Technical Report for Online HD Map Construction in CVPR 2023 Autonomous Driving Challenge

Weijie Luo[1,2,*] Zihao Liu[1,2,*] Hongyu Luo[1,2,*] Ji Zhao[2]
Ningyi Xu[1,2]
Shanghai Jiao Tong University[1] Huixi Technology[2]
vijayluo@sjtu.edu.cn 15534445257@sjtu.edu.cn wsylhy20200912@sjtu.edu.cn
zhaoji84@gmail.com xuningyi@sjtu.edu.cn

Abstract

This technical report presents the third winning model for Online HD Map Construction, a task newly introduced in CVPR 2023 Autonomous Driving Challenge. Compared to conventional lane detection, the constructed HD map provides more semantics information with multiple categories. Given inputs from onboard sensors (cameras), the goal is to construct the complete local HD map. We address this problem by proposing MapSwin which is based MapTr, but we have come up with some more effective ways to improve the performance of the model, including more powerful backbone, better View Transformation, more effective data augmentation and effective supervision. As a result, we achieved the overall third place in this challenge.

Figure 1. High-definition (HD) map is composed of instance-level vectorized representation of map elements (pedestrian crossing (blue), lane divider (red), road boundaries (green), etc.). Given inputs from onboard sensors (cameras), the goal is to construct the complete local HD map.

1. Introduction

Constructing HD maps is a central component of autonomous driving. In the conventional approach, HD maps are typically constructed offline using SLAM-based methods. This process involves a complex pipeline and results in high maintenance costs. However, there has been a growing interest in online HD map construction. This approach involves constructing maps in real-time around the ego-vehicle using sensors mounted on the vehicle itself. By adopting this method, the need for offline human efforts is eliminated. Online HD map construction task aims to dynamically construct the local semantic map based on onboard sensor observations. Compared to lane detection, our constructed HD map provides more semantics information of multiple categories. Vectorized map representation are adopted to deal with complicated and even irregular road structures. The goal of the online HD map construction task is to construct the local HD map from onboard sensor observations. A local HD map can be described by a set of map elements with different categories, e.g. road divider, road boundary and pedestrian crossing as shown in Fig 1.

To solve this challenge, we propose a new model named MapSwin which is based MapTr. Additionally, we also perform ablation study to investigate the effect of our different module as shown in Sec. 3.4.

2. Our Model

In this section, we present our approach to deal with online HD map construction which is introduced in CVPR 2023 autonomous driving challenge. The overall architecture of our model is shown in Fig. 2.

2.1. Backbone

High-definition (HD) map is the high-precision map specifically designed for autonomous driving, composed of instance-level vectorized representation of map elements

*These authors contributed equally to this work

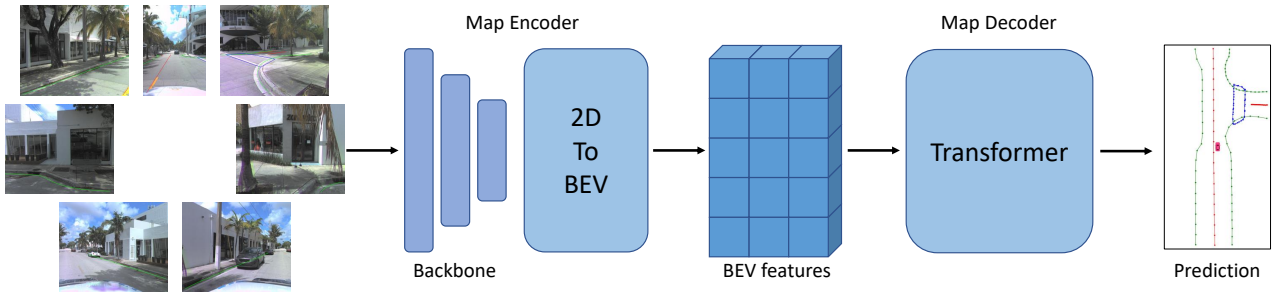

Figure 2. The overall architecture of our method which is based on MepTr [4] The sensor input is transformed into a unified Bird's Eye View (BEV) representation by the map encoder. This encoding process effectively converts the sensor data into a standardized map format. In terms of decoding, the map decoder utilizes a hierarchical query embedding scheme to explicitly encode the different elements of the map.

(pedestrian crossing, lane divider, road boundaries, etc.) In this challenge, the input data is surrounding cameras images, so we need a strong backbone to obtain 2D image features. During inference, we feed multi-camera images to the backbone network (e.g., ResNet-50 and swin-base), and obtain the features $F_i = \{F_i\}_{i=1}^{N_{\text{view}}}$ of different camera views, where F_i is the feature of the i-th view, N_{view} is the total number of camera views. In different versions of MapTr [4], they experimented with different backbones, such as ResNet-50, Swin-tiny [5], Swin-base [5], etc. Based on this, we tried other backbones, such as ConvNeXt [6].

2.2. View Transformation

MapTR [4] uses GKT [1] as the default 2D-to-BEV transformation module. We replace GKT with the Spatial Cross-Attention layer proposed by BEVFormer [3]. Furthermore, we delete the Temporal Self-Attention layer which is used to incorporating historical BEV features. After the multi-scale 2d image features are obtained, they are converted into BEV features through multiple spatial encoder layer.

2.3. Head

Following MapTR, we use a hierarchical query embedding scheme to explicitly encode each map element. Specifically, we define a set of instance-level queries and a set of point-level queries shared by all instances. Each map element corresponds to a set of hierarchical queries which is the sum of the first two.

The head contains several cascaded decoder layers which update the hierarchical queries iteratively. In each decoder layer, we adopt MHSA to make hierarchical queries exchange information with each other. We then adopt Deformable Attention [7] to make hierarchical queries interact with BEV features, following BEVFormer. Each query predicts the 2-dimension normalized BEV coordinate of the reference point. We then sample BEV features around the reference points and update queries.

The prediction head consists of a classification branch and a point regression branch. The classification branch predicts instance class score. The point regression branch predicts the positions of the point sets.

2.4. Loss

The loss function is composed of four parts, classification loss, point2point (p2p) loss, edge direction loss and point2line (p2l) loss. The first three are inspired by MapTR. The point2line loss is inspired by VMA [2]. The total loss is formulated as:

$$L = \lambda L_{cls} + \alpha L_{p2p} + \beta L_{dir} + \gamma L_{p2l} \tag{1}$$

$$L_{cls} = \sum_{i=0}^{N-1} L_{Focal}(\hat{p}_{(i)}, c_i) \tag{2}$$

$$L_{p2p} = \sum_{i=0}^{N-1} \mathbb{I}_{\{c_i \neq \emptyset\}} \sum_{i=0}^{N_v-1} D(P, Q) \tag{3}$$

$$L_{p2l} = \sum_{i=0}^{N-1} \mathbb{I}_{\{c_i \neq \emptyset\}} \sum_{i=0}^{N_v-1} \frac{1}{2}[D(P, L_{left}) + D(P, L_{right})] \tag{4}$$

where λ, α, β and γ are the weights for balancing different loss terms. P matches Q in Hungarian match. L_{left} and

L_{right} are the two adjacent edges of Q. The direction loss is similar to MapTR.

The p2p loss is the L2 distance between paired points, while the p2l loss is the point-to-line distance between predicted point and the two adjacent edges.

2.5. Data augmentation

During the training process, we also apply some classical data augmentation methods.

To make the model more robust to different orientations of objects or scenes of low visibility it might encounter during inference, we apply geometric and photometric distortion. We tried some geometric methods such like flip, resize and rotation. But at last we only apply flip as a geometric distortion because of its effectiveness in our experiment.

We also find that the number of instances containing pedestrian crossing is smaller than that of instances containing the other two classes. So we apply CBGS (Class-balanced Grouping and Sampling) methods on the training dataset to make it more balanced.

2.6. Model ensemble

In order to strengthen the generalization ability of the model, we adopted the strategy of model ensemble. The ensemble method is simple but effective. Given N well-trained models, we collect 50 predicted elements from each model, which result in 50N predicted elements for a single scene. To remove duplicate elements among there predicted elements, we adopted non-maximum-suppression (NMS) on the element points. Each predicted element has its own confidence score and we tend to preserve those elements who have higher confidence. We used chamfer distance as similarity metric in NMS to remove duplicate elements. To be more specific, We always choose the element with highest confidence and remove these elements who has small chamfer distance to it. We repeat this process until no element is deleted.

We tried two ways to set chamfer distance threshold, static way and dynamic way. Static way means the threshold is set to be a non-change value during the ensemble process, which is simple to realize and is mainly used by us in this competition. Dynamic way means the threshold would be changed according the length of the two elements. We believe that a bigger threshold will describe the similarity better between two relatively longer elements. Based on this belief, We designed the following calculation method for threshold δ as shown in Eq. 5

$$\delta = \max(2, \min(0.8, \frac{L-5}{60-5} \times 1.2 + 0.8)) \qquad (5)$$

This simple ensemble method boosted the performance of our model remarkably, which can be found in Sec.3.4.

	mAP	AP_P	AP_D	AP_B
MACH 1^{st}	83.50	86.66	81.54	82.29
MapNeXt 2^{nd}	73.65	68.94	76.66	75.34
MapSwin 3^{nd}	73.39	70.37	75.08	74.73
Baseline	42.11	35.95	50.11	40.26

Table 1. Top 3 entries on the test leaderboard of online hd map construction track of End-to-End Autonomous Driving workshop. AP_P denotes AP of pedestrian crossing, AP_D of divider and AP_B of boundary

3. Experiments

3.1. Dataset

The challenge dataset is built on top of the Argoverse2 dataset. Each segment is tens of seconds long. There are seven images for each sample. The map elements have three classes, including boundaries, dividers, and pedestrian crossing.

3.2. Implementation Details

We adopt AdamW optimizer and cosine annealing schedule. We use ResNet50 as the backbone for ablation study. For test set, we use Swin-B as the backbone. For all experiments except ablation study , λ is set to 2, α is set to 5, β is set to $5e^{-3}$ and γ is set to 2 during training. we set the number of instance-level queries and point-level queries to 100 and 20 respectively. And we set the size of each BEV grid to $0.3m$ and stack 6 transformer decoder layers. We train it with a total batch size of 16, a learning rate of $6e^{-4}$, learning rate multiplier of the backbone of 0.1, and total epoch of 40.

3.3. Results

Our best results was achieved by applying static ensemble method (mentioned in Sec. 2.6) on two well-trained models, where the variable is the point number of each element (20 and 40).

Table. 1 shows top 3 models on the test leaderboard of online HD map construction track of End-to-End Autonomous Driving workshop. The HDMapNet is the official baseline model in the challenge. Our model has a huge performance improvement compared with baseline. And we ranked the 3^{nd} place at last and has a tiny gap between the 2^{nd} MapNeXt.

3.4. Ablation study

We study the effectiveness of all components of MapSwin. The key components include data augmentation (DA), model ensemble (EN), point2line loss (p2l) and element number (e-num). As shown in Table. 2, we can find

p2l	e-num	DA	EN	mAP
×	50	×	×	0.4971
×	100	×	×	0.5025
✓	50	×	×	0.5124
✓	50	✓	×	0.5131
✓	50	×	✓	0.5250

Table 2. Effects of different components in MapSwin framework. The components include data augmentation(DA), model ensemble(EN), point2line loss(p2l) and element number(e-num).

that all applied components contribute to the final performance.

4. Conclusion

In our model, we enhance the performance of MapTR-based model by using more powerful backbone, adding effective supervision, applying effective ways of data augmentation and model ensemble. At the same time, we also utilize Spatial Cross-Attention layer to replace GKT in MapTr to get better performance. Detailed experimental results and ablation study also confirm the effectiveness of our model. As a result, we achieved the overall third place in this challenge and we will make more new attempts at HD map construction.

References

[1] 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. 2

[2] Shaoyu Chen, Yunchi Zhang, Bencheng Liao, Jiafeng Xie, Tianheng Cheng, Wei Sui, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene. *arXiv preprint arXiv:2304.09807*, 2023. 2

[3] 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 *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022. 2

[4] 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. 2

[5] 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 (ICCV)*, pages 10012–10022, October 2021. 2

[6] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, June 2022. 2

[7] 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. 2