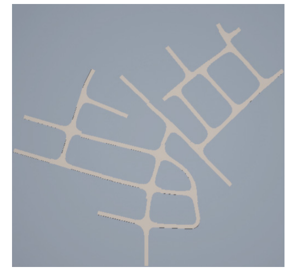
4) listed in Sec. III-D 25%. For all other parameters, we use the default values of CARLA. We generated scenarios with a maximum of 100 vehicles and 25 pedestrians and simulated each configuration for $5\mathrm{min}$. We created over $45\mathrm{h}$ of simulated traffic with more than $450\mathrm{km}$ of traveled distance. In Tab. I, statistical data of the generated scenarios for each map is listed. Fig. 14 shows a generated scenario in CARLA and CommonRoad at different time steps.

V. Conclusions

We present a modular interface for coupling the CommonRoad motion planning framework and CARLA. The CommonRoad-CARLA interface is the first tool interfacing CARLA to realize a) efficient creation of new motion planning scenarios, b) interactive motion planning, c) replaying artificial or recorded real-world scenarios in 3D, and d) visualizing custom information for user studies or debugging. The map conversion from lanelet-based maps to OpenDRIVE

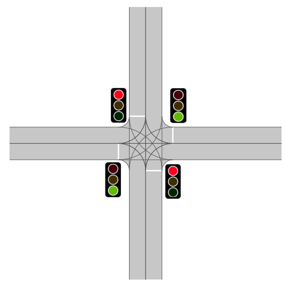


(a) CommonRoad.



(b) CARLA/OpenDRIVE.

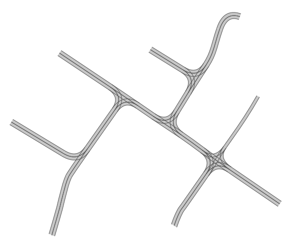
Fig. 11: Conversion example for CommonRoad map DEU_Backnang-1.



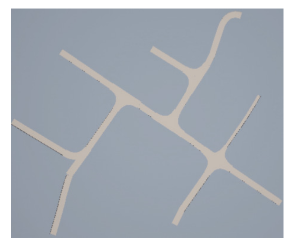
(a) CommonRoad.

(b) CARLA/OpenDRIVE.

Fig. 12: Conversion example for CommonRoad map ZAM_FourWay-1.



(a) CommonRoad.



(b) CARLA/OpenDRIVE.

Fig. 13: Conversion example for CommonRoad map DEU_Guetersloh-1.

extends our existing widely-used map conversion framework and supports researchers and practitioners, even if they are not using CARLA. The toolbox is available at commonroad.in.tum.de.

Acknowledgment

The authors gratefully acknowledge partial financial support by the BMW Group within the CAR@TUM project, the Free State of Bavaria, and the German Federal Ministry for Education and Research (BMBF) in the MCube project (grants 03ZU1105BA and 03ZU1105KA).

References

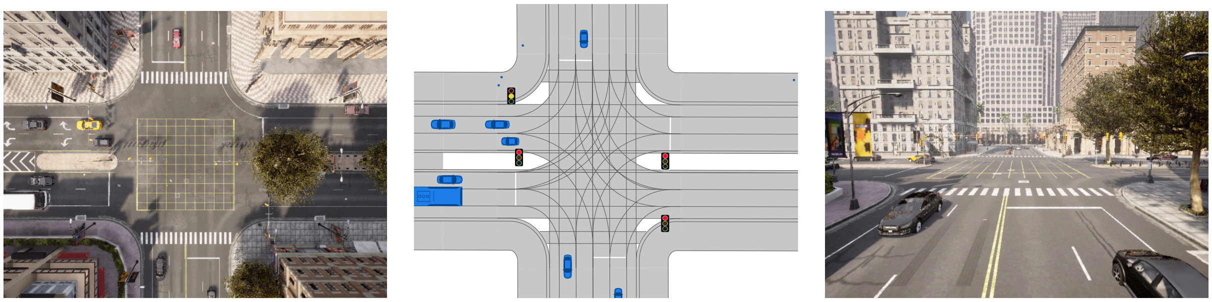
[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[2] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.

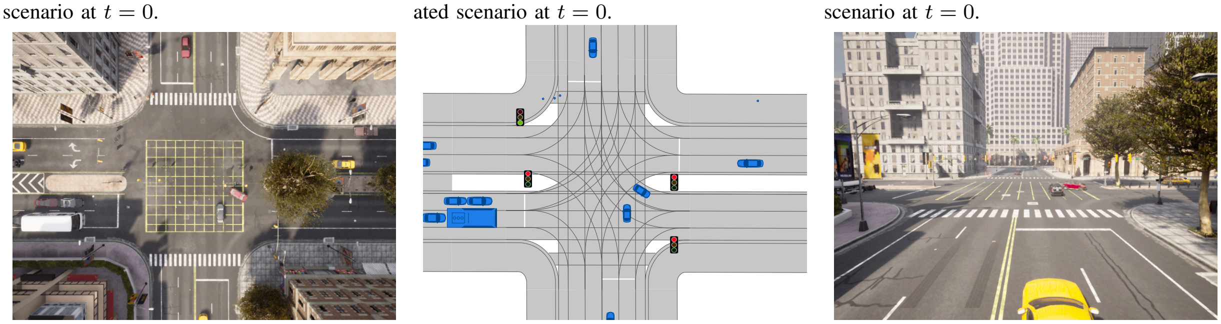
[3] C. Pek, V. Rusinov, S. Manzinger, M. C. Üste, and M. Althoff, "CommonRoad drivability checker: Simplifying the development and validation of motion planning algorithms," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2020, pp. 1013–1020.

TABLE I: Summary of the generated CommonRoad scenarios for each base map.

Map	**#Vehicles**	**#Pedestrians**	**Distance Traveled [km]**	**Time Traveled [h]**	**Lanelet Network [km]**	**#Intersections**
CARLA Town 1	100	17	84.98	9.05	7.68	8
CARLA Town 2	100	20	51.99	9.89	15.85	30
CARLA Town 4	100	22	212.97	9.55	13.48	9
CARLA Town 7	100	16	60.55	9.2	17.41	12
CARLA Town 10	100	23	67.67	9.49	58.23	27
Accumulated	500	98	478.16	47.18	112.65	86



(a) CARLA birds-eye view of a generated scenario at $t = 0$. (b) CommonRoad birds-eye view of a generated scenario at $t = 0$. (c) CARLA wide-angle view of a generated scenario at $t = 0$.



(d) CARLA birds-eye view of a generated scenario at $t = 55$. (e) CommonRoad birds-eye view of a generated scenario at $t = 55$. (f) CARLA wide-angle view of a generated scenario at $t = 55$.

Fig. 14: A snipped of a generated signalized intersection scenario visualized in CARLA and CommonRoad at two time steps with cars and pedestrians.

[4] S. Maierhofer, M. Klischat, and M. Althoff, "CommonRoad Scenario Designer: An open-source toolbox for map conversion and scenario creation for autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2021, pp. 3176–3182.

[5] S. Maierhofer, Y. Ballnath, and M. Althoff, "Map verification and repairing using formalized map specifications," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2023, pp. 1277–1284.

[6] Y. Lin and M. Althoff, "CommonRoad-CriMe: A toolbox for criticality measures of autonomous vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2023, pp. 1–8.

[7] E. I. Liu and M. Althoff, "Computing specification-compliant reachable sets for motion planning of automated vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2021, pp. 1037–1044.

[8] X. Wang, H. Krasowski, and M. Althoff, "CommonRoad-RL: A configurable reinforcement learning environment for motion planning of autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2021, pp. 466–472.

[9] M. Althoff, S. Urban, and M. Koschi, "Automatic conversion of road networks from OpenDRIVE to lanelets," in *Proc. of the IEEE Int. Conf. on Service Operations and Logistics, and Informatics*, 2018, pp. 157–162.

[10] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *Proc. of the Int. Conf. on Connected and Autonomous Driving*, 2021, pp. 62–70.

[11] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 1–6.

[12] P. Cai, Y. Lee, Y. Luo, and D. Hsu, "SUMMIT: A simulator for urban driving in massive mixed traffic," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 4023–4029.

[13] B. Osiński, P. Miłoś, A. Jakubowski, P. Zięcina, M. Martyniak, C. Galias, A. Breuer, S. Homoceanu, and H. Michalewski, "CARLA real traffic scenarios–novel training ground and benchmark for autonomous driving," 2020, arXiv preprint arXiv:2012.11329.

[14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2575–2582.

[15] M. Klischat, O. Dragoi, M. Eissa, and M. Althoff, "Coupling SUMO with a motion planning framework for automated vehicles," in *SUMO: Simulating Connected Urban Mobility*, 2019.

[16] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[17] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles," 2021, arXiv preprint arXiv:2106.11810.

[18] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, J. Harb, X. Pan, Y. Wang, X. Chen, J. Co-Reyes, R. Agarwal, R. Roelofs, Y. Lu,

N. Montali, P. Mougin, Z. Yang, B. White, A. Faust, R. McAllister, D. Anguelov, and B. Sapp, "Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research," in *Proc. of Advances in Neural Information Processing Systems*, 2023, pp. 7730–7742.

[19] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4CV: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, pp. 902–919, 2018.

[20] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3461–3475, 2023.

[21] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7077–7087.

[22] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2022.

[23] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2822–2829, 2021.

[24] H. Wang, Y. Chen, Y. Cai, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "SFNet-N: An improved SFNet algorithm for semantic segmentation of low-light autonomous driving road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 405–21 417, 2022.

[25] J. Karlsson, S. van Waveren, C. Pek, I. Torre, I. Leite, and J. Tumova, "Encoding human driving styles in motion planning for autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 1050–1056.

[26] J. Araluce, L. M. Bergasa, M. Ocaña, E. López-Guillén, R. Gutiérrez-Moreno, and J. F. Arango, "Driver take-over behaviour study based on gaze focalization and vehicle data in CARLA simulator," *Sensors*, vol. 22, no. 24, 2022.

[27] J. Xu, S. Pan, P. Z. H. Sun, S. Hyeong Park, and K. Guo, "Human-factors-in-driving-loop: Driver identification and verification via a deep learning approach using psychological behavioral data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3383–3394, 2023.

[28] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," 2017, arXiv preprint arXiv:1708.06374.

[29] B. Gassmann, F. Oboril, C. Buerkle, S. Liu, S. Yan, M. S. Elli, I. Alvarez, N. Aerrabotu, S. Jaber, P. van Beek, D. Iyer, and J. Weast, "Towards standardization of AV safety: C++ library for responsibility sensitive safety," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 2265–2271.

[30] E. Zapridou, E. Bartocci, and P. Katsaros, "Runtime verification of autonomous driving systems in CARLA," in *Proc. of the Int. Conf. on Runtime Verification*, 2020, pp. 172–183.

[31] A. Karimi and P. S. Duggirala, "Formalizing traffic rules for uncontrolled intersections," in *Proc. of the ACM/IEEE Int. Conf. on Cyber-Physical Systems*, 2020, pp. 41–50.

[32] Q. Zhang, D. K. Hong, Z. Zhang, Q. A. Chen, S. Mahlke, and Z. M. Mao, "A systematic framework to identify violations of scenario-dependent driving rules in autonomous vehicle software," *Proc. of the ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 2, pp. 1–25, 2021.

[33] J. Lin, W. Zhou, H. Wang, Z. Cao, W. Yu, C. Zhao, D. Zhao, D. Yang, and J. Li, "Road traffic law adaptive decision-making for self-driving vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2022, pp. 2034–2041.

[34] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2765–2771.

[35] F. M. Tariq, D. Isele, J. S. Baras, and S. Bae, "SLAS: Speed and lane advisory system for highway navigation," in *Proc. of the IEEE Conf. on Decision and Control*, 2022, pp. 6979–6986.

[36] W. Zhou, Z. Cao, N. Deng, X. Liu, K. Jiang, and D. Yang, "Dynamically conservative self-driving planner for long-tail cases," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3476–3488, 2023.

[37] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[38] L. Gressenbuch, K. Esterle, T. Kessler, and M. Althoff, "MONA: The munich motion dataset of natural driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2022, pp. 2093–2100.

[39] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2020, pp. 1929–1934.

[40] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive driving scenario with semantic maps," 2019, arXiv preprint arXiv:1910.03088.

[41] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét frame," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 987–993.