2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Enhancing Online Road Network Perception and Reasoning with Standard Definition Maps

Hengyuan Zhang*[1], David Paz*[2], Yuliang Guo[2], Arun Das[2], Xinyu Huang[2], Karsten Haug[3], Henrik I. Christensen[1] and Liu Ren[2]

***Abstract*— Autonomous driving for urban and highway driving applications often requires High Definition (HD) maps to generate a navigation plan. Nevertheless, various challenges arise when generating and maintaining HD maps at scale. While recent online mapping methods have started to emerge, their performance especially for longer ranges is limited by heavy occlusion in dynamic environments. With these considerations in mind, our work focuses on leveraging lightweight and scalable priors–Standard Definition (SD) maps–in the development of online vectorized HD map representations. We first examine the integration of prototypical rasterized SD map representations into various online mapping architectures. Furthermore, to identify lightweight strategies, we extend the OpenLane-V2 dataset with OpenStreetMaps and evaluate the benefits of graphical SD map representations. A key finding from designing SD map integration components is that SD map encoders are model agnostic and can be quickly adapted to new architectures that utilize bird's eye view (BEV) encoders. Our results show that making use of SD maps as priors for the online mapping task can significantly speed up convergence and boost the performance of the online centerline perception task by 30% (mAP). Furthermore, we show that the introduction of the SD maps leads to a reduction of the number of parameters in the perception and reasoning task by leveraging SD map graphs while improving the overall performance. Project Page: https://henryzhangzhy.github.io/sdhdmap/.**

I. Introduction

Research and development in the areas of autonomous driving has rapidly evolved over the past few decades. Nevertheless, highly detailed and rich maps, often referred to as High Definition (HD) maps, have become a key component required by most perception and planning modules, such as in Autoware [1] and Apollo [2]. It is especially critical for the fully autonomous driving scenarios, when online perception of individual lanes [3], [4] cannot satisfy the need of complex motion planning or long-term navigation. Today, HD maps serve as the de facto modality and provide static contextual information for behavior prediction [5], [6], [7] and road topology and connectivity information for route planning and motion planning [8], [9].

Fig. 1. Online road network perception and reasoning is challenging due to occlusion by on-road objects, especially at long-range as required by planning. In this example, the left turn map elements are heavily occluded by the vehicles. The baseline (TopoNet) using only image data misses the left turn while our method (TopoNet+OSMR–leveraging rasterized Standard Definition (SD) maps as the prior) predicts it correctly. Visualizations represent centerlines with connectivity information.

However, HD maps present significant challenges in terms of cost, scalability, and maintenance [10]. HD maps often require dedicated mapping teams to gather, process, and label data for the regions within the operational design domain. If a significant change occurs that displaces the original definitions, the original map must be updated which presents high maintenance overhead and potential failure points in the software. This overall process often requires human supervision, manual labeling, and extensive verification. As a result, these considerations present a bottleneck in terms of providing a scalable and cost-efficient solution.

Therefore, cost-effective and online methods present benefits that can potentially address the pain points of HD maps. More recently a research effort has focused on the online component [11], [12], [13], [14], which appears to advance

* These authors contribute equally to this work.
[1]Hengyuan Zhang and Henrik I. Christensen are with the Contextual Robotics Institute, UC San Diego, 9500 Gilman Drive, La Jolla, CA 92122, USA {hyzhang, hichristensen}@ucsd.edu.
[2]David Paz, Yuliang Guo, Arun Das, Xinyu Huang and Liu Ren are with Bosch North America and Bosch Center for AI (BCAI), 384 Santa Trinita Ave, Sunnyvale, CA 94085, USA {david.pazruiz, yuliang.guo2, arun.das, xinyu.huang, liu.ren}@us.bosch.com.
[3]Karsten Haug is with Robert Bosch GmbH, Hessbruehlstrasse 21, Stuttgart-Baihingen Bade-Wuerttemberg 70565, Germany karsten.haug@de.bosch.com.

the classic lane detection by recovering the topological outputs as close to the HD maps as possible. For instance, MapTR [13] introduces a method for online vectorized HD map generation; this work focuses on the perception of road boundaries, pedestrian crossings, and lane marks. As an extension to the perception task, Li et al. [15] introduce a method termed TopoNet to model the underlying topology and road network connectivity for urban driving tasks. This approach focuses on the perception and reasoning task jointly and aims to reduce the gap between the features generated by fully online models and HD maps. Nevertheless, these methods present high computing requirements and still experience challenges in highly dynamic and occluded environments, especially for longer ranges which is required for planning.

To further explore online HD mapping for autonomous driving applications, our work seeks to incorporate lightweight priors as part of the formulation to improve performance while reducing computing complexity. More specifically, we explore the benefits of utilizing coarse priors in the form of the widely available Standard Definition (SD) maps, such as Google Maps and OpenStreetMaps (OSM) [16]. We evaluate the performance and computing implications of SD map representations and encoders for the detection and reasoning tasks. We perform our experiments using various architectures across open-source datasets and additionally introduce a new SD map dataset based on OSM to explore the benefits of graph-based SD map representations. In summary, our key contributions are as follows.

- We introduce different types of lightweight SD map representations into the online mapping tasks. We show that SD maps provide long-range prior information and can visually improve occluded regions; thus, resulting in better overall quantitative results.
- We investigate prototypical representations of the SD maps and the most effective integration with online map baselines with various architectures.
- Additionally, we expand the OpenLane-V2 dataset with OSM data to enable using SD maps as graph representations for graph-based architectures. Our dataset, termed OpenLane-V2-OSM, will be public.

II. Related Work

High Definition (HD) Maps. HD maps are costly to generate and require constant maintenance [10], [17]. To tackle these problems, two major directions emerge. One direction, HD map automation [18], [19], [20], investigates automated HD map generation and map change detection and merging with a fleet. The other direction involves online HD mapping [11], [12], [13], [14], [15], which entails building an HD map on the fly.

For HD map automation, the aspect more related to our work is online map generation. Early work focuses on drivable area or lane semantics [17] that are important for navigation. Related methods introduce BEV-level scene understanding strategies using monocular camera data [18], [19]. Zhou et al. [20] use instance segmentation for lane segmentation and a particle filter to extract lane information. The final vector map is then generated with the information from the OpenStreetMap (OSM) [16].

HD map automation still requires a large fleet to constantly maintain them. To address this issue, online HD mapping has become more popular in recent years. STSU [21] builds BEV centerline road networks from a single camera. HDMapNet [11] takes surrounding view images and first generates semantic segmentation, then post-processes them to construct vectorized maps. This post-processing step is removed in VectorMapNet [12] by directly generating vectorized representations using a transformer decoder. Furthermore, MapTR [13] and MapTRv2 [14] propose a permutation invariant loss for better learning map elements that are not directional. These works typically focus on the $[-30m, 30m]$ range and the performance deteriorates significantly when the range increases.

To make HD maps more useful for downstream tasks such as prediction and planning, they need more than simple map elements. The list extends to traffic elements such as traffic lights/signs and a stronger association component that reasons about their relationship. Building on top of prior large datasets with HD maps such as Nuscenes [22] and Argoverse [23], OpenLane-V2 [24] provides additional annotations including traffic light colors, turn signals, and their association to specific lanes. TopoNet [15] uses this dataset to decode centerlines from BEV features, and applies a Scene Graph Neural Network (SGNN) to learn the final centerline and control relationships.

Standard Definition (SD) Maps. SD maps such as Google Maps and Open Street Maps include high-level road network information without lane-level information. They are scalable and lightweight solutions widely used in human driving that provide context for driving; these qualities can help address some of the issues in real-time perception such as occlusion. However, few methods use SD maps as prior for detailed mapping. In [20], they infer the connectivity of estimated lanes in intersections based on the connectivity of OSM. OSM are also used in [25] and [26] towards e2e autonomous driving. Various works also use SD maps as context for downstream tasks such as prediction [27] and planning [28]. Inspired by these strategies, we explored using OSM as context for online HD mapping.

III. Methodology

To evaluate the effectiveness of introducing SD maps as a prior for online HD mapping, we integrate SD maps into recent online mapping tasks. These tasks can be divided into two folds. One is the perception task, focusing on the map element such as lane line, road boundary, crosswalk and centerline prediction [13], [14]. The other adds reasoning to perception, which not only detects the map elements but also traffic elements such as traffic lights/signs and their relations [15]. We describe our approaches to integrating SD maps into state-of-the-art models for the pure perception task in Section III-A and the joint perception and reasoning tasks in Section III-B.

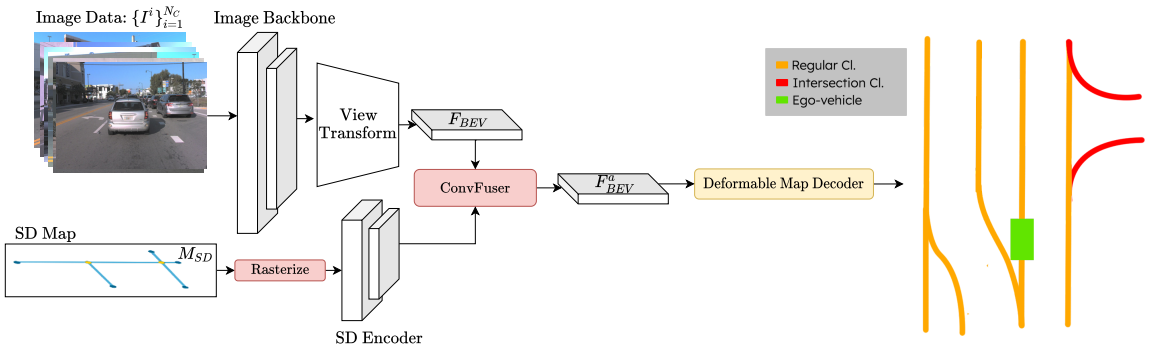


Fig. 2. Our pipeline integrating a rasterized SD map with the state-of-the-art online perception approach MapTR. The model encodes the SD Map in rasterized features in bird's-eye view (BEV) space, and fuse them with image BEV features, and predicts centerlines with a deformable attention decoder.

A. Perception Task

For the perception task, we predict centerlines based on surround-view images and SD maps. More formally, given a sequence of image inputs $\{I^i\}_{i=1}^{N_C}$ generated from N_C surround-view cameras, and an ego-centric SD map representation M_{SD}, the task predicts the centerlines D_C. A centerline predicted $D_C^k \in D_C$ is represented as a 2D line $\{(x_j^k, y_j^k)\}_{j=1}^{N_L}$ with N_L waypoints in the ego-vehicle frame. Additionally, each centerline prediction contains an attribute to denote if a centerline is an intersection segment or a regular segment.

Architecture. We incorporate SD maps into the state-of-the-art online HD mapping architecture from MapTR [13] which is based on an encoder-decoder architecture (Fig. 2). An image encoder encodes surround-view images $\{I^i\}_{i=1}^{N_C}$ into perspective-view (PV) features F_{PV}. These PV features are then transformed into unified BEV features F_{BEV} by a BEV view transform module. The BEV features are further decoded into map elements D_C.

Rasterized SD Map. Given that SD maps are naturally represented in BEV, we fuse SD map features with BEV image features. We rasterize the SD map and generate a BEV SD map M_{SD}, each SD map class represented by a distinct color. An SD encoder, in this case a ResNet-18 [29], is employed to extract SD map features F_{SD}. We chose a lightweight encoder given that the features are already color-coded by semantics.

SD map feature F_{SD} is then interpolated and concatenated with the BEV feature from images F_{BEV} along the channel dimension. Our design leverages this approach to align spatial features from BEV and SD maps together. The intuition revolves around using the SD map canvas to reduce the centerline search space in BEV. Subsequently, similar to fusing with BEV LiDAR feature in [12], [13], the ConvFuser with a simple two-layer convolutional neural network fuses the concatenated feature and output the fused BEV feature F^a_{BEV}.

Losses. The losses are the same as introduced in [13], a combination of classification loss $\mathscr{L}_{cls}$, point distance loss $\mathscr{L}_{p2p}$ and edge directional loss $\mathscr{L}_{dir}$.

B. Perception and Reasoning Task

This section builds on the task explored in Section III-A by incorporating a reasoning component to road network perception. In addition to identifying the centerline elements nearby, with respect to the ego-agent, this task seeks to identify the traffic elements on the road, the relationships between centerlines, and the relationships between the traffic elements and centerlines detected. Traffic elements, such as traffic lights, road markings, and road signs, provide important navigation information to HD maps.

More formally, given the same inputs $\{I^i\}_{i=1}^{N_C}$ and M_{SD}, the task involves predicting the centerlines D_C, the road-side traffic elements D_T, and the relational attributes, the association matrix between centerlines and centerlines A_{CC}, and the association matrix between centerlines and traffic elements A_{CT} [24]. From this these outputs, a centerline predicted $D_C^i \in D_C$ is represented as a 3D line $\{(x^i, y^i, z^i)\}_{i=1}^{N_L}$ with N_L waypoints in the ego-vehicle frame. A traffic element j detected in image i ($D_T^{i,j}$) can be represented as a 2D bounding box $(x, y, w, h, class)$, where x, y denote position, w, h bounding box dimensions, and $class$ denotes the traffic sign attribute such as $turn\_left, turn\_right, red, green, yellow$ for traffic lights.

Architecture. We employ the TopoNet Architecture [15] as the basis of our approach. The approach outlined in Fig. 3 utilizes an image backbone (i.e. ResNet-50) to process N_C image inputs and generate their corresponding images features F_{PV}. These perspective view features are then utilized in multiple downstream components. First, a deformable attention decoder [30] is used as a traffic element decoder where the traffic element queries Q_T attend to the perspective view features F_{PV} to decode traffic elements embeddings. Similarly, the image features F_{PV} are processed by a BEVFormer Encoder [31] to transform the perspective view features into BEV features–this is denoted by F_{BEV}. In the following centerline deformable attention decoder, the centerline queries denoted by Q_C then attend to the BEV features to generate centerline embeddings.

A key difference with respect to TopoNet involves the added SD map feature encoder. In this section, we experiment with two encoders: one that processes SD maps as

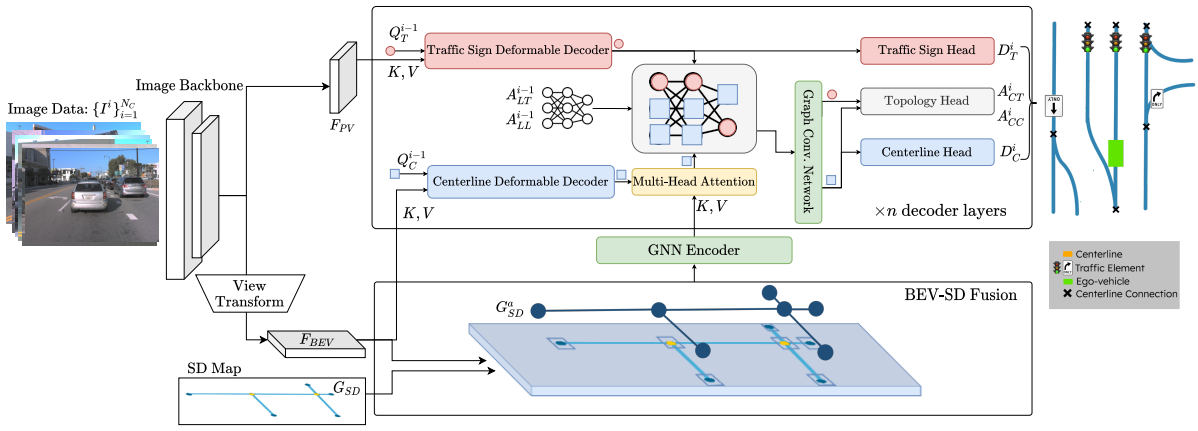


Fig. 3. Our pipeline integrating graph-based SD maps with the state-of-the-art perception and reasoning architecture based on TopoNet with a BEV-SD OSM graph encoder. The method processes multi-view image data, OSM SD map graphs, and leverages deformable decoders along with a Scene Graph Neural Network process to predict centerlines, traffic elements, and their relationships.

rasters and one that operates directly on SD map graphs. The raster-based encoder resizes the input SD map M_{SD} to the same dimensions as the BEV feature map F_{BEV} and stacks them together along the channel dimension; we utilize the encoder setup and rasterization process as introduced in Section III-A. In contrast, the graph-based encoder fuses SD map graphs with BEV features and is combined with the outputs from the centerline deformable decoder by leveraging a multi-head attention mechanism [32]. In the following parts of this section, we discuss the encoder and alignment process for SD map graphs, the perception heads, and reasoning process between centerlines and traffic signs. The section concludes with the losses used in the training process.

BEV-SD Graph Fusion. The BEV-SD fusion component shown in Fig. 3 leverages F_{BEV} to augment the node features from a given SD map graph $G_{SD} = (V_{SD}, E_{SD})$, where $V_{SD} = \{1, ..., n\}$ and $E_{SD} \subseteq V_{SD} \times V_{SD}$. More specifically, vertex i corresponds to node feature X_{SD}^i which contains positional information with respect to the ego-vehicle, namely $X_{SD}^i = (x_i, y_i)$. Since $F_{BEV} \in \mathbb{R}^{H_B \times W_B \times C_B}$ also encodes spatial BEV features in an ego-centric perspective with a fixed perception range given by an $H_B \times W_B$ grid, we can align a given SD node X_{SD}^i in BEV by scaling (x_i, y_i) as shown in Eq. (1) and Eq. (2), where H_B^m and W_B^m are convertion factors in terms of $cell/m$. Thus, the BEV feature corresponding to a given SD map element located at (x_i, y_i) can be indexed by (x_i^B, y_i^B). Similar to the rasterized encoder counterpart, our design is motivated by aiming to spatially align BEV and SD map features–which intuitively translates to using SD maps as a reference to regress centerlines and not start from scratch.

$$x_i^B = \lfloor x_i \cdot H_B^m \rfloor + \frac{H_B}{2} \tag{1}$$

$$y_i^B = \lfloor y_i \cdot W_B^m \rfloor + \frac{W_B}{2} \tag{2}$$

This node-level feature augmentation process is performed for every SD map node within the BEV perception range by concatenating the initial node feature and the BEV feature corresponding that that position as shown in Eq. (3). The augmented SD map graph is denoted by G_{SD}^a.

$$X_{SD}^{a,i} = \text{concat}\left(X_{SD}^i, F_{BEV}(x_i^B, y_i^B)\right) \tag{3}$$

SD Map Graph Encoder. To process the augmented SD map graph G_{SD}^a, we employ an Edge Convolution graph encoder as it has been shown to effectively capture local geometric attributes [33]. Our GNN formulation leverages the road connectivity information provided in the form of an adjacency matrix to determine node i's neighbors.

This GNN approach then utilizes a two-layer Multi-layer Perceptron (MLP) with a ReLU activation function to extract relevant features from an augmented representation, $\left[X_{SD}^{a,i}, X_{SD}^{a,j} - X_{SD}^{a,i}\right]$. The features are subsequently averaged across all $\mathcal{N}(i)$ neighbors as shown in Eq. (4).

$$X_{SD}^{a,i} = \frac{1}{\mathcal{N}(i)} \sum_{j \in \mathcal{N}(i)} \text{MLP}_\theta \left(\left[X_{SD}^{a,i}, X_{SD}^{a,j} - X_{SD}^{a,i} \right] \right) \tag{4}$$

After the graph propagation process, the SD map node features are attended by the output queries from the Centerline Deformable Decoder as denoted by the yellow block in Fig. 3.

A Scene Graph Convolutional Neural Network (SGNN) then takes the output from previous stages to capture the relational attributes among centerlines and between centerlines and traffic elements, as introduced in TopoNet [15].

Losses. We utilize the loss formulation from [15] to supervise the outputs generated at every decoder layer. There are two key components to the loss based on a Bipartite matching process [34] which includes a detection component and a reasoning loss component. The detection component uses an IOU loss, an L1 loss for bounding box regression and the Focal loss [35] for classification of traffic elements in perspective view. Similarly, for centerline detection we use the Focal loss and the L1 loss. Finally, the reasoning

component uses the Focal loss in the process of classifying correct relational assignments.

IV. Experiments

We perform extensive experiments to validate the effectiveness of SD maps in online HD mapping. We introduce the datasets and metrics in Section IV-A and Section IV-B. We subsequently present the experiments and results for the perception task in Section IV-C and for the perception and reasoning task in Section IV-D.

A. Datasets

OpenLane-V2. Our perception experiments are based on the OpenLane-V2 (OLV2) dataset (subset-A) [24], which is based on the Argoverse 2 dataset [23]. They provide egocentric SD maps along with seven surround-view camera images. SD maps from OpenLane-V2 include three classes, *road*, *crosswalk* and *sidewalk*. Each class is presented as a set of polylines. From the labels, only the centerline labels are used. They are represented as a set of points and are resampled to a fixed number of points N_L following MapTR [13]. In our experiments for the perception task, $N_L = 20$.

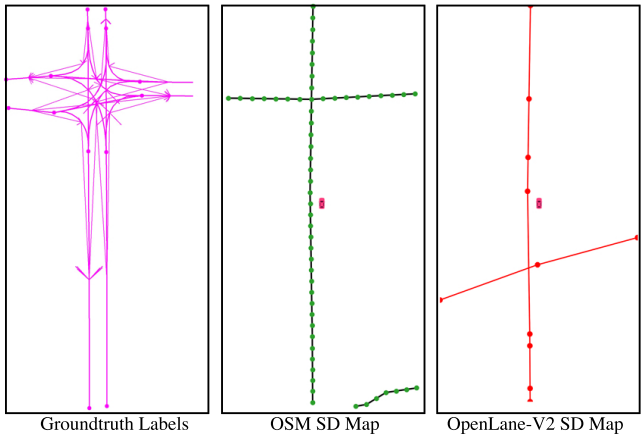


Fig. 4. Visual comparison between the groundtruth online maps, OSM SD maps, and OpenLane-V2 (OLV2) SD maps. OSM SD maps appear to be more consistent with the groundtruth.

OpenLane-V2-OSM. In our experiments, we enhance the OLV2 dataset with OSM data—adding 1,000 maps from Argoverse 2 with WGS84 conversions. Each map includes full OSM attributes, and a post-processing step creates egocentric SD graph representations for faster data loading. Both dataset representations will be publicly accessible.

OSM offers lightweight yet diverse contextual information for driving scenarios. Although not adequate for direct lane-level navigation, it provides road-level details through *way* and *node* elements with various attributes. Node elements describe point-level features like stop signs, while way elements cover a range from small road segments to pedestrian crosswalks, including attributes like category types, speed limits, and lane numbers if applicable. A visualization is shown in Fig. 4; where we observe groundtruth HD map labels, an OSM SD map, and an OLV2 SD map.

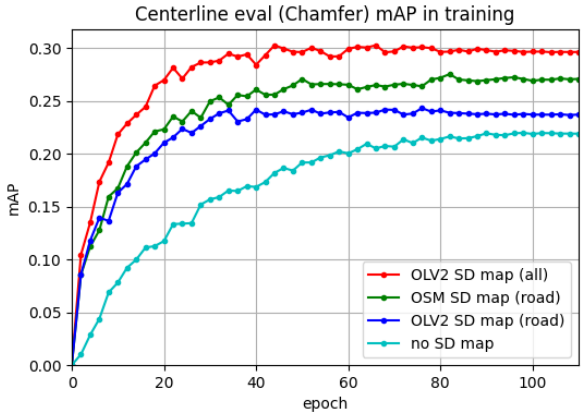


Fig. 5. Evaluation mAP during training with Chamfer distance. The model with an SD map converges much faster and achieves better performance.

B. Metrics

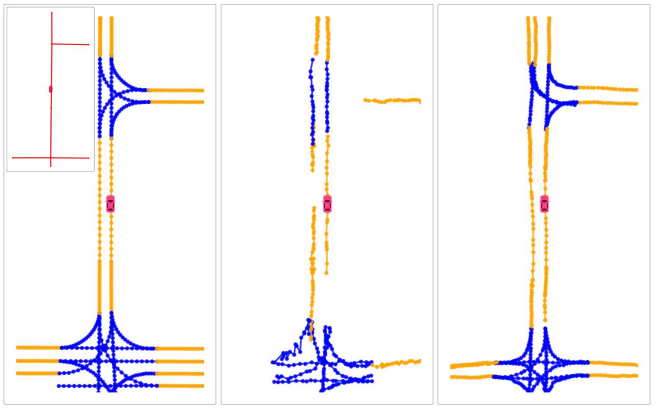
For the perception-only task, we follow [13] to evaluate the proposed architecture using Average Precision (AP) under Chamfer distance. The Chamfer distance gives the distance of two point sets as the average of the closest point distance. An association threshold determines whether a map element is considered a true positive. Three thresholds T_1=0.5m, T_2=1.0m, T_3=1.5m are used.

For the perception and reasoning task, we follow the metric from OLV2 [24]. The OLV2 dataset uses the OLV2 score OLS, which is an average of the 3D lane detection score DET_l, the traffic element recognition score DET_t, the topology score between centerlines TOP_{ll} and the topology score between centerlines and traffic elements TOP_{lt}.

C. Perception Results

We adapt MapTR to OLV2 dataset as the baseline. The perception range is increased from $[-30m, 30m]$ to $[-50m, 50m]$ range. The increase in perception range presents significant challenges for MapTR. For our approaches with rasterized SD maps, we experiment with three type of SD maps. OLV2 (*R*, *CW*, *SW*) has all three classes SD map features *road*, *crosswalk* and *sidewalk*, OLV2 (*R*) only maintains *road* features and OSM (*R*) has *road* features extracted from OSM. These models predict regular and intersection centerlines as a two-class problem. During evaluation, we measure the collective average performance between the two classes since separate benchmarks result in a negligible difference.

As shown in Fig. 5, integrating with rasterized SD map with all classes makes the training converge 10x faster: at epoch 10, the method that uses SD maps reaches similar performance to the model without SD maps but trains for 110 epochs. OLV2 (*R*, *CW*, *SW*) also presents 30% relative improvement in terms of mAP for the model trained for 110 epochs.



Groundtruth + SD map MapTR MapTR+OLV2 (R, CW, SW)

Fig. 6. Qualitative comparison of perception-only methods with and without SD maps. Blue color for intersection or connectors, orange color otherwise. Both models are trained for 25 epochs.

TABLE I

PERCEPTION-ONLY TASK RESULTS WITH RASTERIZED SD MAPS [KEY: R = ROAD, CW = CROSSWALK, SW = SIDEWALK]

	SD Type	Epoch	Chamfer Distance AP			
			mAP	T_1	T_2	T_3
MapTR	None	24	13.4	0.7	11.3	28.1
MapTR+OLV2	*R*		22.3	4.4	23.2	39.4
MapTR+OLV2	*R*, *CW*, *SW*		**27.1**	**7.0**	**28.8**	**45.5**
MapTR+OSMR	*OSM R*		23.0	5.1	24.2	39.8
MapTR	None	110	21.9	3.3	22.3	40.0
MapTR+OLV2	*R*		23.7	6.2	24.5	40.4
MapTR+OLV2	*R*, *CW*, *SW*		**29.6**	**10.8**	**31.4**	**46.6**
MapTR+OSMR	*OSM R*		27.1	9.4	28.6	43.2

As shown in Table I, among all three types of SD maps, we see consistent improvement from the baseline method. The OLV2 (*R*, *CW*, *SW*) with more SD map classes reaches the highest performance. For SD maps with only *road* features, OSM prior gives better results than OLV2 prior. This can potentially be caused by misalignment of the SD maps and groundtruth. We observe this in both SD map types and an example is shown in Fig. 4, where we observe missing SD map labels or undefined HD map features such as parking lot entrances that can cause inconsistencies.

In summary, these results suggest that SD maps provide useful priors for online HD mapping and speed up convergence and boost performance significantly. Qualitatively, we observe that our models with SD map priors perform better for far intersections, especially after the turning point where occlusion happens, as shown in Fig. 6. This is helpful for prediction and planning tasks to navigate intersections.

D. Perception and Reasoning Results

This section covers experiments conducted with rasterized and graph-based SD maps for the perception and reasoning task based on TopoNet [15]. The experiments include performance trade-offs from rasterized vs graph-based methods as TopoNet facilitates integration of not just raster based representations but also vector/graph-based representations.

As shown in Table II, leveraging SD map prior in either rasterized (OSMR) or graph representation (OSMG) leads to consistent improvements in the overall metrics. While other methods such as TopoMLP-YOLO [36] and MFV-ViT-L [37] focus on improving results with better traffic element detectors and larger backbones, our experiments show that SD map can benefit lane perception and reasoning without significant changes to the architecture. We consider these approaches contributing in orthogonal directions compared to ours, thus do not compete with each other. However, we hypothesize that by utilizing SD maps as priors, a performance boost can be observed for TopoMLP, MFV, and other BEV-based architectures.

TABLE II

PERCEPTION AND REASONING RESULTS WITH SD MAPS AND COMPARISONS WITH OTHER METHODS.

	Backbone	OLS	DET_l	DET_t	TOP_{ll}	TOP_{lt}
TopoNet	R50	34.8	28.4	45.0	4.2	20.7
TopoNet+OSMR	R50	**37.7**	**30.6**	44.6	**7.7**	**22.9**
TopoNet+OSMG	R50	36.7	30.0	**47.6**	5.4	21.3
TopoMLP	R50	38.2	28.3	50.0	7.2	22.8
TopoMLP-YOLO	R50	41.2	28.8	53.3	7.8	30.1
MFV-R50	R50	-	18.2	-	-	-
MFV-ViT-L	ViT-L	**53.2**	**35.3**	**79.9**	**23.0**	**33.3**

Impact of rasterized SD map features. To evaluate SD maps as rasters using the Toponet architecture, we replace the GNN SD map encoder introduced in Fig. 3 with the SD map encoder from the perception-only task covered in Section III-A. Furthermore, rather than utilizing the Multi-head Attention mechanism, we simply perform a feature augmentation to the BEV features F_{BEV} along the channel dimension.

TABLE III

PERCEPTION AND REASONING RESULTS USING RASTERIZED SD MAPS AS PRIORS. [KEY: R = ROAD, CW = CROSSWALK, SW = SIDEWALK]

	SD Type	Param	OLS	DET_l	DET_t	TOP_{ll}	TOP_{lt}	t (ms)
TopoNet	None	62.9M	34.8	28.4	45.0	4.15	20.7	**388**
TopoNet+OLV2	*R*	75.9M	34.9	26.2	47.3	4.55	20.1	407
TopoNet+OLV2	*R*, *CW*, *SW*	75.9M	36.1	27.9	**48.1**	5.14	20.9	407
TopoNet+OSMR	*OSM R*	75.9M	**37.7**	**30.6**	44.6	**7.71**	**22.9**	407

We perform an ablation with three different types of maps and features. Similar to Section IV-C. We train each one of the models for 24 epochs and evaluate on the OLV2 benchmark. The results shown in Table III provide an insight into the most relevant attribute types. For instance, even though we incorporate OLV2 SD maps in the first two experiments, the performance of centerline detection DET_l decreases while the traffic element detection metric DET_t indicates performance benefits. Since the OLV2 SD map road-level attributes are not available and the map generation process is not public, we hypothesize–as supported by Fig. 4–that the performance gaps in centerline detection scores originate from the misalignment between groundtruth HD maps and

the utilized SD maps. However, the performance benefits are observed in terms of traffic sign detection as crosswalks and sidewalks can provide contextual information on the location or existence of traffic elements. As an example, crosswalks are often located at intersections which are stop sign or traffic light protected road segments. Although the results may initially be counter intuitive given the experiments presented in Section IV-C (Table I), the TopoNet architecture (perception and reasoning) utilizes an SGNN component to capture relational attributes between centerline and traffic elements. These relationships are then taken into account when regressing centerlines and thus may be influencing their accuracy. We hypothesize that this underlying relationship between traffic elements and crosswalks/sidewalks may be helping boost performance in traffic element detection but also degrading centerline detection as centerlines are not specific to intersections only and can extend to regular road segments.

In contrast, the selected OpenStreetMap attributes are known and selected based on the urban driving operational design domain which aligns well with the OLV2 dataset. As a result, we observe improvements across all of the metrics introduced which include detection and reasoning scores. Lastly, we present qualitative results which portray the benefits of leveraging SD maps in occluded scenarios. In Fig. 1, we present side-by-side comparisons of the TopoNet baseline model and our approach that makes use of rasterized OSM. We observe better alignment of centerline features despite of severe occlusion.

Graph-based SD Maps. We evaluate the performance of graph-based SD map encoders using various backbones in Table IV. The baseline model (TopoNet-R50) utilizes a ResNet-50 [29] backbone. However, we additionally evaluate the performance using lighter backbones including ResNet-18 and ResNet-34. While Wu et al. [36] show that recent backbones [38], [39] can significantly increase the performance in the centerline detection task, our focus is on evaluating the trade-offs between performance and lighter weight methods such as SD map graphs.

In Table IV, the methods that utilize OSM SD map graphs are denoted by TopoNet-R**X**+OSMG, where **X** specifies different backbones. Evidently, as the number of parameters within a backbone increase, the performance across the different metrics also increases and is consistent across the corresponding TopoNet baselines. TopoNet-R34+OSMG is capable of boosting performance with respect to the larger Toponet-R50 while reducing the number of parameters and inference time. Furthermore, the larger TopoNet-R50+OSMG enables higher performance without significantly increasing the number of parameters or inference times. The overall results indicate that improvements can be achieved with lightweight methods with respect to the corresponding baselines. The inference times are measured on a Titan Xp GPU. The graph-based OSM encoder yields lower performance improvement than the rasterized OSM encoder but with significantly fewer parameters (+1.7M vs + 13M).

TABLE IV
PERCEPTION AND REASONING RESULTS WITH OSM SD MAP GRAPHS AND DIFFERENT BACKBONES

	Param	OLS	DET_l	DET_t	TOP_{ll}	TOP_{lt}	t (ms)
TopoNet-R18	49.8M	32.3	24.3	44.0	3.1	18.8	**349**
TopoNet-R18+OSMG	51.6M	33.2	26.9	42.9	3.86	18.9	365
TopoNet-R34	60.0M	33.3	26.2	43.8	3.63	19.6	361
TopoNet-R34+OSMG	61.7M	35.5	29.3	45.0	4.82	21.0	379
TopoNet-R50	62.9M	34.8	28.4	45.0	4.15	20.7	388
TopoNet-R50+OSMG	64.6M	**36.7**	**30.0**	**47.6**	**5.37**	**21.3**	407

Robustness to Localization Error. To demonstrate robustness to localization errors we conduct an experiment by injecting SD map noise to translation and rotation. For translational error, we add fixed magnitude noise along random directions at the scale of 0.25m, 0.5m, 1.0m, and 2.0m (quite high for lanes), and for rotational error, we experiment with noise at the scale of 0, 5, 10 degrees. Our results in Table V show that SD map localization errors with up to 1.0m in translation and 5 degrees in rotational error result in at most 5% performance degradation. This demonstrates exceptional robustness to localization errors that can significantly affect lane-level perception.

TABLE V
PERFORMANCE LOSS WITH VARIOUS LOCALIZATION ERROR

	Rotational Error		
Translational Error	0 Degrees	5 Degrees	10 Degrees
0.25m	0.45%	2.33%	13.64%
0.5m	0.90%	2.80%	14.18%
1.0m	3.07%	5.01%	15.95%
2.0m	10.82%	12.43%	21.14%

V. CONCLUSION

We incorporate SD maps into online HD mapping for both perception-only task and joint perception and reasoning tasks. We show that SD maps can make centerline perception models converge significantly faster and achieve better performance. Adding them in graph form can reduce the model size while improving performance. We additionally curated and made public a dataset based on OSM. The effectiveness of the proposed methods, especially for longer ranges and occluded scenes, contributes to addressing the current online mapping challenges and scalability constraints from autonomous driving HD maps. To create better online maps, future research is needed to address inaccuracies in SD maps and to produce more consistent structural representations.

REFERENCES

[1] Hatem Darweesh, Eijiro Takeuchi, and Kazuya Takeda. Openplanner 2.0: The portable open source planner for autonomous driving applications. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 313–318, 2021.

[2] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[3] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *European Conference on Computer Vision*, pages 666–681. Springer, 2020.

[4] Yeongmin Ko, Younkwan Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8949–8958, 2022.

[5] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation, 2020.

[6] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision (ECCV)*, 2020.

[7] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99. PMLR, 2020.

[8] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019.

[9] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 414–430. Springer, 2020.

[10] J. Jiao. Machine learning assisted high-definition map creation. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 01, pages 367–373, Tokyo, Japan, 23-27 July 2018.

[11] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. *arXiv preprint arXiv:2107.06307*, 2021.

[12] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023.

[13] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *International Conference on Learning Representations*, 2023.

[14] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *arXiv preprint arXiv:2308.05736*, 2023.

[15] Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, Junchi Yan, Ping Luo, and Hongyang Li. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

[16] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008.

[17] Hengyuan Zhang, Shashank Venkatramani, David Paz, Qinru Li, Hao Xiang, and Henrik I Christensen. Probabilistic semantic mapping for autonomous driving in urban environments. *Sensors*, 23(14):6504, 2023.

[18] Yigit Baran Can, Alexander Liniger, Ozan Unal, Danda Paudel, and Luc Van Gool. Understanding bird's-eye view of road semantics using an onboard camera. *IEEE Robotics and Automation Letters*, 7(2):3302–3309, 2022.

[19] Zhongyu Rao, Hai Wang, Long Chen, Yubo Lian, Yilin Zhong, Ze Liu, and Yingfeng Cai. Monocular road scene bird's eye view prediction via big kernel-size encoder and spatial-channel transform module. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7138–7148, 2023.

[20] Yiyang Zhou, Yuichi Takeda, Masayoshi Tomizuka, and Wei Zhan. Automatic construction of lane-level hd maps for urban scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6649–6656. IEEE, 2021.

[21] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15661–15670, 2021.

[22] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, Seattle, WA, USA, 13-19 June 2020.

[23] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. 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.

[24] Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Yuting Wang, Shengyin Jiang, Peijin Jia, Bangjun Wang, Feng Wen, Hang Xu, Ping Luo, Junchi Yan, Wei Zhang, and Hongyang Li. Openlane-v2: A topology reasoning benchmark for scene understanding in autonomous driving. *arXiv preprint arXiv:2304.10440*, 2023.

[25] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[26] Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. Variational end-to-end navigation and localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8958–8964, 2019.

[27] Jing-Yan Liao, Parth Doshi, Zihan Zhang, David Paz, and Henrik Christensen. Osm vs hd maps: Map representations for trajectory prediction, 2023.

[28] David Paz, Hengyuan Zhang, Hao Xiang, Andrew Liang, and Henrik I. Christensen. Conditional generative models for dynamic trajectory generation and urban driving. *Sensors*, 23(15):6764, Jul 2023.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[30] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.

[31] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[34] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[36] Dongming Wu, Jiahao Chang, Fan Jia, Yingfei Liu, Tiancai Wang, and Jianbing Shen. Topomlp: An simple yet strong pipeline for driving topology reasoning. *arXiv preprint*, 2023.

[37] Dongming Wu, Fan Jia, Jiahao Chang, Zhuoling Li, Jianjian Sun, Chunrui Han, Shuailin Li, Yingfei Liu, Zheng Ge, and Tiancai Wang. The 1st-place solution for cvpr 2023 openlane topology in autonomous driving challenge, 2023.

[38] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.