




ADMap: Anti-disturbance Framework for Vectorized HD Map Construction

Haotian Hu¹, Fanyi Wang^{*,2}, Yaonong Wang¹, Laifeng Hu¹, Jingwei Xu¹,
and Zhiwang Zhang^{*,3}

¹ Zhejiang Leapmotor Technology CO., LTD.

{hu_haotian,wang_yaonong, hu_laifeng, xu_jingwei}@leapmotor.com

² Zhejiang University

11730038@zju.edu.cn

³ Ningbo Tech University

zhiwang.zhang@nbt.edu.cn

Abstract. In the field of autonomous driving, online High-definition (HD) map construction is crucial for planning tasks. Recent studies have developed several high-performance HD map construction models to meet the demand. However, the point sequences generated by recent HD map construction models are jittery or jagged due to prediction bias and impact subsequent tasks. To mitigate this jitter issue, we propose the Anti-Disturbance Map construction framework (ADMap), which contains Multi-scale Perception Neck (MPN), Instance Interactive Attention (IIA), and Vector Direction Difference Loss (VDDL). By exploring the point sequence relations between and within instances in a cascading manner, our proposed ADMap effectively monitors the point sequence prediction process, and achieves state-of-the-art performance on the nuScenes and Argoverse2 datasets. Extensive results demonstrate its ability to produce stable and reliable map elements in complex and changing driving scenarios.[†]

Keywords: Autonomous driving · HD map · Anti-disturbance

1 Introduction

In the field of autonomous driving, high-definition (HD) map construction [11, 15, 16, 24, 26] is a very important task. It involves converting the sensor-collected information into instance-level vector representations, such as lane lines, road boundaries, and pedestrian crossings. These representations enable the vehicle to capture detailed road topology and ground semantics information while driving, which can be effectively applied to downstream regulation tasks. In recent years, early HD map construction works [3, 9, 12, 19, 28, 29] predicts dense ground information, which results in redundancy in model computation and annotation. HDMapNet [11] groups dense pixel segmentation results into sparse vectorized

*Corresponding authors.

[†]The code is available at <https://github.com/hht1996ok/ADMap>.

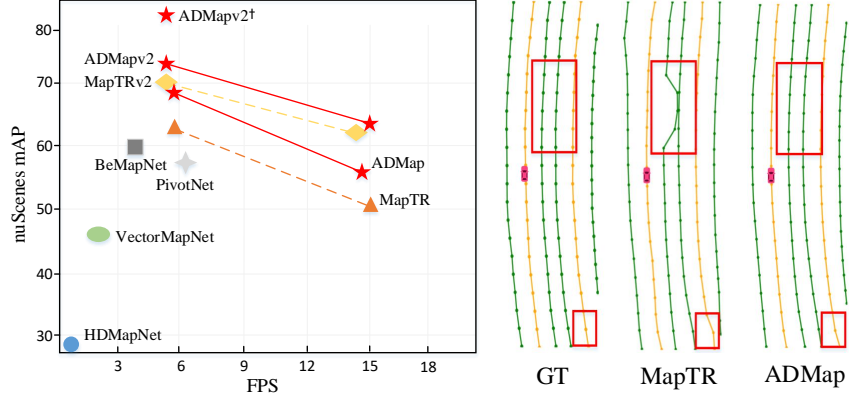


Fig. 1: Performance and visualization comparison between ADMap and baseline. The two endpoints of the line segment in the left figure indicate results of the multi-modal and camera-only frameworks, respectively. The right figure shows that ADMap effectively mitigates the point sequence jitter problem. † has the same meaning as in Table 1.

instances, but requires complex post-processing process. VectorMapNet [16] predicts vectorized instances for the first time, using an autoregressive decoder to predict the instance points in an ordered manner. MapTR [14] predicts vectorised instances end-to-end and resolves feature ambiguities caused by different point order directions effectively. MapTRv2 [15] adds decoupled self-attention to capture intra-instance point relations in parallel.

Although previous HD maps construction methods [11, 14–16] achieve promising results, these methods still suffer from the following issues. Firstly, existing methods do not fully consider inter-instance and intra-instance interactions, resulting in incomplete interaction between instance points and map topology information, which leads to inaccurate prediction points. In addition, these methods neglect multi-scale features in the BEV maps and inaccurately construct instances for different sizes. Furthermore, previous methods only using L1 loss [5, 14, 22] or cosine embedding loss [14, 15] for distance supervision, which does not effectively utilize the geometric relationship between point sequences to constrain the prediction process. As shown in Figure 1, we can find that the predicted points in instances tend to be jittered or shifted caused by the above issues. These results can cause the constructed instance vectors to become distorted or jagged, thus seriously affect the quality and utility of the online high-precision maps.

To address these issues and predict point sequence topology more accurately, we propose the Anti-Disturbance Map construction (ADMap) framework, which contains Instance Interactive Attention (IIA), Multi-scale Perception Neck (MPN), and Vector Direction Difference Loss (VDDL). To capture multi-scale features in the BEV maps, we propose Multi-scale Perception Neck (MPN). MPN improves the accuracy of constructing instances with significant size differences

in the scene without increasing inference time. To investigate inter-instance and intra-instance interactions, we propose Instance Interactive Attention (IIA). IIA flexibly encodes instance-level and point-level information, feature interactions between instance embeddings to further help the network capture the relationships between point-level embeddings. And more accurate point-level information makes the construction more accurate. To better utilize the geometric relationship between point sequences, we propose Vector Direction Difference Loss (VDDL). VDDL models the association between instance points and vector direction differences, and uses vector direction differences as losses to constrain the construction process of point sequences more precisely. Besides, the difference in real vector direction is utilised to assign varying weights to the points in the instances, ensuring that the model can effectively capture the rapidly changing map information in the scene.

We validate the effectiveness of our proposed ADMap in nuScenes benchmark [1] and Argoverse2 benchmark [23]. ADMap achieves state-of-the-art performance in both nuScenes and Argoverse2, compared to existing constructed vectorized HD map models. In nuScenes dataset, ADMap improved performance by 4.2% and 5.5% in camera-only and multimodal frameworks, compared to the baseline method MapTR. The best performance of ADMapv2 (i.e., our proposed ADMap framework on MapTRv2 [15] implementation) reaches 82.8%. ADMapv2 not only reduces inference latency but also improves performance of baseline method MapTRv2. ADMap also performs well in Argoverse2. ADMapv2 improves mAP by 62.9% while FPS remains 14.8, indicating that ADMap is an efficient and high-precision framework for generating accurate and smooth map topology in complex scenes. Our contributions are summarized as follows:

- We propose an End-to-end framework ADMap for stable vectorized HD maps construction.
- In ADMap, our proposed MPN captures multi-scale information more precisely without increasing computational resources. The proposed IIA achieves effective interaction between inter-instance and intra-instance information to alleviate the problem of instance point position offset. The proposed VDDL models the vector direction difference and supervises the construction process of point order position using topological information.
- ADMap enables real-time construction of vectorized HD maps and achieves state-of-the-art performance on both the nuScenes and Argoverse2 benchmarks.

2 Related Work

2.1 Lane detection

In previous works, lane line detection was typically considered as a standalone task. Information was gathered through sensors such as cameras and lidar to identify and locate lane lines. LaneNet [18] proposes a semantic segmentation of 2D lane lines and clusters them. 3D-LaneNet [6] is a pioneering work in the field

of monocular 3D lane lines, which proposes a new type of dual-path structure that implements inverse perspective mapping (IPM) projection of features inside the network. GenLaneNet [7] optimizes the anchor representation of 3D-LaneNet and uses the anchor to predict 3D Lane in a more reliable coordinate system. PersFormer [2] proposes a unified framework for 2D and 3D lane detection. The framework introduces transformer into the spatial transformation module to improve the robustness of features.

2.2 HD map construction

Traditional online local maps [3, 9, 12, 13, 17, 20, 29] are mostly based on semantic segmentation of multiple perspective views, which are converted to bird’s eye view (BEV) using IPM projection and splicing, and then go through a complicated post-processing to get the map elements. HDMapNet [11] constructs the HD map from the generated semantic segmentation results, instance embedding, and direction prediction. However, it requires complicated post-processing and has low accuracy. VectorMapNet [16] achieves end-to-end HD map construction by aligning the construction task with the target detection paradigm and generating ordered point sets one by one through Polyline Generator. MapTR [14] generates map elements directly using the Deformable DETR-based object detection paradigm. It introduces permutation-equivalent in the model to address the order uniqueness constraints during point sequence matching.

ADMap utilizes features between instance embeddings to model the relationships between point-level embeddings. Also, the introduction of vectorial direction differences also supervises the point order position more finely.

3 Method

3.1 Overview

As shown in Figure 2, ADMap takes a multiview image and a point cloud as inputs. The features of the multi-view image $I = \{I_1, \dots, I_M\}$ are extracted by the 2D backbone. The 2D image is then converted to a BEV perspective by LSS-base view transformer so as to obtain camera BEV features $F_{cam} \in R^{H \times W \times C}$. The point cloud $P \in R^{n \times C}$ is voxelized and placed into a 3D backbone to obtain lidar BEV features $F_{lidar} \in R^{H \times W \times C}$. Fusion Layers fuses camera BEV features and lidar BEV features into fusion BEV features $F_{fusion} \in R^{H \times W \times C}$. Then fusion BEV features inputs into the multi-scale perception neck (Section 3.2), which fuses the multi-scale BEV features from top to bottom. This ensures that the network can predict instances of different scales in the scene accurately. We add instance interactive attention (Section 3.3) in decoder, which helps the network to better capture associations between point levels through extracted instance embeddings. Furthermore, the details of vector direction difference loss are presented in Section 3.4.

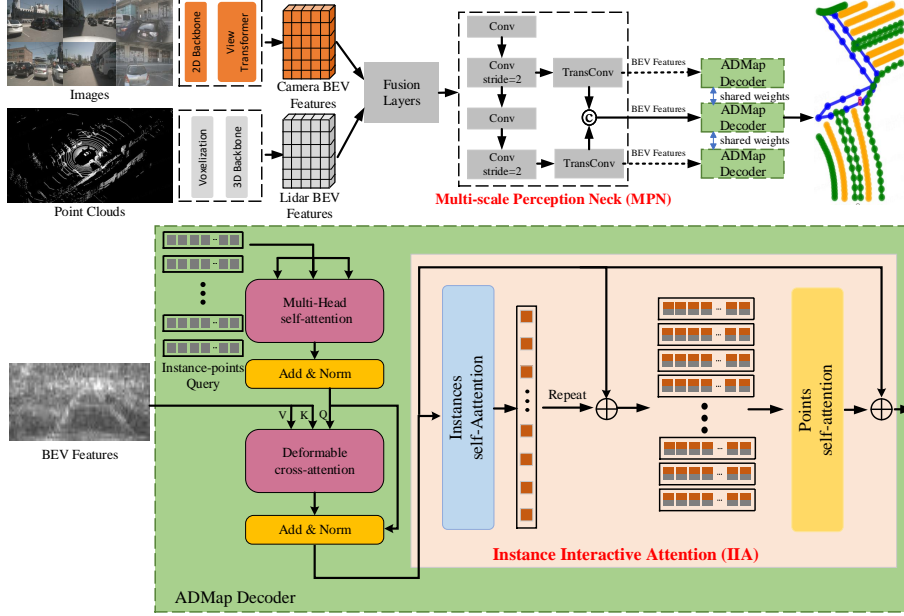


Fig. 2: Schematic diagram of the overall framework of ADMap. The figure displays the MPN and IIA proposed in this paper. The process is performed only during training when indicated by the dashed line, and during both training and inference when indicated by the solid line. In decoder, Instance-Points query are defined to represent the topology of the map, and self-attention and cross-attention are used to interact with the BEV map. Instance self-attention and Points self-attention further interact with inter- and intra-instance information.

3.2 Multi-scale Perception Neck

Conventional FPN structures only output fused multi-scale features, which prevents the network from delicately capturing information at all levels of scale. To make more detailed BEV features available to the network, we propose multi-scale perception neck (MPN), whose inputs are fusion BEV features $F_{fusion} \in R^{H \times W \times C}$. After downsampling, the BEV features of each layer are connected to the upsampled layer to restore its original size. The final feature map of each layer are merged into a multi-scale BEV feature $F_{mc} \in R^{H \times W \times C}$. Figure 2 shows that the dotted line indicates that the step is only implemented during training, while the solid line indicates that the step is implemented during both training and inference. During training, the multi-scale BEV features and the BEV feature maps at each level are fed into the ADMap Decoder, which allows the network to predict the instance information of the scene in different scales. During the inference process, MPN only outputs multi-scale BEV features $F_{mc} \in R^{H \times W \times C}$.

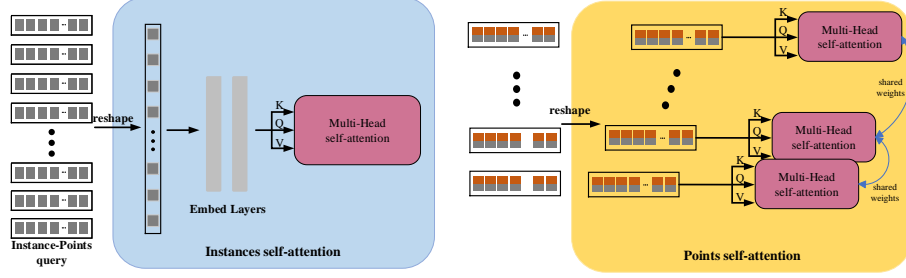


Fig. 3: Schematic diagram of Instance self-attention and Points self-attention. The point and channel dimensions in the Instance-Points query are merged and put into embedded layers consisting of multiple MLPs to compress the dimensions. These query are then used in multi-head self-attention for instance-level interactions. Groups the output query of instance self-attention so that each instance is fed into multi-head self-attention separately for point-level interaction.

and not the feature maps at each level, which ensures that the model inference speed remains constant.

3.3 ADMap Decoder

A set of instance-level queries $q_{ins} \in R^{1 \times N_i \times C}$ and a set of point-level queries $q_{pos} \in R^{N_p \times 1 \times C}$ are defined in the decoder, and we subsequently share the point-level queries across all instances, and Instance-Points queries $q \in R^{N_p \times N_i \times C}$ are defined as:

$$q = q_{ins} + q_{pos} \quad (1)$$

The decoder contains several cascading decoding layers that iteratively update the Instance-Points queries q . In each decoding layer, these queries are fed into self-attention, which allows the Instance-Points queries to exchange information with each other. Deformable attention is used to facilitate interaction between the Instance-Points queries and the multiscale BEV features $F_{mc} \in R^{H \times W \times C}$.

Instance Interactive Attention In order to better acquire the features of each instance in the decoding stage, we propose instance interactive attention (IIA). IIA comprises Instance self-attention and Point self-attention. Unlike the parallel extraction of instance-level and point-level embeddings [15], our IIA cascades to extract point sequence embeddings, feature interactions between instance embeddings assist the network in learning the topological relationships between instance points.

Figure 3 shows that the hierarchical embedding $F_{hie} \in R^{(N_i * N_p) \times C}$ produced by cross-attention is inputted into Instance self-attention. After merging

the point dimension with the channel dimension, the dimension transformation of the hierarchical embedding is $F_{hie} \in R^{N_i \times (C * N_p)}$. Subsequently, the hierarchical embedding is put into Multilayer Perceptrons (MLPs) to obtain the instance features, which is put into multi-head self-attention to capture the topological relations among instances. The instance embedding $F_{ins} \in R^{N_i \times C}$ is obtained. To include instance-level information in the point-level embedding, we add the instance embedding $F_{ins} \in R^{(N_i * 1) \times C}$ to the hierarchical embedding $F_{hie} \in R^{(N_i * N_p) \times C}$. The summed features are fed into Point self-attention, which interacts with the points within each instance to further finely correlate the topological relationships between point sequences.

Vector Direction Difference Loss HD maps contain vectorized static map elements, including lane lines, road boundaries, and pedestrian crossings. We propose vector direction difference loss (VDDL) for these open shapes (lane lines, road boundaries) and closed shapes (pedestrian crossings). By modeling the point sequence vector direction inside the instance, the difference between the predicted vector direction and the real vector direction is used for finer supervision of the point sequence position. Furthermore, it is believed that points with significant differences in real vector direction indicate significant changes in the topology of certain parts of the scene, which are less predictable. Therefore, more attention is required to ensure that the network can accurately predict these points of drastic change.

Figure 4 illustrates the modeling of the predicted vector line $\{L_{i,j}^{pre}\}_{i=0,j=0}^{N_i,N_p-1}$ and the real vector line $\{L_{i,j}^{gt}\}_{i=0,j=0}^{N_i,N_p-1}$ in the predicted point sequence $\{P_{i,j}^{pre}\}_{i=0,j=0}^{N_i,N_p}$ and the real point sequence $\{P_{i,j}^{gt}\}_{i=0,j=0}^{N_i,N_p}$.

To ensure that the loss increases with the angular difference, we compute the vector line angle difference cosine $cos\theta_{i,j}^{pre}$:

$$cos\theta_{i,j}^{pre} = \begin{cases} \frac{sum(L_{i,j}^{pre} * L_{i,j}^{gt})}{norm(L_{i,j}^{pre}) * norm(L_{i,j}^{gt})}, & j \neq N_p, \\ 0, & j = N_p, \end{cases} \quad (2)$$

The $sum()$ accumulates the 2D coordinate positions of the vector lines, while the $norm()$ normalizes them. We assign weights of different magnitudes to the vector angular differences of the points in the real example. The weights $W_{i,j}^{vec}$ are defined as follows:

$$cos\theta_{i,j}^{gt} = \begin{cases} \frac{sum(L_{i,j-1}^{gt} * L_{i,j}^{gt})}{norm(L_{i,j-1}^{gt}) * norm(L_{i,j}^{gt})}, & j \neq N_p \\ 0, & j = 0 \text{ or } N_p, \end{cases} \quad (3)$$

$$W_{i,j}^{vec} = \begin{cases} exp(\frac{1.0 - cos\theta_{i,j}^{gt}}{2}), & i \neq 0 \text{ and } i \neq N_p \\ 1.0, & i = 0 \text{ or } i = N_p, \end{cases} \quad (4)$$

where N_p represents the number of points in the instance, and $exp()$ represents an exponential function with base e. The weight of the first and last points

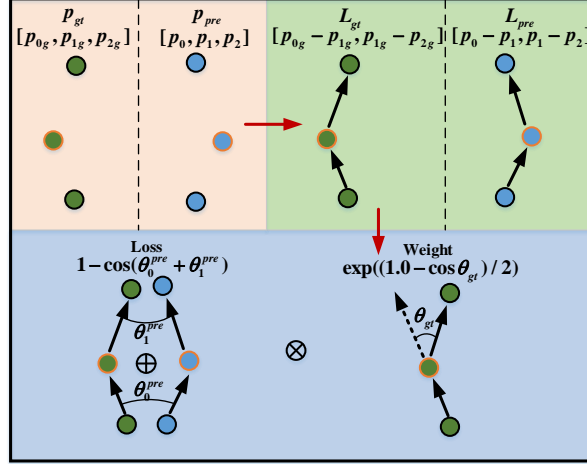


Fig. 4: Flowchart of VDDL. The point sequence P is modeled as a vector line L and the vector direction difference between the predicted and ground truth is calculated. The weights of each instance point are obtained from the geometric topological relations of ground truth.

is set to 1 since they cannot compute the vector angular difference. When the vector angular difference in the ground truth becomes larger, we assign a larger weight to the point, which makes the network more attentive to significantly varying map topology. The vector direction difference loss $L_{i,j}^{vec}$ is:

$$L_{i,j}^{vec} = \sum_{i=0}^{N_i} \sum_{j=0}^{N_p-1} (1 - \cos \theta_{i,j}^{pre}) * W_{i,j}^{vec} + \sum_{i=0}^{N_i} \sum_{j=1}^{N_p} (1 - \cos \theta_{i,j}^{pre}) * W_{i,j}^{vec} \quad (5)$$

We use $1 - \cos \theta$ to adjust the interval of loss by $[0, 2]$. By summing the cosines of the angle differences of neighboring vector lines at each point, the loss captures more comprehensive information about the geometric topology of each point. The loss at the first and last points is the cosine of the angular difference between the unique neighboring vectors.

4 Experiments

4.1 Dataset and metric

The effectiveness of ADMap was verified on nuScenes [1] and Argoverse2 [23] datasets. The nuScenes dataset includes 40,000 labeled data with keyframes

Table 1: Results of ADMap in nuScenes benchmark compared to each state-of-the-art method. We performs validation in both camera-only and multi-modal frameworks with 24 and 110 training epochs, respectively. ADMap uses MapTR as the baseline, while ADMapv2 uses MapTRv2 as the baseline. FPS is measured on NVIDIA RTX 3090 GPU with batch size of 1. 'C' denotes the use of camera, and 'L' denotes the use of lidar. Bolding indicates best performance. † represents the addition of EA-LSS [10], CBGS [30], multi-task training and detection pre-training.

Model	Modality	Backbone	Epoch	AP_{div}	AP_{ped}	AP_{bou}	mAP	FPS
HMapNet [11]	C	EB0	30	27.7	10.3	45.2	27.7	0.9
HMapNet [11]	C & L	EB0 & PP	30	16.3	29.6	46.7	31.0	0.5
VectorMapNet [16]	C	R50	110	42.5	51.4	44.1	46.0	2.2
VectorMapNet [16]	C & L	R50 & PP	110	48.2	60.1	53.0	53.7	-
PivotNet [4]	C	R50	24	56.2	56.5	60.1	57.6	6.7
StreamMapNet [27]	C	R50	30	56.9	55.9	61.4	58.1	14.2
BeMapNet [21]	C	R50	30	62.3	57.7	59.4	59.8	4.2
MapTR [14]	C	R50	24	51.5	46.3	53.1	50.3	15.1
MapTR [14]	C & L	R50 & SEC	24	55.9	62.3	69.3	62.5	6.0
ADMap	C	R50	24	56.2	49.4	57.9	54.5	14.8
ADMap	C & L	R50 & SEC	24	66.6	63.3	74.0	68.0	5.8
ADMap	C & L	R50 & SEC	110	66.5	71.2	76.9	71.5	-
MapTRv2 [15]	C	R50	24	59.8	62.4	62.4	61.5	14.1
ADMapv2	C	R50	24	61.9	63.5	63.3	62.9	14.8
MapTRv2 [15]	C & L	R50 & SEC	24	65.6	66.5	74.8	69.0	5.8
ADMapv2	C & L	R50 & SEC	24	68.2	69.0	75.2	70.8	5.8
ADMapv2	C & L	R50 & SEC	110	67.7	73.8	76.6	72.7	-
ADMapv2†	C & L	R50 & SEC	24	83.0	80.2	84.8	82.8	5.7

Table 2: Results of ADMap in Argoverse2 benchmark compared to each state-of-the-art method. ADMap uses MapTR as the baseline, while ADMapv2 uses MapTRv2 as the baseline. FPS is measured on NVIDIA RTX3090 GPU with batch size of 1. 'C' denotes the use of camera, and 'L' denotes the use of lidar. * indicates the replicated results in the same framework.

Model	Modality	Backbone	Epoch	AP_{div}	AP_{ped}	AP_{bou}	mAP	FPS
HMapNet [11]	C	EB0	-	13.1	5.7	37.6	18.8	-
VectorMapNet [16]	C	R50	-	38.3	36.1	39.2	37.9	-
MapTR* [14]	C	R50	6	65.5	56.6	61.8	61.3	13.0
ADMap	C	R50	6	68.9	60.3	64.9	64.7	12.6
ADMap	C & L	R50 & SEC	6	75.5	69.5	80.5	75.2	8.9
MapTRv2 [15]	C	R50	6	62.9	72.1	67.1	67.4	12.1
ADMapv2	C	R50	6	72.4	64.5	68.9	68.7	12.6
ADMapv2	C & L	R50 & SEC	6	76.2	72.8	81.5	76.9	8.9

sampled at 2 HZ. It consists of 6 surround view cameras and 1 lidar with a FOV of 360 degrees. Argoverse2 comprises 1000 annotated multi-modal data sequences. The dataset includes high-resolution images from 7 surround view cameras and 2 stereo cameras, as well as LIDAR point cloud and attitude-aligned maps.

Table 3: Ablation experiments for each module. IIA for instance interactive attention, MPN for multi-scale perception neck, and VDDL for vector direction difference loss.

IIA	MPN	VDDL	AP_{div}	AP_{ped}	AP_{bou}	mAP
✗	✗	✗	62.3	56.2	69.1	62.5
✓	✗	✗	63.3	59.5	73.3	65.4
✗	✗	✓	64.8	56.8	72.2	64.6
✓	✓	✗	66.0	62.9	73.6	67.3
✓	✗	✓	65.6	59.7	74.2	66.5
✓	✓	✓	66.6	63.3	74.0	68.0
Improvment			+4.3	+7.1	+4.9	+5.5

To ensure a fair evaluation of ADMap, we classify the map elements into three categories: lane lines, road boundaries and pedestrian crossings. We use Average Precision (AP) to evaluate the quality of the map construction, and using the sum of the chamfer distances of the predicted and real point sequences to determine whether they match or not. Chamfer distance thresholds are set to $[0.5, 1.0, 1.5]$, and we calculate the AP under these three thresholds and take the average as the metric.

4.2 Implementation details

ADMap is trained using 8* NVIDIA Geforce RTX A100 GPUs. The batch size is set to 8, the learning rate is set to 6e-4, and the AdamW optimizer as well as the cosine annealing schedule are used. The point cloud range is set to $[-15.0, -30.0, -5.0, 15.0, 30.0, 3.0]$, and the voxel size is set to $[0.15, 0.15, 0.2]$. We use ResNet50 [8] as the camera backbone and SECOND [25] as the lidar backbone. The number of input channels of MPN is 256 and the number of output channels is $[512, 512, 512]$. In the loss function, the weights of L1 Loss and VDDL are set to 5.0 and 1.0, respectively. ADMapv2 increases the Auxiliary Dense Prediction Loss and Auxiliary One-to-Many Set Prediction Loss from MapTRv2 [15]. To speed up training, the replication multiplier in the Auxiliary One-to-Many Set Prediction Loss is set to 3 (6 in MapTRv2), which decreases the model performance.

4.3 Comparative experiments

nuScenes Table 1 reports the metrics of ADMap and the state-of-the-art methods on the nuScenes dataset. It is evident that ADMap outperforms the baseline in both camera-only and multimodal frameworks. In the camera-only framework, ADMap improves 4.2% compared to MapTR, ADMapv2 improves 1.4% compared to MapTRv2. In the multimodal framework, ADMap improves by 5.5% compared to MapTR and ADMapv2 improves by 1.8% compared to MapTRv2, with a maximum accuracy of 72.7%, which achieves the best performance in this benchmark.

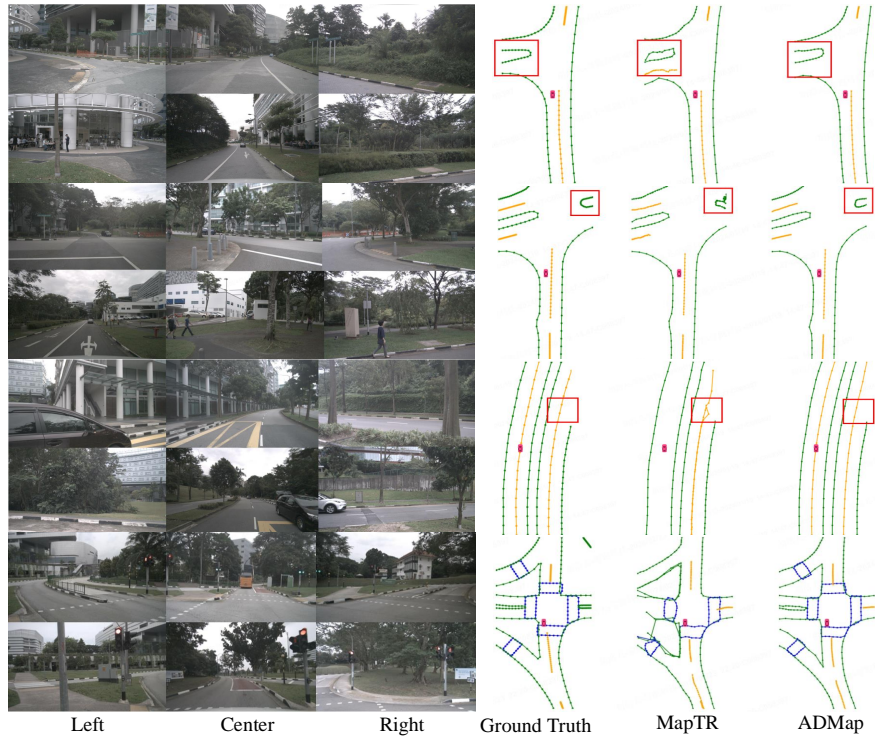


Fig. 5: Visualization results of the nuScenes dataset. Areas of discrepancy are indicated by red boxes. ADMap effectively reduces jitter within instances.

In terms of speed, ADMap demonstrate excellent performance. ADMap significantly improves model performance compared to the baseline, with only a slight decrease in FPS of 0.3 and 0.2. It is worth noting that ADMapv2 not only improves performance but also enhances inference speed compared to the baseline. The inference latency of the camera-only framework is reduced from 70.9ms to 67.6ms. In the multimodal framework, the speed advantage of ADMapv2 cannot be realized due to the model itself has a large latency.

Argoverse2 Table 2 reports the metrics of ADMap and the state-of-the-art methods on the Argoverse2 dataset. In the camera-only framework, ADMap and ADMapv2 improved by 3.4% and 1.3% compared to the baseline, respectively. In the multimodal framework, ADMap and ADMapv2 achieved the best performance in the benchmark with 75.2% and 76.9%. Regarding inference speed, the latency of ADMapv2 decreased by 3.3ms compared to MapTRv2.

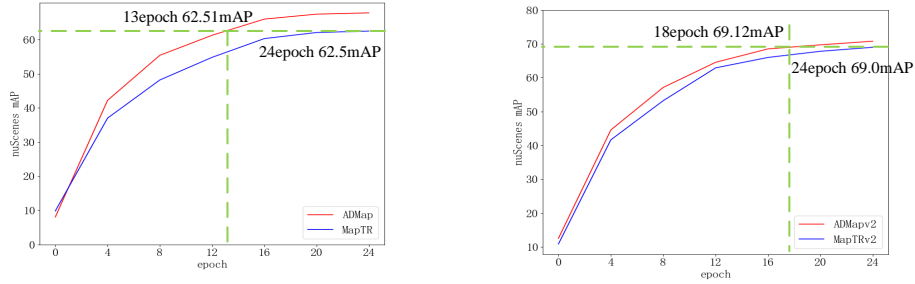


Fig. 6: ADMap, MapTR, ADMapv2, and MapTRv2 convergence curves in the nuScenes dataset. ADMap and ADMapv2 exceed their baseline best performance at epoch 13 and epoch 18, respectively

Table 4: Impact of different self-attention on performance. DSA denotes Decoupled self-attention, IIA denotes instance interactive attention. FPS is measured on NVIDIA RTX3090 GPU with batch size of 1.

Model	AP_{div}	AP_{ped}	AP_{bou}	mAP	FPS
Vanilla	62.3	56.2	69.1	62.5	6.0
+DSA [15]	64.2	57.8	70.2	64.1	5.8
+IIA(Our)	63.3	59.5	73.3	65.4	5.9

Table 5: Impact of the neck structure on the model performance. SECONDNeck denotes the neck structure in SECOND, MPN denotes multi-scale perception neck.

Model	AP_{div}	AP_{ped}	AP_{bou}	mAP	FPS
Vanilla	62.3	56.2	69.1	62.5	6.0
+SECONDNeck [25]	62.8	58.4	70.2	63.7	5.9
+MPN(Our)	65.0	59.6	69.4	64.5	5.9

Table 6: Impact of VDDL weights on performance

Weight	AP_{div}	AP_{ped}	AP_{bou}	mAP
0.0	51.5	46.3	53.1	50.3
0.5	54.6	49.3	53.8	52.5
1.0	55.9	48.4	55.6	53.3
2.0	55.5	45.4	54.2	51.7
3.0	55.4	45.2	55.1	51.9

4.4 Ablation experiment

Table 3 shows the results of the ablation experiments for each module in the nusenes benchmark. To MapTR, we added MPN/IIA and VDDL. It can be observed that the mAP increased by 2.9% with the addition of IIA and 2.1% with the addition of VDDL. The mAP increased by 4.8% after adding both IIA

Table 7: Ablation experiments on the number of downsampled layers in MPN.

num layer	AP_{div}	AP_{ped}	AP_{bou}	mAP	FPS
1	66.5	68.6	74.2	69.8	6.0
2	67.9	68.5	74.5	70.3	5.8
3	68.1	68.3	74.9	70.5	5.3

Table 8: Comparison of ACD/ARD/AJP scores between ADMap and ADMapv2 and their respective baselines.

model	MapTR	ADMap(Our)	MapTRv2	ADMapv2(Our)
ACD ↓	0.597	0.518	0.560	0.422
ARD ↓	7.866	5.358	5.950	3.484
AJP ↓	6.76	4.08	4.53	2.44

and MPN, and by 4.0% after adding both IIA and VDDL. Finally, our proposed method greatly improves the ability of the baseline to improve its mAP by 5.5%. Table 4 shows the comparison between IIA and Decoupled Self-Attention (DSA) and their respective effects. It can be seen that IIA achieves better results. DSA improves the mAP of the baseline by 1.6%, while IIA improves the mAP of the baseline by 2.9%. In addition, IIA also outperforms DSA in terms of speed, and its latency is reduced by 2.9ms compared to DSA. Table 5 shows how the performance is affected by adding neck. The mAP increased by 1.2% with the addition of the neck of SECOND [25], and the model mAP increased by 2.0% with the addition of MPN, without increasing the inference time.

Table 6 reports the impact of VDDL’s weights in the nuScenes benchmark on the performance of ADMap for the camera-only framework. The best performance is achieved when the weights are set to 1.0. Table 7 reports the effect of the number of downsampling layers of the MPN on the performance of ADMapv2 in the nuScenes benchmarks. To balance speed and performance, we set the number of downsampling layers to 2, as increasing the number of downsampling layers can slow down model inference.

4.5 Visualization

Figure 5 show comparison of the visualizations between ground truth, MapTR, and ADMap in the nuScenes benchmark. The figure shows that the lane lines, pedestrian crossings, and road boundaries in MapTR are distorted and deformed to some extent. This distortion can affect subsequent planning, control, and other tasks. In contrast, ADMap effectively mitigates point jitter within vector instances and predicts map elements more accurately.

4.6 Analysis

Anti-jitter effect evaluation We defined three evaluation matrix for Anti-jitter effect. (1)**Average chamfer distance (ACD)**: we select all predicted

instances whose sum of the internal chamfer distance is less than 1.5. We then calculate the average chamfer distance between these predicted instances (true positive instances) and the real instances. (2) **Average radian distance** (ARD): the radian distance is defined as the sum of the radians of the angles between all segments in each positive sample and the true instance segments, followed by an average over all positive samples. (3) **Average number of jitter points** (AJP): jitter point is defined as a point where the angle between the front and back line segments is greater than a certain threshold (30°) and where the front and back line segments of its corresponding true point are less than a certain threshold (5°). Table 8 displays the results of these three evaluation matrix, the lower score means the high ability for anti-jitter. The experiment result demonstrate that our proposed ADMap effectively mitigates the problem of inaccurate point sequence prediction.

Figure 6 shows a comparison of the convergence curves of ADMap, MapTR, ADMapv2, and MapTRv2 in the nuScenes benchmark. By the 13th epoch, mAP of ADMap exceeds the optimal performance of MapTR, while ADMapv2 reaches 69.12 mAP by the 18th epoch, exceeding the optimal performance of MapTRv2.

5 Conclusions

ADMap is an effective and efficient vectorized HD map construction framework that mitigates the problem of map topology distortion caused by instance point jitter. We address this issue through three modules: multi-scale perception neck, instance interactive attention, and vector direction difference loss. Extensive experiments have demonstrated that our proposed method can achieve the best performance in nuScenes and Argoverse2 benchmarks and high efficiency. We believe that ADMap can contribute to the community in advancing research on vectorized HD map construction tasks for better development in areas such as autonomous driving.

Acknowledgements

Up to this point, the conclusion of the proposal is aligned with an expression of profound gratitude. I would like to express my sincerest gratitude to all my colleagues, associates, and professional contacts who provided support and assistance in the completion of this paper.

I would like to extend my gratitude to Zhejiang Leapmotor Technology CO., LTD. for their invaluable sponsorship of this research project. The financial support provided by this company has been instrumental in ensuring the success of this study, particularly in terms of data collection, experiments and thesis writing. I would also like to thank all the participants of this study for their dedication and commitment, without which this study would not have been possible. I would like to reiterate my gratitude to all those who have helped me in this endeavour.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
2. Chen, L., Sima, C., Li, Y., Zheng, Z., Xu, J., Geng, X., Li, H., He, C., Shi, J., Qiao, Y., Yan, J.: Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In: European Conference on Computer Vision (ECCV) (2022)
3. Chen, S., Cheng, T., Wang, X., Meng, W., Zhang, Q., Liu, W.: Efficient and robust 2d-to-bev representation learning via geometryguided kernel transformer. arXiv preprint arXiv:2206.04584 (2022)
4. Ding, W., Qiao, L., Qiu, X., Zhang, C.: Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3672–3682 (2023)
5. Gao, W., Fu, J., Shen, Y., Jing, H., Chen, S., Nanning, Z.: Complementing onboard sensors with satellite maps: A new perspective for hd map construction. ICRA (2024)
6. Garnett, N., Cohen, R., Pe’er, T., Lahav, R., Levi, D.: 3d-lanenet: End-to-end 3d multiple lane detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
7. Guo, Y.G., Chen, G., Peitao, Z., Weide, Z., Miao, J., Wang, J., Choe, T.E.: Gen-lanenet: A generalized and scalable approach for 3d lane detection (2020)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
9. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., A., K.: Fiery: Future instance segmentation in bird’s-eye view from surround monocular cameras. ICCV (2021)
10. Hu, H., Wang, F., Su, J., Wang, Y., Hu, L., Fang, W., Xu, J., Zhang, Z.: Ea-lss: Edge-aware lift-splat-shot framework for 3d bev object detection. arXiv preprint arXiv:2303.17895 (2023)
11. Li, Q., Wang, Y., Wang, Y., Zhao, H.: Hdmapnet: An online hd map construction and evaluation framework (2021)
12. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multicamera images via spatiotemporal transformers. ECCV (2022)
13. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multicamera images via spatiotemporal transformers. ECCV (2022)
14. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: Maptr: Structured modeling and learning for online vectorized hd map construction. In: International Conference on Learning Representations (2023)
15. Liao, B., Chen, S., Zhang, Y., Jiang, B., Zhang, Q., Liu, W., Huang, C., Wang, X.: Maptrv2: An end-to-end framework for online vectorized hd map construction. arXiv preprint arXiv:2308.05736 (2023)
16. Liu, Y., Yuantian, Y., Wang, Y., Wang, Y., Zhao, H.: Vectormapnet: End-to-end vectorized hd map learning. In: International conference on machine learning. PMLR (2023)
17. Liu, Z., Chen, S., Guo, X., Wang, X., Cheng, T., Zhu, H., Zhang, Q., Liu, W., Zhang, Y.: Vision-based uneven bev representation learning with polar rasterization and surface estimation. arXiv preprint arXiv:2207.01878 (2022)

18. Neven, D., Brabandere, B.D., Georgoulis, S., Proesmans, M., Gool, L.V.: Towards end-to-end lane detection: an instance segmentation approach (2018)
19. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018)
20. Phillion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. *ECCV* (2020)
21. Qiao, L., Ding, W., Qiu, X., Zhang, C.: End-to-end vectorized hd-map construction with piecewise bezier curve. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 13218–13228 (June 2023)
22. Wang, S., Jia, F., Liu, Yingfei an Zhao, Y., Chen, Z., Wang, T., Zhang, C., Zhang, X., Zhao, F.: Stream query denoising for vectorized hd map construction. *arXiv preprint arXiv:2401.09112* (2024)
23. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)* (2021)
24. Xie, Z., Pang, Z., Wang, Y.X.: Mv-map: Offboard hd-map generation with multi-view consistency. *arXiv* (2023)
25. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection (2018)
26. Yu, J., Zhang, Z., Xia, S., Sang, J.: Scalablemap: Scalable map learning for online long-range vectorized hd map construction. *CoRL* (2023)
27. Yuan, T., Liu, Y., Wang, Y., Wang, Y., Zhao, H.: Streammapnet: Streaming mapping network for vectorized online hd map construction. *arXiv preprint arXiv:2308.12570* (2023)
28. Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., Cai, D.: Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021)
29. Zhou, B., Krahenbuhl, P.: Cross-view transformers for real-time map-view semantic segmentation. *CVPR* (2022)
30. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492* (2019)