HIMap: HybrId Representation Learning for End-to-end Vectorized HD Map Construction

Yi Zhou[1], Hui Zhang[1], Jiaqian Yu[1], Yifan Yang[1], Sangil Jung[2], Seung-In Park[2], ByungIn Yoo[2],
[1]Samsung R&D Institute China-Beijing (SRC-B)
[2]Samsung Advanced Institute of Technology (SAIT), South Korea
{yi0813.zhou, hui123.zhang, jiaqian.yu, yifan.yang}@samsung.com

Abstract

*Vectorized High-Definition (HD) map construction requires predictions of the category and point coordinates of map elements (*e.g. *road boundary, lane divider, pedestrian crossing,* etc.*). State-of-the-art methods are mainly based on point-level representation learning for regressing accurate point coordinates. However, this pipeline has limitations in obtaining element-level information and handling element-level failures,* e.g. *erroneous element shape or entanglement between elements. To tackle the above issues, we propose a simple yet effective HybrId framework named HIMap to sufficiently learn and interact both point-level and element-level information. Concretely, we introduce a hybrid representation called HIQuery to represent all map elements, and propose a point-element interactor to interactively extract and encode the hybrid information of elements,* e.g. *point position and element shape, into the HIQuery. Additionally, we present a point-element consistency constraint to enhance the consistency between the point-level and element-level information. Finally, the output point-element integrated HIQuery can be directly converted into map elements' class, point coordinates, and mask. We conduct extensive experiments and consistently outperform previous methods on both nuScenes and Argoverse2 datasets. Notably, our method achieves* 77.8 *mAP on the nuScenes dataset, remarkably superior to previous SOTAs by* 8.3 *mAP at least.*

1. Introduction

Constructing accurate High-Definition (HD) maps is very important for the safety of autonomous driving systems [18, 19, 22, 47, 57, 59, 62]. HD maps [28, 32, 39, 44, 58] can provide comprehensive environmental information, such as road boundary, lane divider and pedestrian crossing, for perception [13, 31, 66], prediction [16, 24, 38] and planning [2, 17, 45]. A vectorized HD map consists of multiple map elements [32], each corresponding to a symbol on the road, such as a divider line, a pedestrian crossing area *etc.*

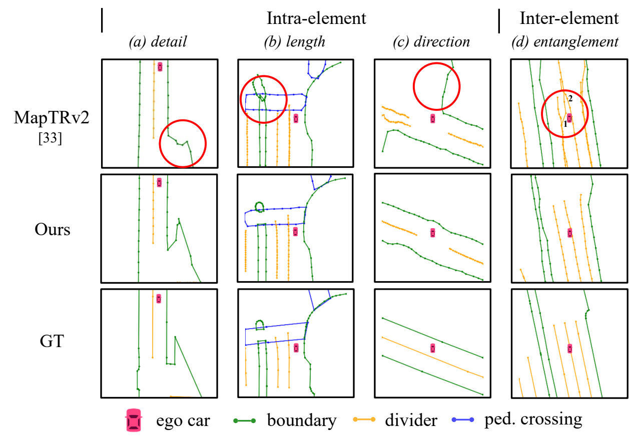

Figure 1. **Examples of previous failures and our improved results.** Compared with the previous point-level representation learning pipeline [33], our proposed hybrid representation learning method generates richer details, more accurate shapes of elements, and avoids the inter-element entanglement. Best viewed in color.

Each vectorized map element is usually represented as a finite set of discrete points. Vectorized HD map construction [10, 28, 32, 39, 48] aims at classifying and localizing the map elements in the Bird's-Eye-View (BEV) space. The reconstruction results contain the class and point coordinates of elements, cf. Figure 1.

Classic works [32, 33, 39, 61] mainly focus on point-level representation learning. VectorMapNet [39] introduces a keypoint representation to represent the outline of map elements and explores a coarse-to-fine two-stage framework. MapTR [32] proposes the permutation-equivalent modeling of the point set and utilizes a deformable decoder [70] to directly regress point coordinates of elements. MapTRv2 [33] further incorporates dense supervision on both BEV and perspective views and a one-to-many matching strategy to improve the accuracy. However, such a pipeline limits the model's capability to learn element-level information and correlations. As shown in the first row of Figure 1(a), the corner detail of the road boundary is missing due to the inaccurate positions of some

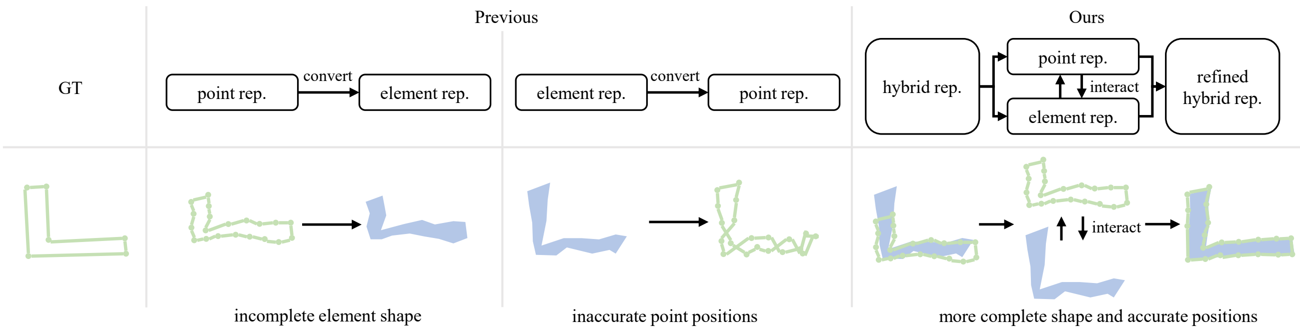

Figure 2. **Illustration of our motivation for point-element interaction.** Previous works [10, 48, 65] usually lack the interaction between point and element, easily leading to either an incomplete element shape or inaccurate point positions. With the point-element interaction based on hybrid representation (shortened to rep.), our method achieves a more complete shape and accurate positions simultaneously.

points. In (b) and (c), the length and direction of the element are not accurate due to the missing overall information. In (d), lane divider 1 and 2 are intertwined on account of similar point-level features of dividers. Based on the above observations, we argue the importance of learning element-level information.

Some intuitive solutions for utilizing element-level information have been studied in a few existing works [10, 48, 65]. MapVR [65] applies differentiable rasterization to vectorized outputs and performs segmentation supervision on the rasterized HD maps. BeMapNet [48] first detects map elements and then utilizes a piecewise Bezier head to output the details of each map element. PivotNet [10] directly converts the point-level representations into element-level representations by designing the Point-to-Line Mask module. However, these attempts utilize point-level and element-level information in a sequential manner, lacking the point-element interaction, cf. Figure 2. This leads to suboptimal performance empirically (cf. Section 4).

To better learn and interact information of map elements, in this paper, we propose a simple yet effective HybrId framework named HIMap based on hybrid representation learning. We first introduce a hybrid representation called HIQuery to represent all map elements in the map. It is a set of learnable parameters and can be iteratively updated and refined by interacting with BEV features. Then we design a multilayer hybrid decoder to encode hybrid information of map elements (*e.g.* point position, element shape) into HIQuery and perform point-element interaction, cf. Figure 2. Each layer of the hybrid decoder comprises a point-element interactor, a self-attention, and an FFN. Inside the point-element interactor, a mutual interaction mechanism is performed to realize the exchange of point-level and element-level information and avoid the learning bias of single-level information. In the end, the output point-element integrated HIQuery can be directly converted into elements' point coordinates, classes, and masks. Furthermore, we propose a point-element consistency constraint to strengthen the consistency between point-level and element-level information.

Our main contributions can be summarized as follows:

- We propose a hybrid representation (*i.e.* HIQuery) to represent all elements in the HD map and a simple yet effective HybrId framework(*i.e.* HIMap) for end-to-end vectorized HD map construction.
- To simultaneously predict accurate point coordinates and element shape, we introduce a point-element interactor to extract and interact information of both point-level and element-level.
- Our method significantly outperforms previous works on both nuScenes [3] and Argoverse2 [56] datasets, achieving new state-of-the-art results of $77.8, 72.7$ mAP respectively.

2. Related Work

HD Map Construction. HD map construction in Bird's-Eye-View (BEV) space [10, 14, 28, 30, 32, 39, 41, 48, 60] generates a map based on onboard sensor observations, such as RGB images from multi-view cameras and point clouds from LiDAR. Existing methods can be categorized into two types: rasterized HD map estimation [28, 30, 41, 60] and vectorized HD map construction [10, 28, 32, 39, 48]. Rasterized HD map estimation is formulated as the semantic segmentation task in the BEV space. The semantic class of each pixel is predicted. However, the rasterized HD map is not an ideal representation for the downstream tasks due to the lack of the instance-level distinction and structure information of map elements. Vectorized HD map construction resolves the above limitations by representing the map with a set of map elements. Each map element is usually represented by an ordered sequence of discrete points. In this paper, we focus on the vectorized HD map construction task and discuss how to produce accurate vectorized elements by exploiting both point-level and element-level information.

Vectorized HD Map Construction. To produce vectorized HD maps, earlier work [28] proposes a multi-task framework with hand-crafted post-processing. However, the heuristic post-processing may accumulate errors from different branches and restrict the model's generalization ability. To solve the above issues, subsequent works attempt to build an end-to-end framework based on point-level rep-

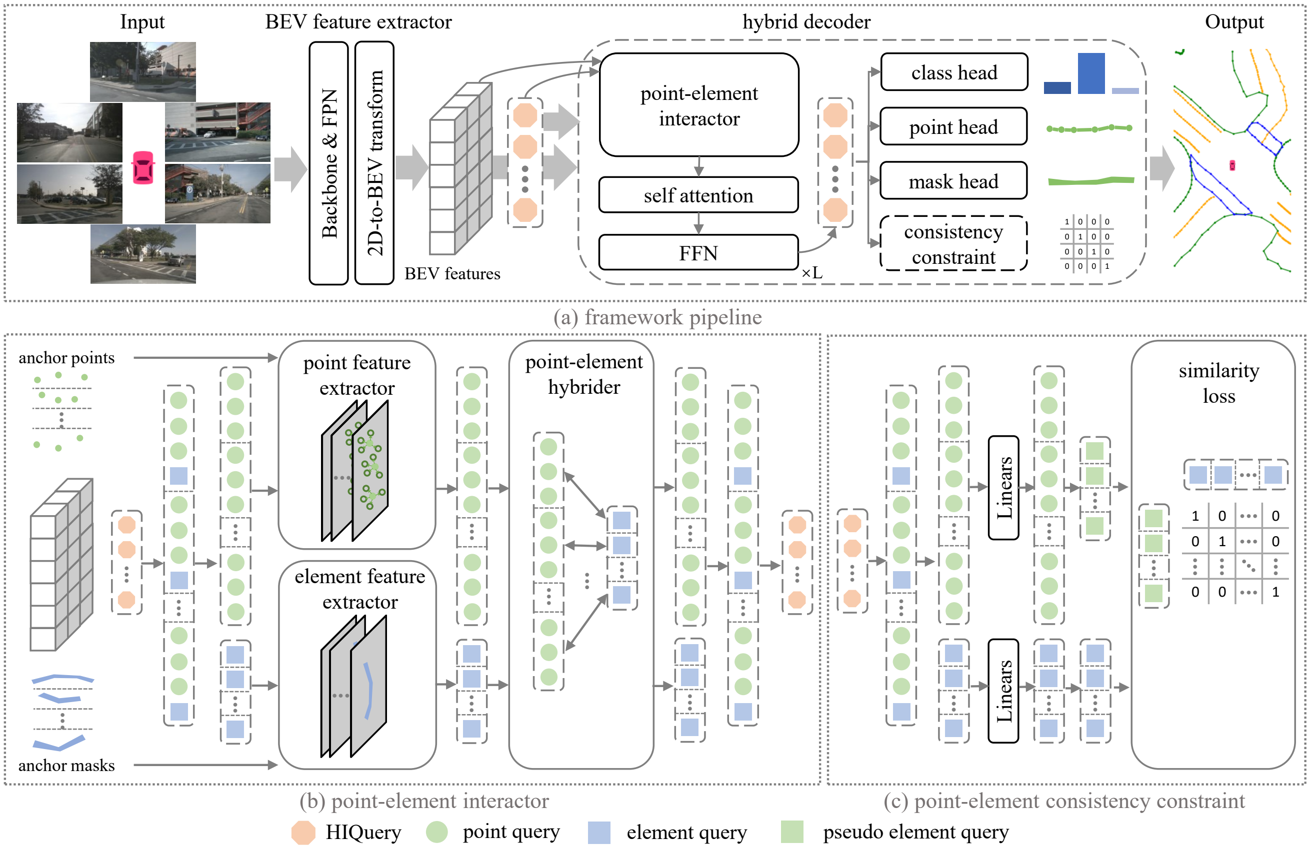

Figure 3. **Overview of the HIMap.** Top: The pipeline of HIMap, consisting of a BEV feature extractor and a hybrid decoder. It takes multi-view images as input and outputs vectorized map elements in an end-to-end fashion. Bottom: Detailed process of the point-element interactor, which interactively extracts both point-level and element-level information of map elements, and the point-element consistency for enhancing the information consistency inside an element and the discrimination between elements. Best viewed in color.

resentations. VectorMapNet [39] explores the keypoint representation and a coarse-to-fine two-stage network. MapTR series [32, 33] propose the permutation-equivalent modeling of the point set and a DETR-like [4] one-stage network. InsightMapper [61] proves the effectiveness of utilizing inner-instance point information. A few recent works try to learn the element-level information. MapVR [65] introduces differentiable rasterization and adds element-level segmentation supervision. BeMapNet [48] detects map elements first and then regresses the detailed points with a piecewise Bezier head. PivotNet [10] designs the Point-to-Line Mask module to convert point-level representation into element-level representation. However, the information interaction between points and elements is lacking in these methods. In this paper, we propose a hybrid representation learning pipeline to simultaneously represent, learn, and interact both point-level and element-level information of map elements.

Lane Detection. Lane detection aims at predicting visible lanes on the road, hence it can be viewed as a subtask of HD map construction. Many existing works focus on 2D lane detection in a single perspective-view image. Traditional methods [9, 21, 23, 52, 69] adopt hand-crafted features and post-processing techniques to predict lanes. Subsequent works replace the hand-crafted feature detectors with deep networks. Lane segmentation pipeline [7, 25–27] and lane detection methods based on different lane representations, *e.g.* point series [29, 35, 49, 50, 54, 67] or parametric curves [11, 36, 51, 53] are explored and proposed. Some recent works extend to 3D lane detection [1, 5, 12, 20, 55, 64], and explore the multi-modality inputs [42]. In comparison, vectorized HD map construction considers more map element categories, and outputs results of the whole surrounding area of the ego car.

3. Method

3.1. Framework Overview

The overall pipeline of HIMap is presented in Figure 3(a).
Input. HIMap is compatible with various onboard sensor data, *e.g.* RGB images from multi-view cameras, point clouds from LiDAR, or multi-modality data. Here we take multi-view RGB images as an example to illustrate HIMap.
BEV Feature Extractor. We extract BEV features from multi-view RGB images with the BEV feature extractor.

It consists of a backbone [15, 40] to extract multi-scale 2D features from each perspective view, an FPN [34] to refine and fuse multi-scale features into single-scale features, and a 2D-to-BEV feature transformation module [6, 30, 43, 46, 68] to map 2D features into BEV features. The BEV features can be denoted as $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, where H, W, C refer to the spatial height, spatial width, and the number of channels of feature maps, respectively.

HIQuery. To sufficiently learn both point-level and element-level information of map elements, we introduce HIQuery to represent all elements in the map. HIQuery is a set of learnable parameters $\mathbf{Q}^h \in \mathbb{R}^{E\times(P+1)\times C}$, where E, P, C denote the maximum number of map elements (*e.g*. 50), the number of points in an element (*e.g*. 20), and the number of channels respectively. Inside HIQuery, $\mathbf{Q}^h_i \in \mathbb{R}^{(P+1)\times C}$ is responsible for one map element with index $i \in \{1, ..., E\}$. In particular, $\mathbf{Q}^h_i$ can be decomposed into two parts, point query $\mathbf{Q}^p_i \in \mathbb{R}^{P\times C}$ and element query $\mathbf{Q}^e_i \in \mathbb{R}^{C}$, corresponding to point-level and element-level information respectively, cf. Figure 3 (b) and (c). With this point-element integrated information, HIQuery can be easily converted into the corresponding elements' point coordinates, classes, and masks.

Hybrid Decoder. The hybrid decoder produces the point-element integrated HIQuery by iteratively interacting HIQuery $\mathbf{Q}^h$ with BEV features $\mathbf{X}$. It contains multiple layers, each comprising a ***point-element interactor***, a self-attention, a Feed Forward Network (FFN), and multiple prediction heads. In each layer $l \in \{1, ..., L\}$, where L is the total number of layers in the hybrid decoder, the point-element interactor first extracts, interacts, and encodes point-level and element-level information of map elements into the input HIQuery $\mathbf{Q}^{h,l-1} \in \mathbb{R}^{E\times(P+1)\times C}$. Then the self-attention and the FFN successively refine both levels of information in the HIQuery. The output point-element integrated HIQuery $\mathbf{Q}^{h,l} \in \mathbb{R}^{E\times(P+1)\times C}$ are forwarded into the class head, point head, and mask head to generate elements' classes, point coordinates, and masks respectively. In the training stage, we apply the point-element consistency constraint on the intermediate representations from point and mask heads to enhance their consistency. The prediction results of the last layer are the final results of HIMap.

3.2. Point-element Interactor

Point-element interactor targets to interactively extract and encode both the point-level and element-level information of map elements into HIQuery. The motivation for interacting two levels of information comes from their complementarity. The point-level information contains the local position knowledge, while the element-level information provides the overall shape and semantic knowledge. Hence the interaction enables mutual refinement of both local and overall information of map elements.

As shown in Figure 3(b), the point-element interactor consists of a point feature extractor, an element feature extractor, and a point-element hybrider. Given BEV features $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ and HIQuery $\mathbf{Q}^{h,l-1} \in \mathbb{R}^{E\times(P+1)\times C}$ generated from the $(l-1)$-th layer, we first discompose $\mathbf{Q}^{h,l-1} \in \mathbb{R}^{E\times(P+1)\times C}$ into point query $\mathbf{Q}^{p,l-1} \in \mathbb{R}^{E\times P\times C}$ and element query $\mathbf{Q}^{e,l-1} \in \mathbb{R}^{E\times C}$. Then we utilize point and element feature extractors to extract respective features from BEV features and leverage point-element hybrider to interact and encode information into HIQuery. In this process, a mutual interaction mechanism is realized by sharing position embeddings when applying two feature extractors and utilizing integrated information to update two levels of query inside the point-element hybrider.

Point Feature Extractor. To extract point-level features, how to sample and let the anchor points close to the element is very important. Inspired by Deformable Attention [70] and DAB-DETR [37], we model the anchor points as a set of learnable 2D points and attend to a small set of key sampling points around an anchor point. Anchor points are randomly initialized with uniform distribution in the $[0, 1]$ range for the first layer, iteratively updated, and forwarded layer by layer. In the l-th layer, the 2D anchor points are the point outputs of $(l-1)$-th layer, which can be denoted as $\mathbf{P}^{l-1} \in \mathbb{R}^{E\times P\times 2}$. Let $j \in \{1, ..., E\times P\}$ index a point query $\mathbf{Q}^{p,l-1}_j \in \mathbb{R}^{C}$ with a 2D anchor point $\mathbf{P}^{l-1}_j \in \mathbb{R}^2$, we first generate the position embeddings $\mathbf{B}^{p,l}_j \in \mathbb{R}^C$ of the point query and the position-aware point query $\widehat{\mathbf{Q}}^{p,l}_j \in \mathbb{R}^C$:

$$
\begin{aligned}
\mathbf{B}^{p,l}_j &= \mathbf{W}_b(\mathbf{P}^{l-1}_j), \\
\widehat{\mathbf{Q}}^{p,l}_j &= \mathbf{Q}^{p,l-1}_j + \mathbf{B}^{p,l}_j,
\end{aligned} \tag{1}
$$

where $\mathbf{W}_b \in \mathbb{R}^{2\times C}$ refers to the learnable parameters of a Linear layer. Denoting the size of the sampling point set as K, the sampled offsets $\Delta\mathbf{P}^l_j \in \mathbb{R}^{K\times 2}$ and attention weights $\mathbf{A}^l_j \in \mathbb{R}^K$ of sampling points can be produced by:

$$
\begin{aligned}
\Delta\mathbf{P}^l_j &= \mathbf{W}_o(\widehat{\mathbf{Q}}^{p,l}_j), \\
\mathbf{A}^l_j &= softmax(\mathbf{W}_a(\widehat{\mathbf{Q}}^{p,l}_j))
\end{aligned} \tag{2}
$$

where $\mathbf{W}_o \in \mathbb{R}^{C\times(K\times 2)}$, $\mathbf{W}_a \in \mathbb{R}^{C\times K}$ refer to the learnable parameters of the Linear layers and the softmax operation is applied along the dimension of sampling points. Then the point feature extractor can be formulated as:

$$
\begin{aligned}
\mathbf{X}^{p,l}_j &= \sum_{k=1}^{K} \mathbf{A}^l_{j,k} \cdot \mathbf{W}_v \mathbf{X}(\mathbf{P}^{l-1}_j + \Delta\mathbf{P}^l_{j,k}), \\
\dot{\mathbf{X}}^{p,l}_j &= \mathbf{X}^{p,l}_j + \mathbf{Q}^{p,l-1}_j,
\end{aligned} \tag{3}
$$

where $\mathbf{W}_v \in \mathbb{R}^{C\times C}$ stands for the learnable parameters of a Linear layer, $\Delta\mathbf{P}^l_{j,k} \in \mathbb{R}^2$ are 2D real numbers with unconstrained range, $\mathbf{A}_{j,k}^{l}$ is scalar attention weight lies in the range $[0,1]$ and normalized by $\sum_{k=1}^{K}\mathbf{A}_{j,k}^{l} = 1$, $\mathbf{X}_{j}^{p,l} \in \mathbb{R}^{C}$ represents the fused point-level features of K sampling points for j-th point, and $\dot{\mathbf{X}}_{j}^{p,l} \in \mathbb{R}^{C}$ denotes the output point-level features by merging the extracted point-level features with point-level information from $(l-1)$-th layer. Following [70], bilinear interpolation is applied in computing $\mathbf{X}(\mathbf{P}_{j}^{l-1} + \Delta\mathbf{P}_{j,k}^{l})$ due to that $\mathbf{P}_{j}^{l-1} + \Delta\mathbf{P}_{j,k}^{l}$ is fractional. After the above process, point-level features of each anchor point are obtained. For all anchor points, the point-level features can be represented as $\dot{\mathbf{X}}^{p,l} \in \mathbb{R}^{E\times P\times C}$.

Element Feature Extractor. We extract element-level features with the element feature extractor built on the Masked Attention [8]. To utilize and enhance the correspondence between points and elements, the position embeddings of the point query are shared with the element query. This is fulfilled by applying weighted summation on $\mathbf{B}^{p,l} \in \mathbb{R}^{E\times P\times C}$ to generate the position embeddings $\mathbf{B}^{e,l} \in \mathbb{R}^{E\times C}$ of the element query. Denote the sinusoidal position embedding [4] of BEV features as $\mathbf{B}^{x,l} \in \mathbb{R}^{H\times W\times C}$, and let i index an element query $\mathbf{Q}_{i}^{e,l-1} \in \mathbb{R}^{C}$ and its position embedding $\mathbf{B}_{i}^{e,l} \in \mathbb{R}^{C}$, where $i \in \{1,...,E\}$, we can formulate the element feature extractor as:

$$\begin{aligned}\widehat{\mathbf{Q}}_{i}^{e,l} &= \mathbf{Q}_{i}^{e,l-1} + \mathbf{B}_{i}^{e,l},\\ \widehat{\mathbf{X}} &= \mathbf{X} + \mathbf{B}^{x,l},\\ \mathbf{X}_{i}^{e,l} &= (\mathbf{M}_{i}^{l-1}\cdot softmax(\widehat{\mathbf{Q}}_{i}^{e,l}\widehat{\mathbf{X}}^{T}))\mathbf{X},\\ \dot{\mathbf{X}}_{i}^{e,l} &= \mathbf{X}_{i}^{e,l} + \mathbf{Q}_{i}^{e,l-1},\end{aligned} \tag{4}$$

where $\widehat{\mathbf{Q}}_{i}^{e,l} \in \mathbb{R}^{C}$ refers to the position-aware element query, $\widehat{\mathbf{X}} \in \mathbb{R}^{H\times W\times C}$ denotes the position-aware BEV features, $\mathbf{X}_{i}^{e,l} \in \mathbb{R}^{C}$ represents the extracted element-level features of i-th element, $\dot{\mathbf{X}}_{i}^{e,l} \in \mathbb{R}^{C}$ refers to the output element-level features by merging the element information from $(l-1)$-the information together, and $\mathbf{M}_{i}^{l-1} \in \{0,1\}^{HW}$ refers to the anchor mask, which is the binarization result of the mask output of i-th element in the $(l-1)$-th layer. The pixel threshold for binarization is empirically set to 0.5. $\widehat{\mathbf{Q}}_{i}^{e,l}\widehat{\mathbf{X}}^{T} \in \mathbb{R}^{H\times W}$ reflects the correlations between the position-aware element query and BEV features. Applying softmax along the $H\times W$ dimension enables the values at positions where the element is located to become high. After utilizing the anchor mask filters out irrelevant areas, the element-level information can be extracted. Element-level features of all elements can be denoted as $\dot{\mathbf{X}}^{e,l} \in \mathbb{R}^{E\times C}$.

Point-element Hybrider. The point-element hybrider aims to interact and encode both point-level and element-level information into HIQuery. It consists of two steps, single-level feature refinement and cross-level query update. Given point-level features $\dot{\mathbf{X}}^{p,l} \in \mathbb{R}^{E\times P\times C}$ and element-level features $\dot{\mathbf{X}}^{e,l} \in \mathbb{R}^{E\times C}$, the single-level feature refinement step is as follows:

$$\ddot{\mathbf{X}}^{p,l} = \mathcal{F}_{rp}(\dot{\mathbf{X}}^{p,l}), \quad \ddot{\mathbf{X}}^{e,l} = \mathcal{F}_{re}(\dot{\mathbf{X}}^{e,l}), \tag{5}$$

where $\ddot{\mathbf{X}}^{p,l} \in \mathbb{R}^{E\times P\times C}$ and $\ddot{\mathbf{X}}^{e,l} \in \mathbb{R}^{E\times C}$ refer to the refined point-level and element-level features respectively, $\mathcal{F}_{rp}$ and $\mathcal{F}_{re}$ represent point-level and element-level refinement procedures (*e.g.* self-attention and FFN) respectively. Then the cross-level query update step can be expressed as:

$$\begin{aligned}\mathbf{Q}^{p,l} &= \ddot{\mathbf{X}}^{p,l} + \mathcal{F}_{ce}(\ddot{\mathbf{X}}^{e,l}),\\ \mathbf{Q}^{e,l} &= \ddot{\mathbf{X}}^{e,l} + \mathcal{F}_{cp}(\ddot{\mathbf{X}}^{p,l}),\end{aligned} \tag{6}$$

where $\mathbf{Q}^{p,l} \in \mathbb{R}^{E\times P\times C}$ refers to the updated point query, $\mathcal{F}_{ce}$ denotes the copy operation for expanding the element-level information into all points of the corresponding element, $\mathbf{Q}^{e,l} \in \mathbb{R}^{E\times C}$ refers to the updated element query, $\mathcal{F}_{cp}$ is the weighted sum operation for accumulating all related point-level information into element level. In this way, both levels of queries are updated with the integrated information of point and element. The point query can not only obtain the local point information but also be complemented and corrected by the overall element information. Meanwhile, the element query earns the overall element information (*e.g.* shape, semantic) as well as the refinement from the local points. Finally, the updated point query $\mathbf{Q}^{p,l} \in \mathbb{R}^{E\times P\times C}$ and element query $\mathbf{Q}^{e,l} \in \mathbb{R}^{E\times C}$ are concatenated to produce the output HIQuery $\mathbf{Q}^{h,l} \in \mathbb{R}^{E\times(P+1)\times C}$.

3.3. Point-element Consistency

Considering the primitive differences between point-level and element-level representations, which focus on local and overall information respectively, the learning of two levels of representations may also interfere with each other. This will increase the difficulty and reduce the effectiveness of information interaction. Therefore, we introduce the point-element consistency constraint to enhance the consistency between point-level and element-level information of each element. As a byproduct, the distinguishability of elements can also be strengthened.

Given point query $\mathbf{Q}^{p,l} \in \mathbb{R}^{E\times P\times C}$ and element query $\mathbf{Q}^{e,l} \in \mathbb{R}^{E\times C}$ from HIQuery $\mathbf{Q}^{h,l} \in \mathbb{R}^{E\times(P+1)\times C}$ in the l-th layer, we first obtain the intermediate point-level representations $\overline{\mathbf{Q}}^{p,l} \in \mathbb{R}^{E\times P\times C}$ and element-level representations $\overline{\mathbf{Q}}^{e,l} \in \mathbb{R}^{E\times C}$ by applying Linear layers in the point head and mask head respectively. Then we generate a pseudo element-level representation $\widetilde{\mathbf{Q}}^{e,l} \in \mathbb{R}^{E\times C}$ as the weighted sum of the point-level representations $\overline{\mathbf{Q}}^{p,l}$, and calculate element-level similarities as:

$$\mathbf{A}^{e,l} = \widetilde{\mathbf{Q}}^{e,l}(\overline{\mathbf{Q}}^{e,l})^{T}, \tag{7}$$

where $\mathbf{A}^{e,l} \in \mathbb{R}^{E\times E}$ is the similarity matrix. Binary cross entropy loss is applied between the calculated similarity matrix and the binary GT matrix. By facilitating

Methods	Backbone	Epoch	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
			hard: $\{0.2, 0.5, 1.0\}$m				*easy:* $\{0.5, 1.0, 1.5\}$m			
HDMapNet [28]	Efficient-B0	30	7.1	28.3	32.6	22.7	24.1	23.6	43.5	31.4
PivotNet [10]	ResNet50	30	34.8	42.9	39.3	39.0	53.8	58.8	59.6	57.4
Ours	ResNet50	30	37.2	49.2	42.8	43.1	62.6	68.4	69.1	66.7
VectorMapNet [39]	ResNet50	110	18.2	27.2	18.4	21.3	36.1	47.3	39.3	40.9
MapTR [32]	ResNet50	110	31.4‡	40.5‡	35.5‡	35.8‡	55.8‡	60.9‡	61.1‡	59.3‡
MapVR [65]	ResNet50	110	-	-	-	-	55.0	61.8	59.4	58.8
BeMapNet [48]	ResNet50	110	44.5	52.7	44.2	47.1	62.6	66.7	65.1	64.8
MapTRv2 [33]	ResNet50	110	-	-	-	-	68.1	68.3	69.7	68.7
MapTRv2† [33]	ResNet50	110	42.9†	49.3†	43.3†	45.2†	67.1†	69.2†	69.0†	68.4†
Ours	ResNet50	110	**47.3**	**57.8**	**49.6**	**51.6** (+4.5)	**71.3**	**75.0**	**74.7**	**73.7** (+5.0)

Table 1. **Comparison to the state-of-the-art on nuScenes *val* set.** The best results with the same backbone are in **bold** and the second in underline. Gains are calculated based on the best and the second results. †, ‡ mean the result is reproduced with public code and released model respectively. "-" means that the corresponding results are not available. The APs under the easy setting of [28] and the APs under the hard setting of [39] are taken from [10].

Methods	Backbone	Epoch	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
			hard: $\{0.2, 0.5, 1.0\}$m				*easy:* $\{0.5, 1.0, 1.5\}$m			
HDMapNet [28]	Efficient-B0	6	9.8	19.5	35.9	21.8	13.1	5.7	37.6	18.8
MapTR [32]	ResNet50	6	28.3	42.2	33.7	34.8	54.7	58.1	56.7	56.5
MapVR [65]	ResNet50	-	-	-	-	-	54.6	60.0	58.0	57.5
PivotNet [10]	ResNet50	6	31.3	47.5	43.4	40.7	-	-	-	-
MapTRv2† [33]	ResNet50	6	34.8†	52.5†	40.6†	42.6†	63.6†	**71.5**†	67.4†	67.5†
Ours	ResNet50	6	**39.9**	**53.4**	**44.3**	**45.8** (+3.2)	**69.0**	69.5	**70.3**	**69.6** (+2.1)
VectorMapNet[39]	ResNet50	24	18.3	33.3	20.4	24.0	38.3	36.1	39.2	37.9
MapTRv2† [33]	ResNet50	24	39.2†	56.5†	43.8†	46.5†	68.3†	**74.1**†	69.2†	70.5†
Ours	ResNet50	24	**44.2**	**57.9**	**47.9**	**50.0** (+3.5)	**72.4**	72.4	**73.2**	**72.7** (+2.2)

Table 2. **Comparison to the state-of-the-art on Argoverse2 *val* set.** The best results trained with the same epochs are in **bold** and second underline. Gains are calculated based on the best and the second results. † means the result is reproduced with public codes. "-" means that the corresponding results are not available. The APs under the easy setting of [28] are taken from [39] and the APs under the hard setting of [28, 32, 39] are taken from [10].

the high similarity between the pseudo and actual element-level representations, the consistency between point-level and element-level information is enhanced.

4. Experiments

4.1. Experimental Settings

NuScenes Dataset. NuScenes dataset [3] provides 1000 scenes, each lasting around 20 seconds, and is annotated at 2Hz. Each sample includes 6 RGB images from surrounding cameras and point clouds from LiDAR sweeps. Following previous methods [32, 33], 700 scenes with 28130 samples are utilized for training and 150 scenes with 6019 samples are used for validation. For a fair comparison, we mainly focus on three categories of map elements, including road boundary, lane divider, and pedestrian crossing.

Argoverse2 Dataset. There are 1000 logs in the Argoverse2 dataset [56]. Each log contains 15s of 20Hz RGB images from 7 cameras, 10Hz LiDAR sweeps, and a 3D vectorized map. The train, validation, and test sets contain $700, 150, 150$ logs, respectively. Following previous works [32, 33], we report results on its validation set and focus on the same three map categories as the nuScenes dataset.

Evaluation Metric. For a fair comparison with previous methods [32, 33], we adopt the mean Average Precision (mAP) metric based on chamfer distance for evaluation. A prediction is considered as True-Positive (TP) only if its chamfer distance to ground truth is less than a specified threshold. Following [10, 48], two different threshold sets corresponding to hard and easy settings, $\{0.2, 0.5, 1.0\}m$ and $\{0.5, 1.0, 1.5\}m$, are considered for evaluation. For each setting, the final AP result is calculated by averaging across three thresholds and all classes. With the ego-car as the center, the perception ranges are $[-15.0m, 15.0m]$ for the X-axis and $[-30.0m, 30.0m]$ for the Y-axis.

Training. For the main results, we employ ResNet50 [15] as the backbone for the multi-view RGB images input, and the SECOND [63] for the LiDAR point clouds input. Training losses include binary semantic segmentation loss [41], classification loss, point coordinate loss, point direction loss [32], mask segmentation loss, and point-element consis-

Methods	Epoch	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
		easy: $\{0.5, 1.0, 1.5\}$m			
HDMapNet [28]	30	16.3	29.6	46.7	31.0
VectorMapNet[39]	110+ft	48.2	60.1	53.0	53.7
MapTR [32]	24	55.9	62.3	69.3	62.5
MapVR [65]	24	60.4	62.7	67.2	63.5
MapTRv2 [33]	24	65.6	66.5	74.8	69.0
Ours	24	**71.0**	**72.4**	**79.4**	**74.3** (+5.3)
MapTR† [32]	110	65.4†	68.3†	74.9†	69.5†
Ours	110	**77.0**	**74.4**	**82.1**	**77.8** (+8.3)

Table 3. **Comparison to the state-of-the-art on nuScenes *val* set with multi modality data.** Both multi-view images and LiDAR point clouds are used as inputs. "ft" refers to the fine-tuning trick adopted in [39]. The APs of [28] are taken from [39].

class	method	$\text{AP}_{0.2m}$	$\text{AP}_{0.5m}$	$\text{AP}_{1.0m}$	$\text{AP}_{1.5m}$
ped.	MapTRv2	0.7	39.7	77.2	88.1
	Ours	3.4 (+2.7)	48.8 (+9.1)	80.4 (+3.2)	88.2 (+0.1)
div.	MapTRv2	29.0	63.6	76.9	81.8
	Ours	35.2 (+6.2)	64.2 (+0.6)	74.3 (-2.6)	78.6 (-3.2)
bou.	MapTRv2	8.0	48.1	75.2	84.2
	Ours	10.9 (+2.9)	54.4 (+6.3)	78.3 (+3.1)	86.8 (+2.6)

Table 4. **Detailed comparison between MapTRv2 [33] and ours under different thresholds on Argoverse2 *val* set.**

tency loss. Corresponding loss weights are empirically set to $2.0, 2.0, 5.0, 0.005, 2.0, 2.0$. The model is trained for $110, 24$ epochs on the nuScenes and Argoverse2 datasets respectively. All the data pre-processing steps for both datasets follow MapTR [32]. More details can be found in the Supplementary Material.

4.2. Comparisons with State-of-the-art Methods

Results on nuScenes. Table 1 presents the comparison of the results on the nuScenes dataset with multi-view RGB images as input. Our HIMap achieves novel state-of-the-art performance ($73.7, 51.6$ mAP) under both easy and hard settings. Specifically, HIMap outperforms MapTRv2 [33], the previous SOTA under the easy setting, by 5.0 mAP. This validates the effectiveness of our hybrid representation in capturing more comprehensive information of elements than the point-level representation [33]. HIMap also exceeds BeMapNet [48], the previous SOTA under the hard setting, by 4.5 mAP. This proves that point-element interaction is superior to sequentially utilizing both levels of information [48]. In addition, Table 3 presents the results with multi-modality inputs (multi-view RGB images and LiDAR point clouds). HIMap also achieves novel SOTA performance, 74.3 mAP for 24 epochs and 77.8 mAP for 110 epochs, exceeding previous methods by 5.3 and 8.3 mAP at least respectively.

Results on Argoverse2. As shown in Table 2, on the Argoverse2 dataset, HIMap consistently exceeds previous SOTAs under both easy and hard settings, no matter training with 6 or 24 epochs. With 24 epochs scheduler, our method

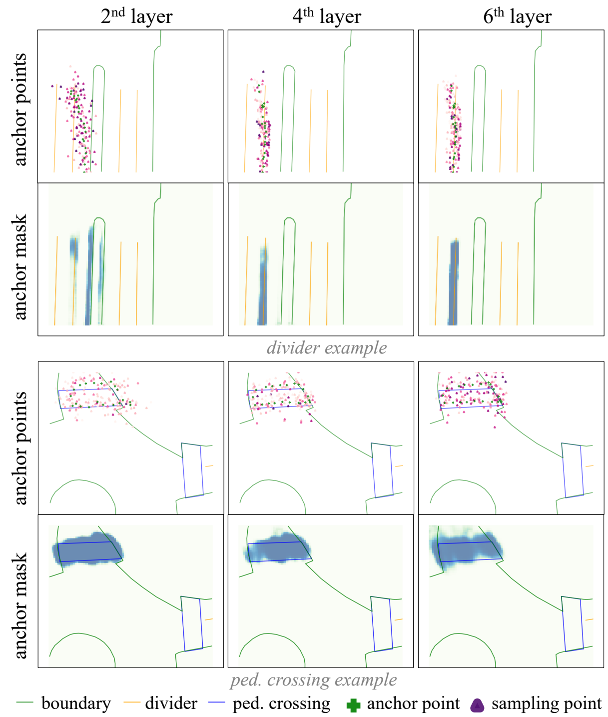

Figure 4. **Attention maps of HIQuery at different layers.** Attention maps are overlaid on the GT. The darker the color, the greater the attention value. Best zoom-in and viewed in color.

outperforms MapTRv2 [33] by $3.5, 2.2$ mAP under hard and easy settings respectively. What's more, we observe that our result of the divider class is lower than MapTRv2 under easy setting, but higher under hard setting. We speculate that our HIMap generates more TPs for a strict threshold (*i.e.* 0.2m). Additionally, in Table 4, we show the detailed result comparison to MapTRv2 [33] under different thresholds. Our HIMap produces larger improvements for more strict thresholds (*e.g.* $0.2, 0.5$ m) indeed.

4.3. Ablation Study

In this part, we analyze the HIQuery and study several aspects[1] to illustrate the effectiveness of the proposed method. Unless otherwise specified, experiments are conducted with ResNet50 as the backbone on the nuScenes *val* set with multi-view RGB images as input, trained for 110 epochs, and evaluated under the easy setting.

What has HIQuery learnt? To better understand what has HIQuery learnt and the effect of point-element interaction, we visualize attention maps of anchor points with its sampling points and anchor masks for a single map element at different layers in Figure 4. As we can observe, anchor points and masks, corresponding to point query and element query inside HIQuery, focus on local and overall

[1]More extensive studies on the hybrid decoder, number of points and elements *etc*., are provided in Supplementary Material.

hybrid	interaction	consistency	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
✗	✗	✗	66.6	69.5	69.3	68.5
✓	✗	✗	67.9	71.6	72.5	70.6
✓	✓	✗	70.6	73.6	75.2	73.1
✓	✓	✓	**71.3**	**75.0**	**74.7**	**73.7**

Table 5. **Effect of key designs in HIMap.** Rows rendered in violet are the final settings.

extractors	hybrider					
share pos	*inte*-P	*inte*-E	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
✗	✗	✗	67.9	71.6	72.5	70.6
✓	✗	✗	69.2	72.7	72.9	71.6
✓	✓	✗	70.0	72.9	73.4	72.1
✓	✗	✓	70.8	73.5	74.1	72.8
✓	✓	✓	**70.6**	**73.6**	**75.2**	**73.1**

Table 6. **Variations in point-element interactor.** "*inte*-P" and "*inte*-E" refer to utilizing integrated information to update point query and element query respectively.

loss weight	$\text{AP}_{ped.}$	$\text{AP}_{div.}$	$\text{AP}_{bou.}$	mAP
0.0	70.6	73.6	75.2	73.1
1.0	72.1	73.9	74.5	73.5
2.0	**71.3**	**75.0**	**74.7**	**73.7**
3.0	70.2	72.9	74.3	72.5

Table 7. **Effect of loss weight for point-element consistency.**

information of elements, respectively. In the divider example, anchor points and masks at 2nd layer stretch across the target divider and a nearby boundary. At the 4th layer, both of them focus on the target divider, but the direction of anchor points is still tilted to the left and the length of the anchor mask is not perfect. At the 6th layer, anchor points and masks fit the target divider better. In the pedestrian crossing example, at the 2nd layer, the anchor points drift to the right and the anchor mask includes extra pixels outside the target pedestrian crossing. After iterative learning and interaction, both anchor points and mask are shifted to the pedestrian crossing. These visualizations validate that point-element interaction helps to achieve mutual refinement.

HIMap. In Table 5, we study several key designs of HIMap step-by-step, including the hybrid representation, the point-element interactor, and the point-element consistency. We first build a point-level representation learning baseline by adjusting configurations of MapTR [32], *e.g.* FPN, 2D-to-BEV transformation module, *etc*. As shown in the 1st row of Table 5, it achieves 68.5 mAP. Then we leverage the hybrid representation to learn both point-level and element-level information simultaneously. The element-level information is refined with Masked attention [8] and supervised with segmentation loss. This method (2nd row) reaches 70.6 mAP, bringing 2.1 mAP gain over the baseline. To interact and achieve mutual refinement of both levels of information, we further replace the deformable and mask attention with the point-element interactor. This setting (3rd row) obtains 73.1 mAP and brings 2.5 mAP extra gain. After adding the point-element consistency, HIMap finally obtains 73.7 mAP, securing a 5.2 mAP gain over the baseline.

Point-element Interactor. There are several key factors in point-element interactor, including whether sharing position embeddings between feature extractors, whether utilizing the integrated information to update point query and element query. Correspondingly, we denote these factors as "share pos", "*inte*-P", and "*inte*-E" and study them in Table 6. To focus on the effect of point-element interactor, point-element consistency is not employed in this part. Without all these factors, it is equivalent to learning HIQuery with deformable and mask attention, which gets 70.6 mAP. Sharing position embeddings aims to utilize and enhance the correspondence between points and elements, and brings 1.0 mAP gain (2nd row). Utilizing the integrated information to update only point query, only element query, or both queries (3rd, 4th, and 5th rows) brings 0.5, 1.2, 1.5 mAP gains, respectively. This validates that utilizing the integrated information to update both queries enables mutual refinement of points and elements. With all these factors, the point-element interactor finally brings a 2.5 mAP gain.

Point-element Consistency. We adjust the loss weight of the point-element consistency constraint to observe the effect. As shown in Table 7, the results are not sensitive to the loss weight, but a too-large weight may cause the two levels of information to be too similar to reduce the effect of point-element interaction. Empirically, we set the loss weight to 2.0 and achieve 73.7 mAP.

5. Conclusion

In this paper, we introduce a simple yet effective HybrId framework (*i.e.* HIMap) based on hybrid representation learning for end-to-end vectorized HD map construction. In HIMap, we introduce HIQuery to represent all map elements, a point-element interactor to interactively extract and encode both point-level and element-level information into HIQuery, and a point-element consistency constraint to strengthen the consistency between two levels of information. With the above designs, HIMap achieves new SOTA performance on both nuScenes and Argoverse2 datasets.

Limitation Discussion. (1) This paper mainly focuses on improving the map reconstruction accuracy, and we leave the model acceleration for future work. (2) Currently the proposed method constructs 2D HD maps. Considering that the height change of the road is very important for autonomous driving, how to predict accurate 3D HD maps is worth exploring further. (3) We consider the point-element consistency in HIMap but do not discuss the consistency of HD maps across multiple timestamps. We believe that exploring temporal information and predicting consistent HD maps are valuable research directions.

References

[1] Yifeng Bai, Zhirong Chen, Zhangjie Fu, Lang Peng, Pengpeng Liang, and Erkang Cheng. Curveformer: 3d lane detection by curve propagation with curve queries and attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7062–7068. IEEE, 2023. 3

[2] Thibault Buhet, Emilie Wirbel, Andrei Bursuc, and Xavier Perrotton. Plop: Probabilistic polynomial objects trajectory planning for autonomous driving. *arXiv preprint arXiv:2003.08744*, 2020. 1

[3] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 6

[4] 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. 3, 5

[5] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*, pages 550–567. Springer, 2022. 3

[6] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022. 4

[7] Ziye Chen, Yu Liu, Mingming Gong, Bo Du, Guoqi Qian, and Kate Smith-Miles. Generating dynamic kernels via transformers for lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6835–6844, 2023. 3

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 5, 8

[9] Hendrik Deusch, Jürgen Wiest, Stephan Reuter, Magdalena Szczot, Marcus Konrad, and Klaus Dietmayer. A random finite set approach to multiple lane detection. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 270–275. IEEE, 2012. 3

[10] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3682, 2023. 1, 2, 3, 6

[11] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022. 3

[12] Noa Garnett, Rafi Cohen, Tomer Pe'er, Roee Lahav, and Dan Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2921–2930, 2019. 3

[13] Chunrui Han, Jianjian Sun, Zheng Ge, Jinrong Yang, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception. *arXiv preprint arXiv:2303.05970*, 2023. 1

[14] Xiaoshuai Hao, Hui Zhang, Yifan Yang, Yi Zhou, Sangil Jung, Seung-In Park, and ByungIn Yoo. Mbfusion: A new multi-modal bev feature fusion method for hd map construction. In *IEEE International Conference on Robotics and Automation*, 2024. 2

[15] 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. 4, 6

[16] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021. 1

[17] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12732–12741, 2021. 1

[18] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 1

[19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1

[20] Shaofei Huang, Zhenwei Shen, Zehao Huang, Zi-han Ding, Jiao Dai, Jizhong Han, Naiyan Wang, and Si Liu. Anchor3dlane: Learning to regress 3d anchors for monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17451–17460, 2023. 3

[21] Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo. Multi-lane detection in urban driving environments using conditional random fields. In *2013 IEEE Intelligent vehicles symposium (IV)*, pages 1297–1302. IEEE, 2013. 3

[22] Keishi Ishihara, Anssi Kanervisto, Jun Miura, and Ville Hautamaki. Multi-task learning with attention for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2902–2911, 2021. 1

[23] Heechul Jung, Junggon Min, and Junmo Kim. An efficient lane detection algorithm for lane departure detection. In *2013 IEEE Intelligent vehicles symposium (IV)*, pages 976–981. IEEE, 2013. 3

[24] Alexey Kamenev, Lirui Wang, Ollin Boer Bohan, Ishwar Kulkarni, Bilal Kartal, Artem Molchanov, Stan Birchfield, David Nistér, and Nikolai Smolyanskiy. Predictionnet: Real-time joint probabilistic traffic prediction for planning, control, and simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8936–8942. IEEE, 2022. 1

[25] 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, 2021. 3

[26] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1947–1955, 2017.

[27] Jun Li, Xue Mei, Danil Prokhorov, and Dacheng Tao. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE transactions on neural networks and learning systems*, 28(3):690–703, 2016. 3

[28] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022. 1, 2, 6, 7

[29] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1): 248–258, 2019. 3

[30] 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 *European conference on computer vision*, pages 1–18. Springer, 2022. 2, 4

[31] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020. 1

[32] 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. *arXiv preprint arXiv:2208.14437*, 2022. 1, 2, 3, 6, 7, 8

[33] 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. 1, 3, 6, 7

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[35] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: a top-to-down lane detection framework based on conditional convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3773–3782, 2021. 3

[36] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021. 3

[37] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 4

[38] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021. 1

[39] 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. 1, 2, 3, 6, 7

[40] 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. 4

[41] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023. 2, 6

[42] Yueru Luo, Xu Yan, Chaoda Zheng, Chao Zheng, Shuqi Mei, Tang Kun, Shuguang Cui, and Zhen Li. M^ 2-3dlanenet: Multi-modal 3d lane detection. *arXiv preprint arXiv:2209.05996*, 2022. 3

[43] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991. 4

[44] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4227–4236, 2021. 1

[45] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified multi-task model for behavior prediction and planning. *arXiv preprint arXiv:2106.08417*, 2(7), 2021. 1

[46] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 4

[47] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021. 1

[48] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13218–13228, 2023. 1, 2, 3, 6, 7

[49] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020. 3

[50] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 294–302, 2021. 3

[51] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021. 3

[52] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, and Keqiang Li. A novel curve lane detection based on improved river flow and ransa. In *17th international ieee conference on intelligent transportation systems (itsc)*, pages 133–138. IEEE, 2014. 3

[53] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[54] Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, and Tianzhu Zhang. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1392–1401, 2022. 3

[55] Ruihao Wang, Jian Qin, Kaiying Li, Yaochen Li, Dong Cao, and Jintao Xu. Bev-lanedet: An efficient 3d lane detection based on virtual camera via key-points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1002–1011, 2023. 3

[56] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 2, 6

[57] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022. 1

[58] Xindi Wu, KwunFung Lau, Francesco Ferroni, Aljoša Ošep, and Deva Ramanan. Pix2map: Cross-modal retrieval for inferring street maps from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17514–17523, 2023. 1

[59] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M López. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547, 2020. 1

[60] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Neural map prior for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17535–17544, 2023. 2

[61] Zhenhua Xu, Kenneth KY Wong, and Hengshuang Zhao. Insightmapper: A closer look at inner-instance information for vectorized high-definition mapping. *arXiv preprint arXiv:2308.08543*, 2023. 1, 3

[62] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023. 1

[63] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 6

[64] Chengtang Yao, Lidong Yu, Yuwei Wu, and Yunde Jia. Sparse point guided 3d lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8363–8372, 2023. 3

[65] 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. *arXiv preprint arXiv:2306.10502*, 2023. 2, 3, 6, 7

[66] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4537–4546, 2022. 1

[67] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 898–907, 2022. 3

[68] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13760–13769, 2022. 4

[69] Shengyan Zhou, Yanhua Jiang, Junqiang Xi, Jianwei Gong, Guangming Xiong, and Huiyan Chen. A novel lane detection based on geometrical model and gabor filter. In *2010 IEEE Intelligent Vehicles Symposium*, pages 59–64. IEEE, 2010. 3

[70] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 4, 5