Generation of Training Data from HD Maps in the Lanelet2 Framework

Fabian Immel[1], Richard Fehler[1], Frank Bieder[1] and Christoph Stiller[2] *†

January 29, 2025

Abstract:

Using HD maps directly as training data for machine learning tasks has seen a massive surge in popularity and shown promising results, e.g. in the field of map perception. Despite that, a standardized HD map framework supporting all parts of map-based automated driving and training label generation from map data does not exist. Furthermore, feeding map perception models with map data as part of the input during real-time inference is not addressed by the research community. In order to fill this gap, we present lanelet2_ml_converter, an integrated extension to the HD map framework Lanelet2, widely used in automated driving systems by academia and industry. With this addition Lanelet2 unifies map based automated driving, machine learning inference and training, all from a single source of map data and format. Requirements for a unified framework are analyzed and the implementation of these requirements is described. The usability of labels in state of the art machine learning is demonstrated with application examples from the field of map perception. The source code is available embedded in the Lanelet2 framework under `https://github.com/fzi-forschungszentrum-informatik/Lanelet2/tree/feature_ml_converter`.

1 Introduction

HD Maps

In many of today's automated driving stacks, high-definition (HD) maps are used as an essential component. By providing static contextual information, HD maps can complement a vehicle's sensor suite and online perception, which can suffer from occlusion, limited sensor range or poor weather conditions. Due to their wide field of applications and the lack of an established map standard, HD maps can come in a wide range of formats. These map formats are usually tailored for one specific task or software stack and created by a specific map manufacturer, such as TomTom [1] and HERE [2]. Hence, map formats are hardly compatible and the corresponding software suite to create, maintain or further develop the map and its format is generally not publicly available.

*[1]FZI Research Center for Information Technology, Karlsruhe, Germany `{immel, fehler, bieder}@fzi.de`

†[2]Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany `stiller@kit.edu`
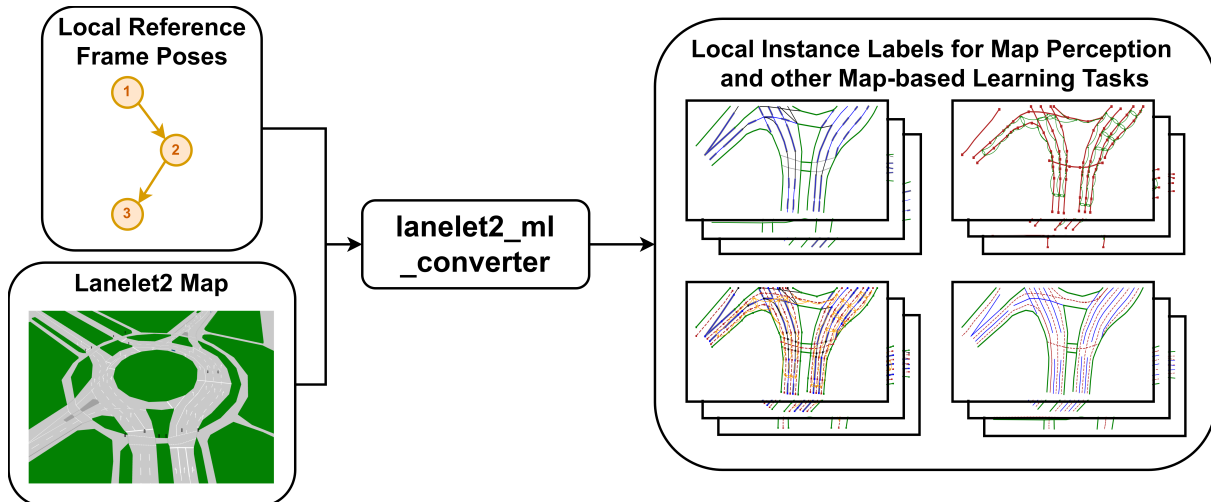
Figure 1: Usage and applications of the software module. Together with the module, a Lanelet2 map can be used as a unified format for both map-based highly automated driving and as labels for a variety of map perception and map-based learning tasks.

Lanelet2

In order to fill this gap, [3] provides the software map framework Lanelet2. It aims to define a unified map framework which is suitable for serving a comprehensive range of automated driving tasks. In contrast to many other map formats, e.g. OpenDRIVE [4], Lanelet2 is modelled bottom-up which allows a modular and flexible representation of complex road topologies and intersections. The relational and topological layers emerge bottom-up from the physical layer and form the basis of the Lanelet2 philosophy. The Lanelet2 framework is furthermore equipped with a set of tools for simple and standardized creation of new HD maps, data retrieval, handling of map information on multiple abstraction layers and efficient real-world application in automated driving systems. Since its release in 2018, Lanelet2 has been widely adopted and applied in various academic and industrial projects. Support has been demonstrated for many applications such as semantic localization [5], HD map creation [6] and behavior planning [7], [8]. Lanelet2 is the default HD map format in most datasets related to planning and prediction tasks [6]–[12]. In this context the research community developed tools to convert maps into the Lanelet2 format if the supplied map is not available in this format. The extensive functionalities of Lanelet2, efficient implementation and ROS 1/2 support [13] enable most modern academic and open-source automated driving stacks [14].

Online Map Perception

Map perception is one of the key technologies in scaling automated driving. Online map perception can compensate outdated or incomplete map data, extending the available geolocations of automated driving systems and offline map perception can support cloud mapping and map updates. Training of map perception models is feasible for academic research, thanks to the release of several large-scale sensor datasets [15]–[19], that include either city-wide or scenario-specific maps on a lane-level with corresponding sensor data. Usual maps released with sensor datasets are published in a custom format, with varying

fidelity and map content without a standardized API and format. These maps and supporting software have not demonstrated the ability to run classic map based automated driving software stacks such as Autoware [14] and do not implement or support features such as path planning, map validation or routing.

Contributions

In the following we summarize the main contributions of this work:

- We motivate the need for a unified HD map format and framework that is flexible enough to be used in fully map based driving, can derive map information of different abstraction levels and thus allows for training map perception models.

- We define a set of requirements for a framework that is suitable for generating training data from HD maps.

- Embedded in the Lanelet2 library, we design an extension of the framework that follows this set of requirements. We describe implementation details and provide source-code.

- We demonstrate how the lanelet2_ml_converter extension generates training data from Lanelet2 maps for a variety of recent map perception problems in different abstraction layers.

2 Requirements for Modern HD Map Frameworks

In this section we specify the requirements needed to exceed the state of the art provided by [3], [4] and especially datasets like nuScenes [18] or Argoverse 2 [20], for online map perception and learning from 3D map labels.

Unified HD Map Framework for Automated Driving and Deep Learning Inference and Training

To implement state of the art map perception in automated driving systems, a HD map framework needs to support training data generation as well as fast online local instance label retrieval in the necessary inference representation. Additionally support for classical tasks like manual and automatic mapping, map updates, behavior generation and path planning are necessary to build a automated driving system and can not be disregarded. Local instance label retrieval methods are implemented by [21][19], but their high runtime latency and a generally offline oriented software framework hinder progress in real-world application of these methods.

Generation of Training Labels from Maps

These local instance labels are used in offline applications like scalable label generation for perception, given a high-quality localization. Ideally these local instances include not only physically present features such as road markings, but also high level information such as yielding rules, the lane-graph structure given by the relationship to lane successors and neighbors, driving directions and relationships to traffic elements as signs and lights.

Traceability

In case of corrupted or false instance labels, it is necessary to backtrace them to their origin in the HD map. This demands a traceable connection between a instance label and the corresponding map element it is derived from. This can be extended up to the underlying mapping process of the map element, allowing for full quality control.

Map Validation

HD Maps included in publicly available automated driving datasets lack the provided software API or formal definition to check for validity regarding traffic rule compliance and traffic flow consistency. A map validation module within the map framework can ensure a more meaningful, consistent and less error-prone label generation. This is especially crucial for the generation of higher-level training labels, which are inferred from multiple interacting map annotations and are sensitive to incomplete or inconsistent map annotations. A well tested system for this is provided by the Lanelet2 validation software module [3], however it is not usable for map labels without a process to generate labels directly from Lanelet2 maps, like in the extension proposed in this paper.

Independence of Label Instances from Map Annotation Artifacts

Both manual and automatic mapping can produce almost infinitely many permutations of map element annotations, with different polyline point positions, ordering, breaks in polylines, and resulting lane subdivisions into lanelets. All these annotation permutations can represent the very same objects and information in the real world. To ensure consistency, a unambiguous representation of the corresponding instance label needs to be derivable for map elements.

Support for Varying Local Reference Frame Poses

Different datasets and mapping setups, produce reference frame poses with varying degrees of freedom. Training on custom setups and public datasets requires support for varying pose qualities.

Real-Time Capability

While the generation of training data is usually performed offline, an online application of the new module would be required to fuse the map perception with prior knowledge given by the HD map. This would require a high-performant implementation regarding the processing of selected map elements and their conversion to efficient numerical representations.

3 Implementation

This chapter describes the extension of the Lanelet2 framework implementing the requirements specified in section 2.

(a) Original map labels (b) Compound feat. of (c) Compound feat. of (d) Overlapping labels
                             path 1   path 2   deleted
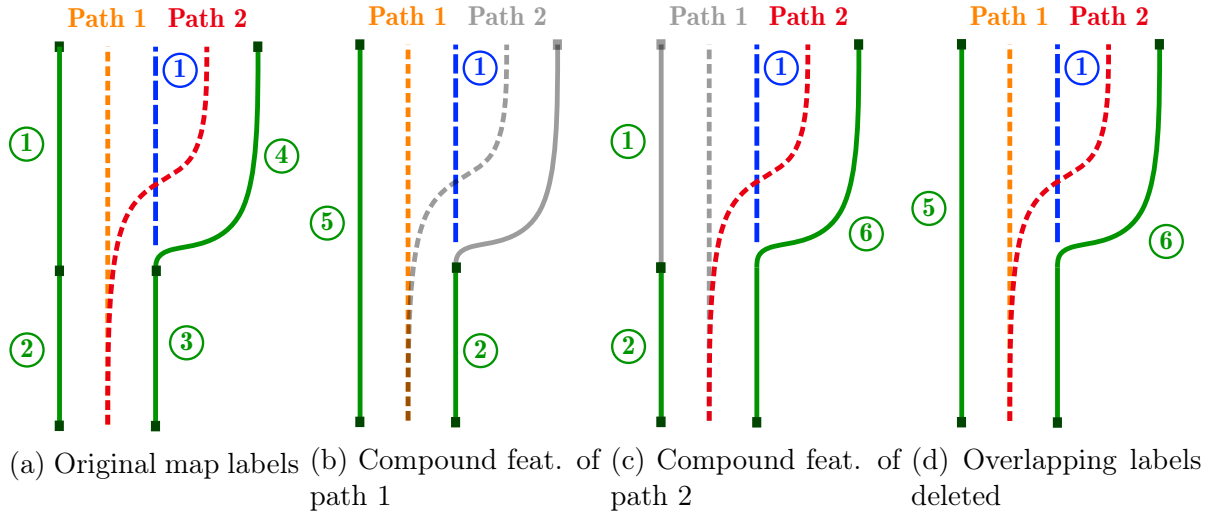
Figure 2: Visualization of compound labels and the elimination of overlaps. The overlapping compound road borders 2 and 4 are eliminated in favor of the larger compound road borders 5 and 6.

Compound Labels for Independence from Map Annotation Artifacts

To implement the essential requirement of independence from map annotation artifacts, both local and non-local, we make use of a concept we name "compound labels". The underlying idea is used in many online HD map construction models (e.g. [22], [23]) and replaces each coherent chain of individual polylines with one single polyline. Lanelet2 and most other map formats divide the road into smaller subsegments, often referred to as lanelets. While this has many advantages like map consistency and non-duplication, it also causes issues with local instance label generation for map learning.

A coherent chain of polylines of the same type can be replaced by a single line, greatly simplifying the learning task for the HD map construction model.

Compound Label Generation

The consistent and universally applicable generation of these compound labels encompasses a significant part of the implementation and is described here in greater detail. In general, the algorithm uses the routing graph from [24] of the region of interest submap to extract all possible lanelet paths in the submap and follows those paths to merge the individual lanelet boundaries into compound labels.

The path following is executed by a recursive algorithm, which is initialized by the set of lanelets without a predecessor in the routing graph of the local submap. A path of such a lanelets successors is followed until a lanelet no longer has any successors, recursively branching for a lanelet which has multiple successors. Figure 2 shows the paths generated in this fashion for an example road segment.

From the created paths, compound labels are composed of the left and right boundaries of the path lanelets as well as their centerlines, splitting at changes of the boundary type for the boundary labels. As illustrated by Figure 2, the right border of path 1 results in two compound labels due to the change from road border to dashed line.

Since the same lanelet can be passed through by multiple paths, multiple compound

labels containing the same map element can be created. This is also the case for lane borders shared by two adjacent lanelets. A map polyline can only ever be part of one compound label, so we can eliminate all but one of the overlapping compound labels. This is also shown in Figure 2, where the smaller compound road borders 2 and 4 of path 1 and 2 are eliminated in favor of the larger compound road borders 5 and 6, also from path 1 and 2 respectively. The elimination of overlapping labels is not performed for the lane centerlines, as the underlying assumption is not applicable in this case and the training with independent path labels has been shown to improve performance [25].

All local instance labels are cropped with respect to the region of interest and resampled with a fixed number of points if needed.

Traceability

(a) Original local instance labels   (b) Different local instance labels
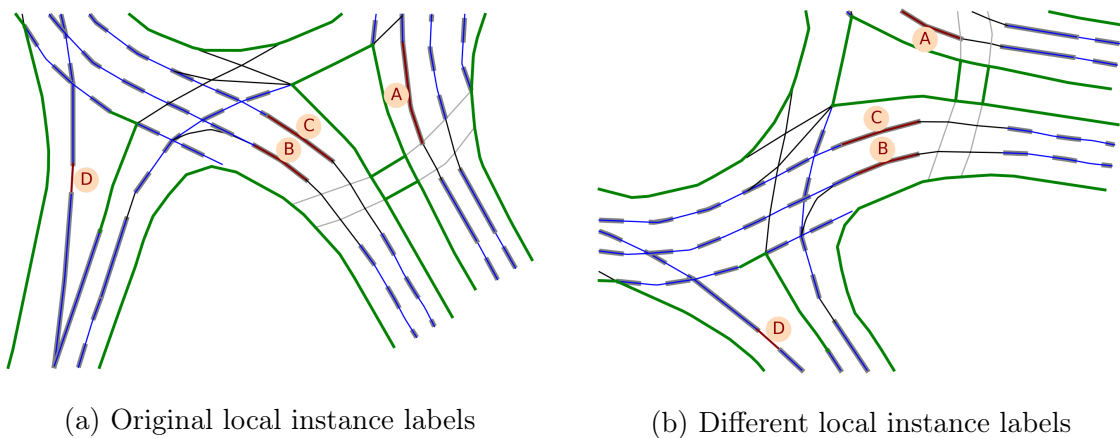
Figure 3: Traceability of labels in local instance labels. Every label is associated with the underlying map element, visualized here with the letters A to D. This association is maintained over local instance labels. Compound labels have traceability of the composing labels available as well.

A key requirement for ground truth generation in scale is the traceability of labels to the origin of their data, which in our case is the underlying Lanelet2 map. This requirement is fulfilled here using the unique ID of each map element in a Lanelet2 map. The ID of the underlying map element is available for every local instance label and carried through all processing steps, preserving traceability across geometric operations. An example can be seen in 3, where labels A-D are uniquely identified across different local reference frames based on their associated map element ID.

The association with map elements is preserved for compound labels as well, which contain the IDs of the map elements they are composed of as well as the arc lengths of the individual labels in the compounded polyline. Centerline labels are associated to the respective lanelet instead.

Support for Varying Local Reference Frame Poses

lanelet2_ml_converter supports 3D reference frame poses composed of a 2D position on a flat plane and heading of the vehicle as the minimum required input. To generate the highest quality of local instance labels, up to full 6D reference frame poses can be used.

Real-Time Capable Implementation

To enable online inference of models that use map data as an input, not just as training labels, a real time capable implementation is necessary. The extension is therefore realized in a highly performant C++ implementation, averaging a single threaded performance of 3 ms for generating a set of local instance labels from a map, including conversions to learning representations, on a Intel i7-11800H processor.
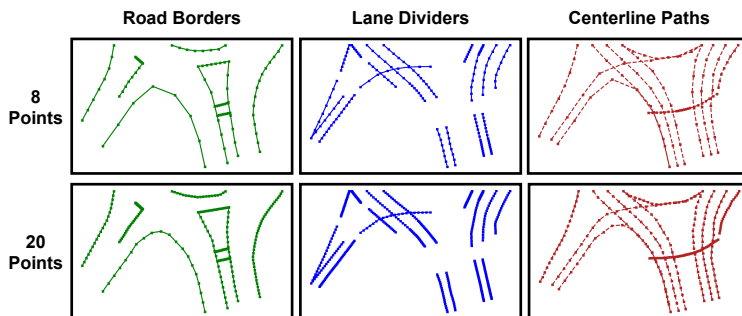
4   Application examples

Figure 4: Labels for road borders, lane dividers and lane centerline paths with different fixed point numbers similar to the labels in MapTR [22] and MapTRV2 [21].
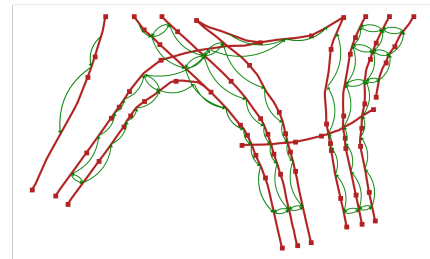
Figure 5: Labels for lanelet centerlines with connectivity similar to OpenLaneV2 [19].

The map instance labels supplied from the extension can be used to supervise a wide variety of map perception training tasks. Besides perception, other map-based learning tasks using map data as input such as trajectory prediction can be fed by this work. In the following we demonstrate the flexibility of the resulting local label instances with three popular application examples from the field of map perception.

MapTR: Online HD Map Construction

One of the key features of the recent MapTR architecture for vectorized map perception is the representation of map elements as compounded linestrings with a fixed point number. This is natively supported in the extension and all element types can be separately used and easily resampled with a chosen fixed point number. Figure 4 shows the default element types with both a 8-point and 20-point representation.

To validate the effectiveness of the extension and show examples for the newly possible training tasks, we also apply it to the Argoverse 2 dataset [17] to train a model with the MapTRV2 [21] architecture. The available maps in Argoverse 2 are converted to a temporary rudimentary Lanelet2 map during the label generation to be compatible with lanelet2_ml_converter. Contrary to other available methods, this allows us to easily separate the commonly used single divider class into dashed and solid lane dividers as well as fix some bugs in the label generation process. This distinction is missing from currently used training labels, but crucial for prediction and planning in autonomous vehicles.

Table 1 shows the resulting metrics in comparison with the baseline without distinct divider types. Even though the training task is substantially more difficult, performance

Labels	Epoch	AP$_{div}$	AP$_{dsh}$	AP$_{sol}$	AP$_{ped}$	AP$_{bou}$	AP$_{cen}$	mAP
No Divider Types	18	72.1	-	-	60.0	68.9	66.0	**66.7**
Dashed / Solid	18	-	61.5	68.8	60.4	68.1	66.6	**65.1**

Table 1: Performance Metrics of MapTRV2 [21] on Argoverse 2 [17] depending on the chosen divider label representation

is still comparable, with the AP for pedestrian crossings and centerlines even slightly increasing. This further strengthens the suggestion of the proposed labels being a better representation of reality. Fig. 6 shows an example of the different labels for a scene, changing the representation in a large way for downstream prediction and planning tasks. An example of the previously mentioned bugs in the original labels can also be observed implicitly from the output in fig. 6b, the solid divider in the middle of the road, present in the new labels in fig. 6d, is missing here.

(a) Input images with dashed / solid predictions

(b) Standard pred.

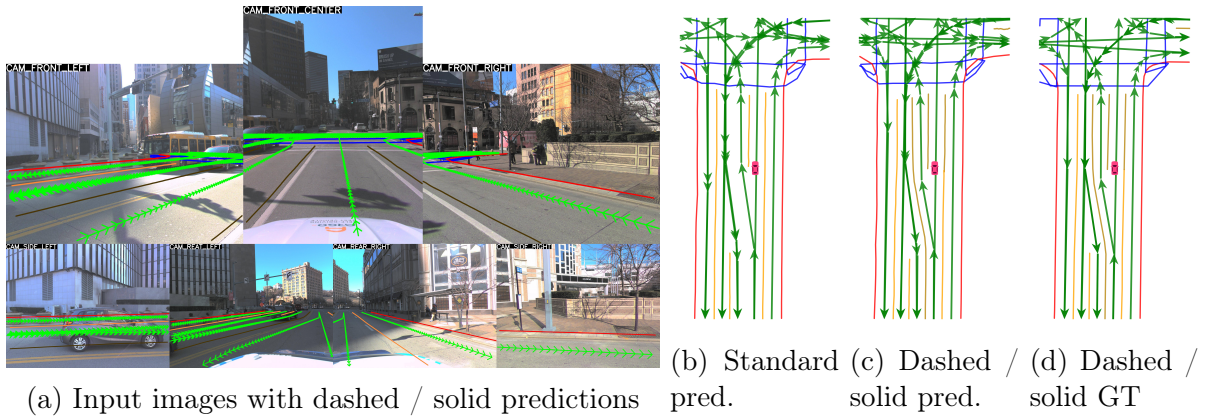(c) Dashed / solid pred.

(d) Dashed / solid GT

Figure 6: MapTRV2 [21] prediction sample for the standard and dashed / solid divider labels. Red denotes road borders, blue pedestrian crossings and green centerlines. Yellow denotes either dashed lane dividers or the standard single divider type and dark yellow solid dividers.

OpenLaneV2: Topology Inference

Another important area in map perception is scene topology inference, the inference of higher level information such as lane connectivity and traffic element to lane association. Topology inference has seen a large boost in research activity from the OpenLaneV2 dataset [19], a new large-scale dataset that explicitly includes higher level labels of this kind. The label representation used in OpenLaneV2 can easily be mirrored from the label instances in the extension, with both lanelet centerlines and their connectivity edges already part of the label data. A visualization of the formed graph can be seen in figure 5.

Online Map Inference and Fusion

The full traceability of all labels to their underlying map elements and the real time capability of the extension enable another application so far unsupported by existing

map data converters: Online map fusion of incomplete maps or updating existing map elements[26]. The Argoverse 2 dataset defines a similar task with the trust-but-verify subset [15] and this extension bridges the gap towards applicability in a real system using a unified, standardized HD map framework for all parts of an automated driving stack.

5 Conclusions and Future Work

This work describes a extension to the Lanelet2 framework that enables data generation for online map perception in automated driving. The current state of HD map formats and frameworks is reviewed and based on new developments in HD maps and map perception, the need for a unified HD map format and framework that can enable both fully map based driving and map perception tasks is motivated. Our proposed requirements for such a framework are described together with limitations of currently available alternatives. We detail the implementation of these requirements by this Lanelet2 extension and finally show application examples for recent popular map perception tasks. Future work could include improved support for non-road features such as traffic lights and traffic signs.

Acknowledgements

References

[1] TomTom International BV. (2024). ASAM OpenDRIVE 1.8.0, Static Road Network Description. Accessed on: 2024-/01-21, [Online]. Available: https://www.tomtom.com/products/hd-map/.

[2] HERE. (2024). Maps for ADAS and HAD. Accessed on: 2024-/01-21, [Online]. Available: https://www.here.com/platform/adas-had.

[3] F. Poggenhans, J. H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018, ISBN: 9781728103235.

[4] ASAM e.V. (2023). HD Map: Precise HD maps for safe autonomous driving. Accessed on: 2024-/01-21, [Online]. Available: https://publications.pages.asam.net/standards/ASAM_OpenDRIVE/ASAM_OpenDRIVE_Specification/latest/specification/index.html.

[5] J.-H. Pauls, K. Petek, F. Poggenhans, and C. Stiller, "Monocular localization in hd maps by combining semantic segmentation and distance transform," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4595–4601.

[6] Y. Xu, W. Shao, J. Li, K. Yang, W. Wang, H. Huang, *et al.*, *Sind: A drone dataset at signalized intersection in china*, 2022. arXiv: 2209.02297 [cs.CV].

[7] M. Szántó, C. Hidalgo, L. González, J. P. Rastelli, E. Asua, and L. Vajta, "Trajectory planning of automated vehicles using real-time map updates," *IEEE Access*, vol. 11, pp. 67 468–67 481, 2023.

[8] D. Majstorovic and F. Diermeyer, *Dynamic collaborative path planning for remote assistance of highly-automated vehicles*, 2023. arXiv: 2308.15167 [cs.RO].

[9] 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 *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.

[10] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.

[11] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[12] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 958–964.

[13] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, eabm6074, 2022.

[14] Autoware Foundation. (2023). Autoware documentation. Accessed on: 2024-/01-21, [Online]. Available: `https://autowarefoundation.github.io/autoware-documentation/main/`.

[15] J. Lambert and J. Hays, "Trust, but verify: Cross-modality fusion for hd map change detection," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., vol. 1, Curran, 2021.

[16] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[17] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, *et al.*, *Argoverse 2: Next generation datasets for self-driving perception and forecasting*, 2023. arXiv: `2301.00493 [cs.CV]`.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, *et al.*, "Nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[19] H. Wang, T. Li, Y. Li, L. Chen, C. Sima, Z. Liu, *et al.*, "Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[20] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[21] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, *et al.*, "Maptrv2: An end-to-end framework for online vectorized hd map construction," *arXiv preprint arXiv:2308.05736*, 2023.

[22] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, *et al.*, "Maptr: Structured modeling and learning for online vectorized hd map construction," in *The Eleventh International Conference on Learning Representations*, 2022.

[23] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "VectorMapNet: End-to-end vectorized HD map learning," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 23–29 Jul 2023, pp. 22 352–22 369.

[24] F. Poggenhans and J. Janosovits, "Pathfinding and routing for automated driving in the lanelet2 map framework," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, 2020.

[25] B. Liao, S. Chen, B. Jiang, T. Cheng, Q. Zhang, W. Liu, *et al.*, "Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction," *arXiv preprint arXiv:2303.08815*, 2023.

[26] R. Sun, L. Yang, D. Lingrand, and F. Precioso, "Mind the map! Accounting for existing map information when estimating online HDMaps from sensor data," arXiv, Tech. Rep., Nov. 2023, arXiv:2311.10517 [cs] type: article.