

MGMap: Mask-Guided Learning for Online Vectorized HD Map Construction

Xiaolu Liu¹, Song Wang¹, Wentong Li¹, Ruizi Yang¹, Junbo Chen^{2*}, Jianke Zhu^{1*}
¹Zhejiang University ²Udeer.ai

{xiaoluliu, songw, liwentong, ruiziyang, jkzhu}@zju.edu.cn, junbo@udeer.ai

Abstract

Currently, high-definition (HD) map construction leans towards a lightweight online generation tendency, which aims to preserve timely and reliable road scene information. However, map elements contain strong shape priors. Subtle and sparse annotations make current detection-based frameworks ambiguous in locating relevant feature scopes and cause the loss of detailed structures in prediction. To alleviate these problems, we propose MGMap, a mask-guided approach that effectively highlights the informative regions and achieves precise map element localization by introducing the learned masks. Specifically, MGMap employs learned masks based on the enhanced multi-scale BEV features from two perspectives. At the instance level, we propose the Mask-activated instance (MAI) decoder, which incorporates global instance and structural information into instance queries by the activation of instance masks. At the point level, a novel position-guided mask patch refinement (PG-MPR) module is designed to refine point locations from a finer-grained perspective, enabling the extraction of point-specific patch information. Compared to the baselines, our proposed MGMap achieves a notable improvement of around 10 mAP for different input modalities. Extensive experiments also demonstrate that our approach showcases strong robustness and generalization capabilities. Our code can be found at <https://github.com/xiaolul2/MGMap>.

1. Introduction

High-definition (HD) map plays an important role in autonomous driving [19], as it provides centimeter-level road information for self-positioning [18], path planning [7, 11] and other downstream tasks [14, 15, 17]. Typically, offline HD map constructions rely on manual annotations on global LiDAR point clouds, which usually require tedious financial and manual investments. Moreover, it is challenging for offline maps to reflect the ever-changing road conditions due to the lack of real-time updates. To tackle the above is-

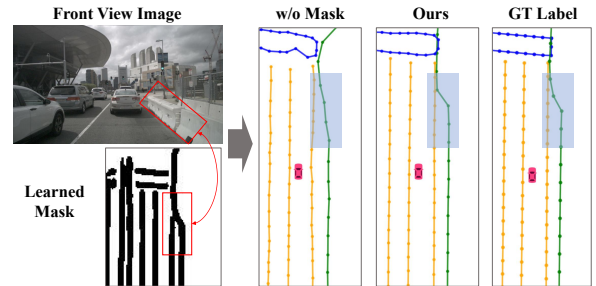


Figure 1. For some detailed structures, our proposed MGMap achieves effective map element localization by highlighting the informative regions through the learned masks.

sues, a lightweight and real-time online map construction paradigm [19] has gradually become a promising approach by incorporating information from onboard sensors.

Some existing approaches [10, 19, 36, 43, 50] consider online map construction as a segmentation task, where pixel-level rasterized maps are learned in bird’s-eye-view (BEV) space. Nevertheless, for vectorized expression, an additional step of post-processing is required to cluster and fit lane instances. Recently, efficient and straightforward approaches like VectorMapNet [26] and MapTR [23] have been proposed for vectorized map construction, in which map elements are represented by sparse point sets. Transformer-based architectures are directly employed to update instance queries and regress point locations.

Despite having achieved promising results, current map vectorization frameworks are still constrained by inherent issues. As shown in Figure 1, map elements, such as road edges, dividing lines, and pedestrian crossings, always have strong shape priors. Obscure features and coarse locations can easily lead to the loss of detailed expressions in prediction, especially for irregular boundaries and sudden changes in corner angles. Besides, subtle and sparse annotations pose significant challenges for deformable attention [51], which is known as a sparse and local feature extraction strategy employed in current vectorization frameworks. Considering the sparsity of detection targets, such under-sampling techniques can readily lead to coarse localization and the loss of effective information. These issues intensify the difficulties in determining the relevant feature scopes, confirm-

*Corresponding authors

ing lane line instances, and accurately locating lane points.

To tackle the above issues, in this paper, we propose a fine-grained approach called MGMap, which aims to improve localization and highlight specific features by incorporating the guidance of learned map masks. Initially, the enhanced multi-level BEV features are dynamically constructed for richer semantic and position features. Based on this, masks can be generated and adequately utilized at the holistic instance level and the more granular point level. At the instance level, we design the Mask-Activated Instance (MAI) decoder to guide the construction and feature aggregation of lane queries. By leveraging learned instance masks, MAI decoder enables the activation of lane queries to possess global instance structures and shape characteristics upon activation. Furthermore, at the point level, a Position-Guided Mask Patch Refinement (PG-MPR) module is proposed to alleviate the difficulty in locating relevant features due to the sparse and irregular detection targets. By focusing on the specific patch regions, binary mask features can be extracted to gather more detailed information at lanes' surrounding locations, which is well-designed for finer regression of the detailed structure and point locations.

Extensive experiments on *nuScenes* [1] and *Argoverse2* [46] datasets demonstrate that MGMap achieves state-of-the-art (SOTA) performance on the task of online HD map construction. Besides, the promising experiment results under different settings show the robustness and generalization capability of our presented model. The main contributions of our work can be summarized as follows:

- An effective approach for precise online HD map vectorization with the guidance of learned masks. The effective features of instance masks and binary masks are extracted for unique lane lines and shape learning.
- A mask-activated instance decoder and a novel position-guided mask patch refinement module to decode map elements from the instance level and the point level by fully leveraging the potential of mask features.
- Promising results on two testbeds show that our MGMap outperforms the previous approaches at a large margin and has strong robustness and generalization capability.

2. Related Work

Online HDMap Construction. Unlike traditional offline HD map annotations on global LiDAR points [37, 38], recent studies [19, 23, 26, 43] explore the online construction directly from on-board sensor data to reduce the cost of labeling and provide up-to-date road information. Some approaches [10, 19, 20, 33, 43] treat HD map construction as a segmentation task to predict pixel-level rasterized maps, which requires post-processing for vectorized construction. For more straightforward vectorized construction, Liu *et al.* [26] propose a two-stage framework VectorMapNet with an auto-regressive decoder to recurrently connect vertices.

Jeong *et al.* [39] detect points first and employ an adjacent matrix in InstaGraM to build the connections among instance points. Further, Liao *et al.* [23, 24] propose to represent map elements as the ordered points with fixed numbers in MapTR, so that transformer architecture can be used to regress points' positions simultaneously. Later, Qiao *et al.* [32] and Ding *et al.* [9] present new modeling strategies, in which BeMapNet and PivotMap utilize Bezier curves and dynamic pivot points to model map elements separately for more detailed representations. In contrast to the above methods, our MGMap approach introduces the guidance of generated mask features to handle lane shapes in detail with specific feature enhancement.

Camera-based BEV Perception. HD map construction relies on high-quality BEV features, which are also the basis for most 3D perception tasks [16, 25, 31, 44, 45]. Generally, BEV features are extracted and transformed from perspective-view (PV) images. Initially, Reiher *et al.* [35] utilize homography transformation to project PV images into BEV space in IPM. After that, learning-based approaches are widely used to construct more reliable BEV features [2, 44]. Pan *et al.* [29] employ a fully connected layer in VPN to convert perspective view features into BEV space. In [31] and [16], depth estimation is utilized to establish the connection between the surrounding view images and BEV features. In order to enhance the robustness of the model and the effect of detection, multi-modality and temporal fusion strategies are used in BEVFormer [20] and BEVFusion [22, 27]. Besides, transformer-based architectures [20, 25, 30] are widely used for feature aggregation. BEV features are represented by queries at different positions, which can be updated by interaction with PV image features. To obtain a more reliable HD map, we employ a pyramid-like network for multi-scale BEV features, which enables to capture the rich semantic and location information for online HD map construction.

Mask Refinement for Segmentations. Mask refinement strategies are widely used for different segmentation tasks, aiming to improve the quality of instance or semantic features. In previous work [5] and [40], boundary-aware mask features are constructed by an extra branch, which is used to improve mask localization accuracy. Cheng *et al.* [6] utilize the interaction between learned mask features with instance activation maps for representing objects with instance features. In [3, 4, 8, 21], mask features are integrated into the transformer-based architecture, in which attentions are utilized for feature extraction. Based on the initial prediction outputs, Tang *et al.* [42] propose a refinement strategy with small boundary patches to improve the mask quality. As for map construction, our proposed MGMap approach takes advantage of the learned mask features from both the instance level and point level to highlight and enhance informative regions of subtle map annotations.

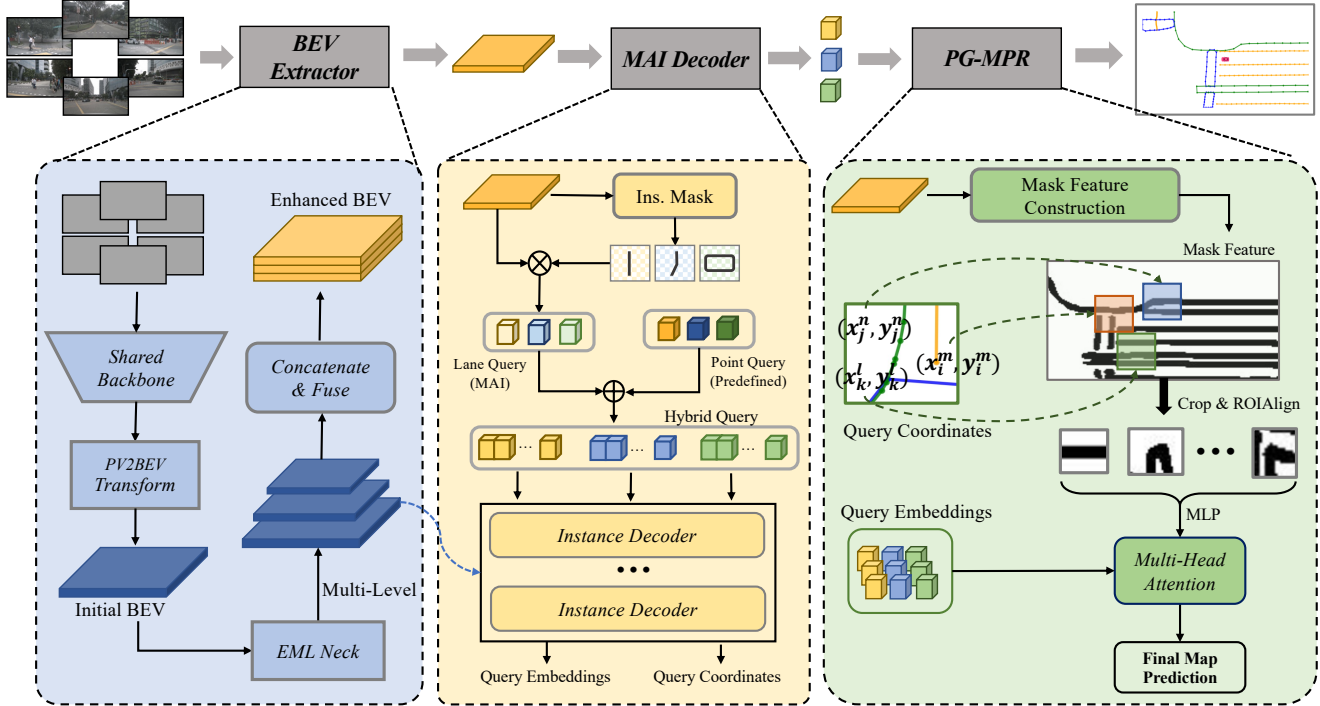


Figure 2. **Overview of MGMap framework.** MGMap mainly consists of three components: (1) BEV Extractor to obtain multi-scale BEV features by transforming from perspective view (PV) to BEV with the enhanced multi-level neck; (2) Mask-Activated Instance (MAI) Decoder is employed to construct and update queries at instance level; (3) Position-Guided Mask Patch Refinement (PG-MPR) module is designed to refine points’ positions from local patch features at point level.

3. MGMap

Given surround-view images captured from onboard cameras, our goal is to distinguish local BEV map instances while locating their corresponding structures. Each map element comprises a class label c and an ordered sequence of points $\mathbf{P} = \{(x_i, y_i)\}_{i=1}^N$ that represents the lane structure, with N denoting the number of points for each lane.

Figure 2 illustrates the framework of our proposed MGMap. To obtain multi-scale BEV features, a pyramid network with fused attention is designed, which is built after initial PV-to-BEV feature transformations (Section 3.1). Subsequently, a cascaded transformer decoder is utilized to update mask-activated instance queries (Section 3.2). Finally, by interacting with the local semantic context derived from the patch of mask features, we achieve fine-grained position refinement at the point level (Section 3.3).

3.1. BEV Feature Extraction

Initially, we employ shared CNN backbones to extract 2D features from PV images. PV features can be gathered into a unified BEV representation via the strategy in [20], which utilizes deformable attention [51] to update BEV queries by interacting with surround-view image features. The extracted BEV feature can be represented as $\mathbf{F}_0 \in \mathbb{R}^{D \times H_{BEV} \times W_{BEV}}$, where $H_{BEV} \times W_{BEV}$ is the size of

the BEV feature and D represents the dimension.

Enhanced Multi-Level Neck. To obtain BEV features with rich semantic and location information, at BEV space, we design a 3-layer Enhanced Multi-Level (EML) neck with fused attention to construct the unified BEV features. Multi-scale BEV features with larger receptive fields can be obtained for a better understanding of the overall structures.

In the EML neck, we construct the cascaded layers using residual blocks based on the hybrid of channel attention (CA) and spatial attention (SA) [47], in which hybrid multiplications among channels and spaces are designed to capture both local and global contextual information. Such dynamic selections at different BEV spaces are beneficial to detect lane lines with irregular shapes. The calculation of learnable attention maps can be formulated below

$$\mathbf{F}_{i+1} = (\text{CA}(\mathbf{F}_i) \times \mathbf{F}_i) \times \text{SA}(\mathbf{F}_i). \quad (1)$$

After that, multi-level BEV features $\{\mathbf{F}_i\}_{i=1}^3$ can be obtained with the shape of $D_i \times \frac{H_{BEV}}{2^{i+1}} \times \frac{W_{BEV}}{2^{i+1}}$ for each layer i , where D_i is the corresponding dimension. To preserve all the information at different levels, we employ bilinear interpolation to upsample 3-level outputs $\{\mathbf{F}_i\}_{i=1}^3$. All these features are aligned to have the same resolution as the initial \mathbf{F}_0 . Finally, a 3×3 convolutional layer after concatenation is used to aggregate multi-level features and obtain the enhanced BEV features F_c . Such enhanced F_c contains local

and semantic information across varying receptive fields, enhancing the capability to detect fine-grained structures.

3.2. Mask-Activated Instance Decoder

For each lane instance, specific query embeddings with instance and structure information are required for the regression of lane shapes and positions. Based on enriched BEV features, this section focuses on the design of mask-activated lane queries and the subsequent update through the cascaded deformable transformer decoder.

Mask-Activated Query. To achieve a more detailed and specific representation, MGMap employs a hybrid approach that combines lane queries \mathbf{Q}_{lane} and point queries \mathbf{Q}_{point} to encode individual map instances.

For lane queries, in contrast to the fixed query designs [23, 51], $\mathbf{Q}_{lane} \in \mathbb{R}^{M \times D}$ is dynamically initialized with the guidance of learned instance masks, which is more flexible and contains specific shape prior and instance information. Here M is the number of instance queries and D is the dimension. Specifically, we obtain a set of instance segmentation maps $\mathbf{M}_{ins} \in \mathbb{R}^{M \times H_{BEV} \times W_{BEV}}$ by applying basic convolution with the sigmoid function $\sigma(\cdot)$ to the enhanced BEV feature \mathbf{F}_c , as illustrated in the lower half of Figure 3. By leveraging \mathbf{M}_{ins} , we generate the mask-activated instance query \mathbf{Q}_{lane} via multiplication between instance masks M_{ins} and the transpose of enhanced BEV features

$$\mathbf{Q}_{lane} = (\sigma(\text{Conv}(\mathbf{F}_c))) \times \mathbf{F}_c^\top. \quad (2)$$

During the training stage, \mathbf{M}_{ins} is supervised by instance mask annotations, and bipartite matching is used to pair each instance mask with the ground-truth label. This activated instance query allows for a more tailored and precise representation of each map element, in which unique features in instance masks can be aggregated to the specific instance queries.

For point queries, $\mathbf{Q}_{point} \in \mathbb{R}^{N \times D}$ is constructed using a set of predefined learnable weights, which are broadcasted by multi-layer perceptions (MLP) to fuse with each \mathbf{Q}_{lane} . Thus, the hybrid query $\mathbf{Q} \in \mathbb{R}^{M \times N \times D}$ can be obtained by

$$\mathbf{Q} = \mathbf{Q}_{lane} + \text{MLP}(\mathbf{Q}_{point}). \quad (3)$$

Deformable Decoder. To update the hybrid queries, an L -layer decoder can be built by following the multi-scale deformable DETR [51] structure. At each layer l , query embeddings can be updated by interacting with structured multi-level BEV features. Specifically, deformable DETR assigns reference points \mathbf{P}^l as anchors to collect sparse features from $\{\{\mathbf{F}_i\}_{i=1}^3, \mathbf{F}_c\}$, in which \mathbf{P}^l is also the intermediate stage of the normalized lane point positions. Simultaneously, the updated lane point positions \mathbf{P}^{l+1} can be obtained by adding the learned offsets, which are derived from

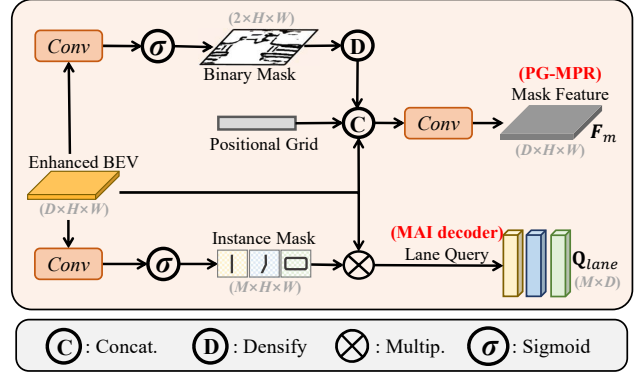


Figure 3. Illustration of **mask constructions** at different stages. In MAI decoder, instance masks are generated to activate lane queries, while binary masks are extracted to provide fine-grained patch features in PG-MPR.

MLP regression branch Reg_{pos} on query \mathbf{Q}^l as follows

$$\mathbf{Q}^l = \text{DeformAttn}(\mathbf{Q}^{l-1}, \mathbf{P}^l, \{\mathbf{F}_i\}_{i=1}^3, \mathbf{F}_c), \quad (4)$$

$$\mathbf{P}^{l+1} = \sigma(\sigma^{-1}(\mathbf{P}^l) + \text{Reg}_{pos}(\mathbf{Q}^l)). \quad (5)$$

The classification scores \mathbf{c}^{l+1} can also be updated by MLP regression on \mathbf{P}^l . With iterative interaction in the L -layer decoder, MGMap can progressively update the semantic and positional information to query embeddings and constantly revise the shape and position of lane lines.

3.3. Position-Guided Mask Patch Refinement

Although the general shapes and structures can be regressed from the instance level, some detailed expressions are still hard to construct. Therefore, it is necessary to achieve fine-grained refinement at a more specific point level. In this section, we design a refinement module to utilize binary mask features more specifically.

Mask Feature Construction. As shown in Figure 3, binary mask $\mathbf{M}_b \in \mathbb{R}^{2 \times H_{BEV} \times W_{BEV}}$ can be obtained by applying sigmoid function after basic convolution on \mathbf{F}_c , in which auxiliary loss with rasterized supervision is used at the training stage. With the guidance of binary segmentation learning, more highlighted features can be utilized to distinguish lane line information from the background.

After that, the binary mask feature \mathbf{F}_m can be constructed based on the binary map \mathbf{M}_b . We densify the dimension of \mathbf{M}_b from 2 to 32 by $\text{D}(\cdot)$. Then, we concatenate the densified binary map with \mathbf{F}_c and a 2-channel normalized positional grid \mathbf{G}_{bev} , which comprises spatial local information features for each pixel to provide relative location information. Finally, these three kinds of characteristic features are fused by the convolutional operation as below

$$\mathbf{F}_m = \text{Conv}(\text{Concat}(\mathbf{F}_c, \text{D}(\mathbf{M}_b), \mathbf{G}_{bev})). \quad (6)$$

Compared to \mathbf{F}_c , \mathbf{F}_m emphasizes specific location and semantic information around lane lines to differentiate the

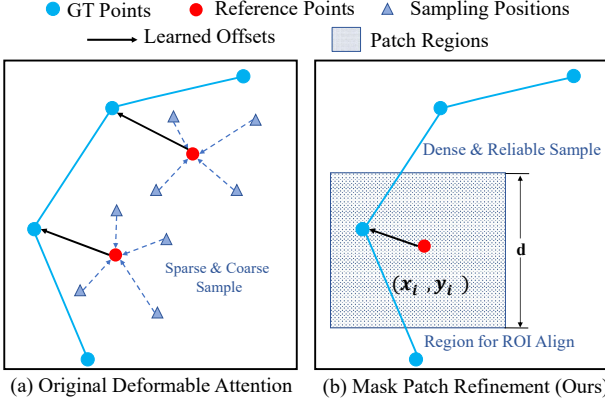


Figure 4. (a) The conventional deformable attention extracts sparse features from sampling points, which may select irrelevant features; (b) Our proposed **Mask Patch Refinement** extracts more relevant features from the region of reliable patch.

feature from ambiguous backgrounds, which provides more clear location information.

Patch Extraction and Refinement. As shown in Figure 4 (b), for the i -th point of the lane line, the patch region is a squared bounding box B_i centered at the point’s coordinates (x_i, y_i) , which is regressed from the last layer of the MAI decoder. The size of the patch region is determined by the hyper-parameter d .

Once the location and region scale are determined, we apply the function f_{ext} , to extract local patch features \mathbf{v}_i for each point in each map instance. This process is implemented using ROIAlign [13], which is employed to extract and unify the features as follows

$$\mathbf{v}_i = f_{ext}(\mathbf{F}_m, \mathbf{R}(B_i), (x_i, y_i)), \quad (7)$$

where $\mathbf{R}(\cdot)$ denotes the denormalized patch region computation. The extraction function f_{ext} utilizes basic convolutional blocks after bilinear interpolation and pooling to locate and align the corresponding values in the patch features. Consequently, each extracted semantic patch feature \mathbf{v}_i has the shape of $D \times 5 \times 5$, which effectively mitigates the issue of misalignment. Compared to the sparse deformable attention design in Figure 4 (a), the mask patch design contains dense and more reliable features.

With query embedding \mathbf{Q} from the L-layer MAI decoder and patch region feature $\mathbf{V} = \{\mathbf{v}_i\}_{i=0}^N$, we leverage multi-head attention to refresh the query features, which can be written as follows

$$\mathbf{Q}^{s+1} = \text{MultiHeadAtt}\left(\frac{\mathbf{Q}^s \mathbf{V}^s}{\sqrt{D}}\right), \quad (8)$$

where s represents the stage in the PG-MPR module and D is the dimension of features. Query embeddings can be refreshed at a more detailed and specific point level.

Finally, point coordinates and the classification scores can be regressed from query \mathbf{Q}^s through MLP branches

Reg_{pos} and Reg_{cls} , which can be formulated as

$$\mathbf{P}^{s+1} = \sigma(\sigma^{-1}(\mathbf{P}^s) + \text{Reg}_{pos}(\mathbf{Q}^s)), \quad (9)$$

$$\mathbf{c}^{s+1} = \text{Reg}_{cls}(\mathbf{Q}^s). \quad (10)$$

3.4. Training Loss

MGMap is trained in an end-to-end manner. Bipartite matching is employed to pair predicted map instances with their ground-truth counterparts. With the regression of points and class labels, auxiliary loss is required for mask segmentation. Concretely, the total loss is the sum of detection loss and mask segmentation loss $\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{mask}$.

Detection Loss. Lane detection aims to regress lane coordinates and classification labels. As in [23], we employ L_1 loss to calculate point-to-point Manhattan distance between the predicted points \hat{p}_{ij} and ground-truth p_{ij} . The edge direction loss is also considered in adjacent points by cosine similarity. \mathcal{L}_{focal} calculates the classification loss, which can be formulated as

$$\mathcal{L}_{lane} = \sum_{i,j=0}^{M,N} (\lambda_{dis} \text{Dis}(\hat{p}_{ij}, p_{ij}) + \lambda_{dir} \text{CosSim}(\hat{e}_{ij}, e_{ij})), \quad (11)$$

$$\mathcal{L}_{det} = \mathcal{L}_{lane} + \lambda_{cls} \sum_{i=0}^M \mathcal{L}_{focal}(\hat{\mathbf{c}}_i, \mathbf{c}_i), \quad (12)$$

where p_{ij} is the j -th point of the i -th instance and e_{ij} is the direction of two adjacent points. λ_{dis} , λ_{dir} , and λ_{cls} are the weighted factors for the losses of point regression, direction adjustment, and label regression, respectively.

Mask Construction Loss. Mask learning is able to reduce the risk of overfitting by pixel-level intensive supervision. We use the combination of cross-entropy loss and dice loss [28] to deal with the unbalanced segmentation problems. Given the output masks $\hat{\mathbf{M}}_{ins}$ and $\hat{\mathbf{M}}_b$, we formulate the auxiliary loss as below

$$\mathcal{L}_{mask} = \lambda_{ins} \mathcal{L}_{ins}(\hat{\mathbf{M}}_{ins}, \mathbf{M}_{ins}) + \lambda_b \mathcal{L}_b(\hat{\mathbf{M}}_b, \mathbf{M}_b), \quad (13)$$

where λ_{ins} and λ_b are the corresponding loss weights.

4. Experiments

4.1. Datasets and Benchmarks

We conduct extensive experiments on two public datasets, including *nuScenes* [1] and *Argoverse2* [46]. *nuScenes* dataset contains 1000 driving scenes collected from Boston and Singapore. 750 and 150 scene sequences are annotated for training and validation, respectively. Each scene sequence consists of 40-keyframe data with a sampling rate of 2Hz. For each keyframe, there are 6 PV images and the corresponding point clouds from 32-beam LiDAR. *Argoverse2* contains 1000 scenes collected from six cities with 7 PV images. We use a subset of *Argoverse2*, which is provided by the Online HD Map Construction Challenge¹. We

¹<https://github.com/Tsinghua-MARS-Lab/Online-HD-Map-Construction-CVPR2023>

Method	Backbone	Epochs	$AP_{chamfer}$				AP_{raster}				FPS
			ped.	div.	bou.	avg.	ped.	div.	bou.	avg.	
<i>Camera-Based Methods</i>											
HMapNet [ICRA22] [19]	EB0	30	14.4	21.7	33.0	23.0	-	-	-	-	-
InstaGraM [CVPRW23] [39]	EB4	30	33.8	47.2	44.0	41.7	-	-	-	-	-
MapTR [ICLR23] [23]	R50	24	46.3	51.5	53.1	50.3	32.4	23.5	17.1	24.3	15.7
MapTR [ICLR23] [23]	R50	30	45.2	53.8	54.3	51.1	32.9	24.9	18.9	25.6	15.7
MapVR [NeurIPS23] [49]	R50	24	47.7	54.4	51.4	51.2	37.5	33.1	23.0	31.2	15.7
PivotNet [ICCV23] [9]	R50	30	53.8	55.8	59.6	57.4	-	-	-	-	9.6
BeMapNet [CVPR23] [32]	R50	30	57.7	62.3	59.4	59.8	-	-	-	-	4.4
MGMap(Ours)	R50	30	57.4	63.5	63.3	61.4	46.5	36.5	28.7	37.2	11.6
MapTRV2 [arxiv23] [24]	R50	24	59.8	62.4	62.4	61.5	-	-	-	-	-
MGMap*(Ours)	R50	24	61.8	65.0	67.5	64.8	-	-	-	-	-
VectorMapNet [ICML23] [26]	R50	110+ft	42.5	51.4	44.1	46.0	-	-	-	-	-
MapTR [ICLR23] [23]	R50	110	56.2	59.8	60.1	58.7	43.6	35.7	25.8	35.0	15.7
MapVR [NeurIPS23] [49]	R50	110	55.0	61.8	59.4	58.8	46.0	39.7	29.9	38.5	15.7
BeMapNet [CVPR23] [32]	R50	110	62.6	66.7	65.1	64.8	-	-	-	-	4.4
MGMap(Ours)	R50	110	64.4	67.6	67.7	66.5	54.5	42.1	37.4	44.7	11.6
<i>LiDAR-Based Methods</i>											
HMapNet [ICRA22] [19]	PP	30	10.4	24.1	37.9	24.1	-	-	-	-	-
VectorMapNet [ICML23] [26]	PP	110	42.5	51.4	44.1	34.0	-	-	-	-	-
MapTR [ICLR23] [23]	Sec	24	48.5	53.7	64.7	55.6	38.9	30.1	41.1	36.7	6.0
MGMap(Ours)	Sec	24	63.5	66.7	73.6	67.9	53.4	44.4	52.6	50.1	5.5
<i>Camera-LiDAR Fusion Methods</i>											
HMapNet [ICRA22] [19]	EB0 & PP	30	16.3	29.6	46.7	31.0	-	-	-	-	-
VectorMapNet [ICML23] [26]	EB0 & PP	110+ft	48.2	60.1	53.0	53.7	-	-	-	-	-
MapTR [ICLR23] [23]	R50 & Sec	24	55.9	62.3	69.3	62.5	46.4	38.4	49.2	44.7	5.2
MapVR [NeurIPS23] [49]	R50 & Sec	24	60.4	62.7	67.2	63.5	52.4	46.4	54.4	51.1	5.2
MGMap(Ours)	R50 & Sec	24	67.7	71.1	76.2	71.7	59.6	47.3	54.6	53.8	4.8

Table 1. Quantitative evaluation of map vectorization on *nuScenes val.* at $60m \times 30m$ perception range under different input modalities and backbone settings, “EB0”, “EB4”, “R50”, and “Sec” correspond to the backbones Efficient-B0, Efficient-B4 [41], ResNet50 [12], and SECOND [48] for LiDAR, respectively. “ft” means the two-stage fine-tune strategy. “MGMap*” means the reimplemented structure based on stronger MapTRV2 [32]. The inference speed is measured on the same computer with a single NVIDIA Tesla V100 GPU.

mainly focus on three map elements, including lane-divider (div.), ped-crossing (ped.), and road boundary (bou.).

4.2. Evaluation Metrics

To facilitate comprehensive evaluations, we employ the Chamfer distance-based metrics, including average precision $AP_{chamfer}$ [19] and the IoU-based average precision AP_{raster} [49], which evaluates the model from the point-coordinate aspect considers each map element as a whole unit separately. This ensures that the map vectorization quality can be assessed from different perspectives.

Chamfer-Distance AP. For fair comparisons, $AP_{chamfer}$ is inherited from previous map vectorization works [23, 26]. Specifically, the average Euclidean distances between sampled points in each map line and the nearest points in the ground-truth labels are measured. The AP is calculated under the average of three distance thresholds, $\tau_{chamfer} \in \{0.5m, 1m, 1.5m\}$. Each prediction is treated as true-positive (TP) when the distance is below the threshold. $AP_{chamfer}$ evaluates map construction quality from point-level perspectives, since it calculates the distances between points as the measured errors.

IoU-based AP. Following MapVR [49], AP_{raster} is calculated by the Intersection over Union (IoU) among pixels. Both predictions and ground truth labels are rasterized into polylines in HD maps. Raster map size is set to 480×240 , and each map element is dilated by two pixels on each side. Detection quality is evaluated by calculating the IoU of the rasterized representations between the prediction and ground truth. The calculation of AP is set under thresholds $\tau_{IoU} \in \{0.25 : 0.5 : 0.05\}$ for line-shaped elements and $\tau_{IoU} \in \{0.5 : 0.75 : 0.05\}$ for polygon-shaped pedestrian crossings, where 0.05 represents the step size for the change of threshold. AP_{raster} is calculated under the average of all the thresholds among the upper and lower bounds. Thus, AP_{raster} considers each lane line at the pixel level and evaluates map quality from the whole instance perspective.

4.3. Implementation Details

To ensure fair comparisons, we choose ResNet50 [12] as the image backbone. SECOND [48] is adopted as the backbone for LiDAR modality. The BEV size, defined as $H_{BEV} \times W_{BEV}$, is set to 200×100 . The maximum number of instances and point queries are set to 50 and 20,

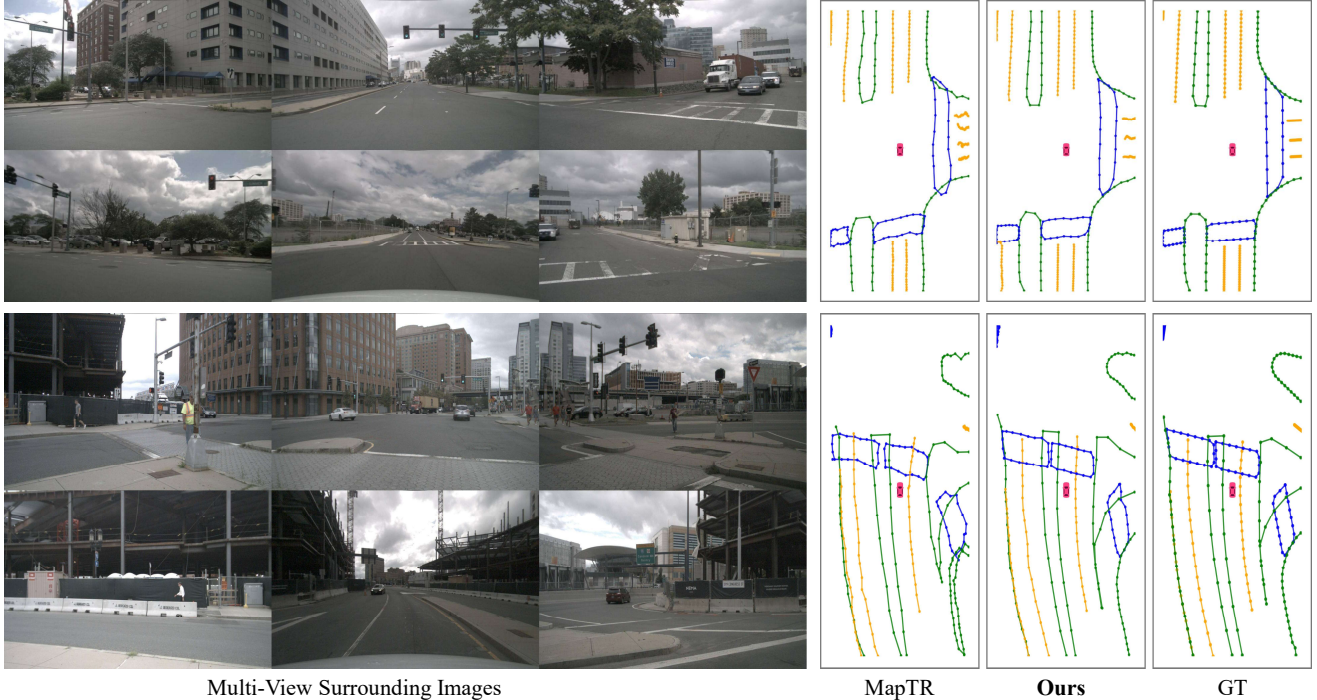


Figure 5. The visual results of MapTR [23], our proposed **MGMap** approach and the corresponding ground truth.

respectively. In PG-MPR module, the hyper-parameter d is set to 0.1, which is the normalized patch size. We use AdamW [34] optimizer with a learning rate of $6e^{-4}$. All models are trained on 8 NVIDIA Tesla V100 GPUs with a batch size of 6 per GPU. The weighted factors for detection loss are the same as those in MapTR [23]. For mask branches, $\{\lambda_{ins}, \lambda_b\}$ are set to $\{2, 15\}$, respectively.

4.4. Main Results

Performance on nuScenes Dataset. As shown in Table 1, we compare MGMap approach against the SOTA methods on the validation set of *nuScenes* with different settings. The experimental performance is evaluated under $AP_{chamfer}$ and AP_{raster} . It can be seen that our proposed approach outperforms previous methods and obtains the best performance. Compared with baseline MapTR [23], MGMap archives 10.3 mAP improvement with multi-view camera-based input under the same settings of ResNet-50 and 30 epochs for training. It is worth noting that MGMap achieves 67.9 mAP for LiDAR and 71.7 mAP for fusing camera data with LiDAR, which demonstrates the strong generalization capability of our scheme. Furthermore, the visual results of MGMap in several driving scenarios are shown in Figure 5. More comparison results under different conditions are included in the supplementary material.

Performance on Argoverse2 Dataset. By following the settings on the online HDMap construction challenge, we re-implement MapTR and MGMap on the *Argoverse2* dataset. Table 2 presents our experimental results. It can be

Method	Backbone	$AP_{ped.}$	$AP_{div.}$	$AP_{bou.}$	mAP
HDMaNet [19]	EB0	5.7	13.1	37.6	18.8
VectorMapNet [26]	R50	37.2	50.4	40.6	42.7
MapTR [23]	R50	50.6	60.7	61.2	57.4
MGMap(Ours)	R50	52.8	67.5	68.1	62.8

Table 2. Performance comparison with baseline methods at $60m \times 30m$ perception range on a subset of *Argoverse2* provided by the Online HD Map Construction Challenge.

observed that our method achieves competitive performance on the *Argoverse2* dataset, MGMap achieves 5.4 mAP improvement compared with MapTR, which further shows the effectiveness of our proposed approach.

Performance on Enlarged Perception Ranges. To evaluate the robustness of model, we conduct experiments on the enlarged perception ranges. Under the same settings, we re-implement MapTR and our MGMap with the perception ranges of $60m \times 60m$ and $30m \times 90m$ at the X-axis and Y-axis in BEV space, in which query numbers are proportionally enlarged to account for the basic property. All models are trained for 30 epochs. Table 3 reports the experiment results. Compared to MapTR, our MGMap achieves consistent performance improvements, with a 9.5 mAP improvement on the $60m \times 60m$ perceptions range setting and a 10.2 mAP improvement on the $30m \times 90m$ range setting.

4.5. Ablation Study

In this section, ablation experiments are conducted to examine the effectiveness of our proposed modules and designs.

BEV range	Method	AP _{ped.}	AP _{div.}	AP _{bou.}	mAP
60m × 60m	MapTR [23]	37.9	44.4	41.6	41.3
	MGMap(Ours)	48.7	53.4	50.0	50.8
	ΔmAP	+10.8	+9.0	+8.4	+9.5
90m × 30m	MapTR [23]	36.5	43.7	39.1	39.8
	MGMap(Ours)	46.4	56.1	49.1	50.5
	ΔmAP	+9.9	+12.4	+10.0	+10.2

Table 3. Experimental results with enlarged perception range settings on the *nuScenes* dataset. Our proposed MGMap approach outperforms MapTR significantly on all evaluation metrics.

For fair comparisons, all experiments are conducted on the camera modality of the *nuScenes* dataset with 30 epochs for training. ResNet50 is employed as the image backbone.

Ablation on Mask-Guided Design. Our mask-guided design comprises two main components: the MAI decoder and the PG-MPR module, which are utilized to handle the overall shape structure and specific details from the instance level and the point level. The MAI construction leverages the masks to obtain global structure information, which contributes to shape understanding. Furthermore, the patch refinement considers the local level more precisely. This strategy allows for more specific and fine-grained adjustments to individual points, which contributes to precise localization. The ablation results presented in Table 4 demonstrate the impact of each level design. When compared to the absence of mask guidance, the inclusion of the MAI decoder and PG-MPR module leads to individual improvements of 1.9 mAP and 2.6 mAP, respectively. Furthermore, the combination of these two components in the overall design achieves the highest performance, with a mAP of 61.4.

Ablation on EML Neck. To investigate the effectiveness of the EML neck design, we conduct ablation experiments to compare it with FPN in PV image space. Multi-scale BEV features contain enriched semantic and location information with larger receptive fields, which is beneficial for the positioning of irregular shapes. As shown in Table 5, compared to baselines without multi-level features, EML design at BEV space (referred to as “BEV”) achieves a larger performance gain. While PV space design fails to achieve the expected effect (“PV”). Besides, enhanced BEV features also contribute to mask generation and boost the performance of the mask-guided design (“MG”).

Ablation on PG-MPR Design. Finally, we investigate the ablation experiment on the setting of the position-guided mask patch refinement module. In our experiment settings, we compare different selections of patch size d and the number of refinement stages s . The evaluated normalized patch sizes range from 0.08 to 0.12, and refinement stages are set from 1 to 3. As illustrated in Table 6, applying the refinement strategy contributes to performance improvement. Experiments also indicate that the oversized patches will introduce irrelevant information and adversely affect performance while undersized patches lead to information loss

Ins.	Point	AP _{ped.}	AP _{div.}	AP _{bou.}	mAP
		53.1	60.1	59.5	57.6
✓		55.4	61.7	61.5	59.5
	✓	54.9	63.4	62.3	60.2
✓	✓	57.4	63.5	63.3	61.4

Table 4. Ablation study of mask-guided design at the instance and point levels. Ins. denotes the instance-level MAI decoder and Point means the point-level PG-MPR module design.

Baseline	PV	BEV	MG	AP _{ped.}	AP _{div.}	AP _{bou.}	mAP
✓				45.2	53.8	54.3	51.1
✓	✓			41.7	49.9	52.6	48.1
✓	✓		✓	47.3	54.1	55.1	52.2
✓		✓		53.1	60.1	59.5	57.6
✓		✓	✓	57.4	63.5	63.3	61.4

Table 5. Ablation study of EML neck design by investigating the performance of multi-level features at PV and BEV stages. Empirical results show that the best performance is achieved by taking advantage of the BEV-level EML neck for mask-guided design.

d	s	AP _{ped.}	AP _{div.}	AP _{bou.}	mAP
/	/	55.4	61.7	61.5	59.5
0.08	2	55.8	64.3	62.3	60.8
0.1	1	56.3	62.6	62.5	60.5
0.1	2	57.4	63.5	63.3	61.4
0.1	3	56.0	62.9	62.9	60.6
0.12	2	56.0	62.7	61.5	60.1

Table 6. Performance of point-level PG-MPR design with different patch sizes (d) and refinement stages (s). The first row means the result without the point-level refinement.

and also result in suboptimal results. The best performance is achieved when the selected patch size is set to 0.1 along with the two-stage refinement.

5. Conclusion

In this paper, we propose MGMap, an effective approach to online HD map vectorization with the guidance of learned masks. By taking advantage of masks at both the instance and point levels, we alleviate the challenges of rough detection and loss of details arising from subtle and sparse annotations in HD maps. Our proposed MGMap not only demonstrates state-of-the-art performance but also exhibits strong robustness in online map vectorization across various experimental settings. For future work, fusing with other perceptual tasks to construct a more comprehensive representation is still a direction worth exploring, which holds promise for further advancements in autonomous driving.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants (62376244). It is also supported by Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

- [1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 2, 5
- [2] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird’s-eye-view traffic scene understanding from onboard images. In *ICCV*, pages 15661–15670, 2021. 2
- [3] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Pixel classification is not all you need for semantic segmentation. pages 17864–17875, 2021. 2
- [4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 2
- [5] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *ECCV*, pages 660–676. Springer, 2020. 2
- [6] Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, and Wenyu Liu. Sparse instance activation for real-time instance segmentation. In *CVPR*, pages 4433–4442, 2022. 2
- [7] Fang Da and Yu Zhang. Path-aware graph attention for hd maps in motion prediction. In *ICRA*, pages 6430–6436. IEEE, 2022. 1
- [8] Jian Ding, Nan Xue, Gui-Song Xia, Bernt Schiele, and Dengxin Dai. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15413–15423, 2023. 2
- [9] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In *ICCV*, pages 3672–3682, 2023. 2, 6
- [10] Hao Dong, Xianjing Zhang, Xuan Jiang, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, Huimin Lu, Juho Kannala, and Xieyuanli Chen. Superfusion: Multilevel lidar-camera fusion for long-range hd map generation and prediction. *arXiv preprint arXiv:2211.15656*, 2022. 1, 2
- [11] 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. In *CVPR*, pages 11525–11533, 2020. 1
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6
- [13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 5
- [14] Yuzhe He, Shuang Liang, Xiaofei Rui, Chengying Cai, and Guowei Wan. EgoVn: Achieving precise ego-localization using lightweight vectorized maps. *arXiv preprint arXiv:2307.08991*, 2023. 1
- [15] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023. 1
- [16] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2
- [17] Bo Jiang, Shaoyu Chen, Xinggang Wang, Bencheng Liao, Tianheng Cheng, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, and Chang Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022. 1
- [18] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: science and systems*, page 1. Atlanta, GA, USA, 2007. 1
- [19] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, pages 4628–4634. IEEE, 2022. 1, 2, 6, 7
- [20] 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. In *ECCV*, pages 1–18. Springer, 2022. 2, 3
- [21] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *CVPR*, pages 9131–9140, 2020. 2
- [22] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. In *NeurIPS*, 2022. 2
- [23] 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 *ICLR*, 2022. 1, 2, 4, 5, 6, 7, 8
- [24] 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. 2, 6
- [25] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, pages 531–548. Springer, 2022. 2
- [26] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, pages 22352–22369. PMLR, 2023. 1, 2, 6, 7
- [27] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huiji Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. In *ICRA*, pages 2774–2781. IEEE, 2023. 2
- [28] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. Ieee, 2016. 5
- [29] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. 2

- [30] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird’s eye view semantic segmentation from arbitrary camera rigs. In *WACV*, pages 5935–5943, 2023. [2](#)
- [31] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210. Springer, 2020. [2](#)
- [32] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *CVPR*, pages 13218–13228, 2023. [2](#), [6](#)
- [33] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird’s-eye-view. In *ICCV*, pages 8690–8699, 2023. [2](#)
- [34] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. [7](#)
- [35] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird’s eye view. In *ITSC*, pages 1–7. IEEE, 2020. [2](#)
- [36] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, pages 11138–11147, 2020. [1](#)
- [37] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IROS*, pages 4758–4765. IEEE, 2018. [2](#)
- [38] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IROS*, pages 5135–5142. IEEE, 2020. [2](#)
- [39] Juyeb Shin, Francois Rameau, Hyeonjun Jeong, and Dong-suk Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. In *CVPRW*, 2023. [2](#), [6](#)
- [40] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *ICCV*, pages 5229–5238, 2019. [2](#)
- [41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. [6](#)
- [42] Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *CVPR*, pages 13926–13935, 2021. [2](#)
- [43] Song Wang, Wentong Li, Wenyu Liu, Xiaolu Liu, and Jianke Zhu. Lidar2map: In defense of lidar-based semantic map construction using online camera distillation. In *CVPR*, pages 5186–5195, 2023. [1](#), [2](#)
- [44] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, pages 913–922, 2021. [2](#)
- [45] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, pages 180–191. PMLR, 2022. [2](#)
- [46] 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 *NeurIPS*, 2021. [2](#), [5](#)
- [47] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. [3](#)
- [48] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. [6](#)
- [49] Gongjie Zhang, Jiahao Lin, Shuang Wu, Yilin Song, Zhipeng Luo, Yang Xue, Shijian Lu, and Zuoguan Wang. Online map vectorization for autonomous driving: A rasterization perspective. In *NeurIPS*, 2023. [6](#)
- [50] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. [1](#)
- [51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. [1](#), [3](#), [4](#)