Technical Report on Track 1 OpenLane Topology

1. Introduction

In the Autonomous Driving Challenge (OpenLane Topology Track), the goal is to surpass conventional lane detection by directly detecting the 3D center-line. The objective is to accurately identify 3D center-lines as an abstraction of the scene and establish the topology between 3D center-lines and traffic elements. This topology is crucial for effective planning and routing in autonomous driving systems. To achieve reliable performance on this track, several factors need to be taken into account. These factors include domain shift, the presence of unlabeled samples, imbalanced data distribution, network structure, and the format of the supervised labels provided. The dataset used for evaluation consists of a test set and a validation set. The data statistics of the test set are presented in Table 7. These statistics provide insights into the characteristics of the data and can be used to analyze the performance of the model on different subsets of the dataset.

To meet the goal of this challenge, it is necessary to leverage state-of-the-art deep models and effective topology modeling approaches. In our method, we extensively investigate the effect of different deep models (ResNet, ViT [4], Res2Net [5]), data augmentation (image flip, image crop, GridMask [3]), pre-training weights (Imagenet-pretrained, depth-pretrained, InternImage [11]), semi-supervised methods (Consistent-Teacher [12], Soft-teacher [14]). Finally, with the model ensemble techniques, we achieve the second place with Specifically, the detalied results are 42% mean center-line detection accuracy, 64% mean traffic element detection accuracy, 7% center-line topology prediction accuracy, 30% topology prediction accuracy between center-lines and traffic elements.

2. Method

2.1. Domain-oriented Pre-training

To bridge the gap between the image domains of the Imagenet dataset and the Openlanev2 dataset, we made efforts to obtain pre-training weights that were more closely aligned with our task. One approach we explored was utilizing the depth map of existing lane detection datasets as supervised information for training.

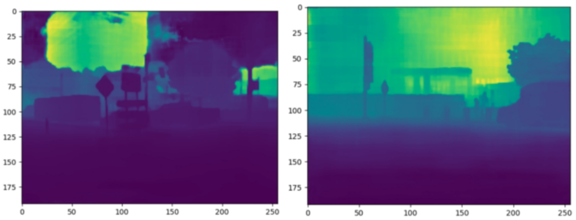

Figure 1. Depth prediction results of the depth-pretrained models.

By incorporating the depth map as a source of supervised information, we aimed to enhance the model's ability to understand the spatial relationships and depth cues within the scene. This pre-training strategy helped the model learn features and representations that were more relevant to the Openlanev2 dataset, potentially improving its performance on the task. Additionally, we also investigated the use of pre-training weights from models like InterImage. These models have been developed specifically for large-scale training and have demonstrated effectiveness in learning meaningful representations without the need for extensive labeled data. Among these pre-trained weights, the depth-pretrained weight can bring more significant performance boost of around 3% for the center-line detection task. Examples of the predicted depth maps on Openlanev2 are shown in Fig. 1.

2.2. Techniques for Center-line Detection

Different Backbones. we look into the widely studied model structures, specifically, ResNet-50, ResNet-101, ViT-Small, ViT-Base, ViT-Large are tested. The results are shown in Table 1. One can see that strong transformer based backbones (i.e., ViT-B) show obvious advantages in terms of accuracy on both center-line and traffic element detection tasks by exploiting inherent capabilities in capturing fine-grained details and long-range relationships.

Different View Transformer. During our experimentation, we explored different view transformers to obtain robust representations of the Bird's Eye View (BEV) perspective. The view transformers we considered included conventional non-parametric Inverse Perspective Mapping (IPM), PersFormer [2], BEVFormer [8], and BEVFormerV2 [15]. Since IPM relies on the strict assumption of a flat ground, it limits its performance in more complex and varied real-world scenarios. Thus IPM exhibited the worst performance among the tested view transformers. To overcome the limitations of

Dataset	Validation Set				
Method	ResNet-50	ResNet-101	ViT-S	ViT-B	ViT-L
DET_l	25%	31%	33%	37%	37%
DET_t	51%	53%	57%	56%	58%

Table 1. Results of combing different backbone networks.

IPM, we explored more advanced view transformers. PersFormer is a transformer-based method specifically designed for perspective transformation and feature extraction. BEVFormer is another transformer-based approach that focuses on extracting high-quality features from the Bird's Eye View. Both PersFormer and BEVFormer offer improvements over IPM in terms of capturing spatial information and handling more complex scenes.

Among the tested view transformers, BEVFormerV2 demonstrated the best performance. BEVFormerV2 is an enhanced version of BEVFormer, specifically tailored for extracting high-quality features from the Bird's Eye View perspective. By leveraging the power of transformers and incorporating improvements, BEVFormerV2 surpassed the other view transformers in terms of accuracy and effectiveness in representing the BEV scene.

Different Center-line Detection Head. During our experimentation, we explored two different methods for center-line prediction: Deformable DETR and BEV-LaneDet [10]. After comparing their performances, we observed that BEV-LaneDet achieved approximately 3% higher results compared to Deformable DETR when using ResNet-50 as backbone network, which demonstrates that it can effectively harness the distinctive characteristics of the BEV domain and incorporates strategies tailored specifically for lane-line detection.

Different Annotations. During our experimentation, as shown in Table. 4, we investigated two different forms of annotation for center-line modeling: Bezier curves and uniformly sampled points. Through our evaluation, we found that uniformly sampled points yielded better results compared to Bezier curves. Bezier curves provide a more abstract representation for modeling center-lines, as they require the prediction of curve coefficients. However, we observed that the model had difficulty accurately predicting the Bezier curve coefficients in unknown scenarios. This limitation might be attributed to the complexity of capturing the precise characteristics of Bezier curves and the challenges associated with generalizing the predictions. In contrast, uniformly sampled points offered a more intuitive approach for center-line annotation. This method allowed us to sample points along the center-line, providing potential indicative information for the network to predict the center-line accurately. By adopting this approach, the model could focus on learning and understanding the relationships between the sampled points, leading to improved performance in center-line prediction.

Dataset	Validation Set			
Method	IPM	PersFormer	BEVFormer	BEVFormerV2
DET_l	14%	23%	25%	26%

Table 2. Results of combing different view transformers with ResNet-50 as backbone.

Dataset	Validation Set	
Method	Deformable DETR	BEV-LaneDet
DET_l	25%	28%

Table 3. Results of combing different center-line detection heads with ResNet-50 as backbone.

Dataset	Validation Set			
Method	3-Bezier	5-Bezier	10-Points	20-Points
DET_l	12%	14%	21%	25%

Table 4. Results of different forms of annotations.

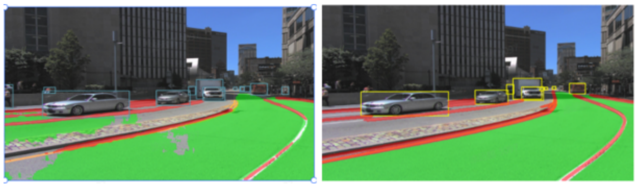

Figure 2. Predicted driveable area and 2D lane detection results of YoloP (left) and YoloPv2 (right) on Openlanev2.

2.3. Multi-task Training

To fully leverage the benefits of existing large-scale pretrained models, we employ SAM [7]/YoloP [13]/YoloPv2 [6] to locate driveable areas and 2D lane line results. Since the result of YoloP is inferior to YoloPv2, we ensemble the results of YoloPv2 and SAM to get the annotations. This process allows us to obtain additional annotation information that aids in the localization of center-lines. To provide explicit cueing information for center-line prediction, we utilize the driveable area labels and lane line labels obtained from the previous steps as inputs to the common decoder and DETR header, respectively for the supervised training process. Some of the prediction results for driveable areas and 2D lane are shown in Fig. 2.

The driveable area labels indicate the regions within the image that are considered safe and drivable. By incorporating these labels into the decoder, we provide explicit cues and guidance for the model to focus on the driveable areas when predicting the center-lines. This information helps the model learn to associate the center-lines with the appropriate regions in the image, leading to more accurate and reliable center-line predictions. Similarly, the lane line labels obtained from the previous steps are used as inputs to the DETR headers. The presence of lane line labels serves as explicit cueing information for the model to detect and understand the lane lines in the scene. By incorporating this information during supervised training, we enable the model to learn the characteristics and patterns of lane lines, enhancing its ability to predict accurate center-lines. Specifically, by introducing the multi-task training, a relative improvement of around 4% in center-line prediction can be achieved.

2.4. Semi-supervised Training

By analyzing the dataset statistics, we observed a significant presence of images without traffic signs, where there are no traffic elements for 8,891 frames in 22,477 frames of the whole training set. These images lack any instances of traffic signs within the scene. Additionally, we discovered that there are a considerable number of unused traffic signs present in images captured from views other than the front-center view, which contain abundant discriminative information. Unfortunately, these images with unused traffic signs do not have corresponding supervised labels.

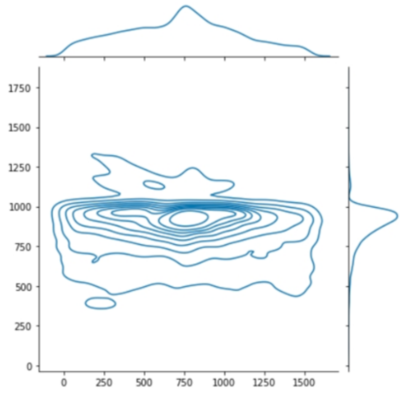

Figure 3. Statistics of the locations of traffic elements.

Addressing this issue would require careful consideration. One potential approach could involve leveraging unsupervised or weakly supervised learning techniques to make use of the abundant but unlabeled images with unused signs. By incorporating these images during training, the model can potentially learn to generalize and recognize traffic signs more effectively in various viewpoints. Thus we try to combine existing state-of-the-art semi-supervised methods including Consistent-Teacher, Soft-teacher. We empirically found that all of these semi-supervised training techniques are useful to boost test performance, where the Consistent-Teacher and Soft-teacher can bring a relative improvement of 3% and 7% in 2D object detection task respectively.

2.5. Techniques for Object Detection

Different 2D Object Detection Heads. First, we try different 2D object detection heads, including Faster-RCNN, DETR [1], Deformable DETR [17] and DAB-DETR [9] heads. The corresponding results are shown in Table. 5, where Deformable DETR head is better at handling object deformations and variations in scale by incorporating deformable attention mechanisms into the DETR framework. For DAB-DETR, it can achieve the best result by improving the query-to-feature similarity and modulating the positional attention map using the box width and height information.

Dataset	Validation Set			
Method	Faster-RCNN	DETR	Deformable DETR	DAB-DETR
DET_t	42%	48%	49%	51%

Table 5. Results of combing different object detection heads with ResNet-50 as backbone network.

Dataset	Validation Set	
Method	RFLA	QueryDet
DET_t	50%	52%

Table 6. Results of combing different small object detection methods.

Dataset	Test Set					
Metric	DET_l	DET_t	TOP_{ll}	TOP_{lt}	OLS	F-Score
Accuracy	42%	64%	7%	30%	47%	48%

Table 7. Best results on test set of Openlanev2.

Small Object Detection. Furthermore, based on our empirical analysis, we have observed that the majority of traffic signs in the dataset are relatively small targets. This characteristic poses a challenge as conventional detection methods may not be specifically optimized for effectively detecting small objects. To address this challenge, we have explored the combination of various small target detection methods. By leveraging these specialized techniques, we aim to enhance the model's ability to accurately detect and localize small traffic signs within the scene.

The integration of small target detection methods involves incorporating approaches such as RFLA [9], QueryDet [16]. By combining these small target detection methods, we can potentially improve the model's performance in accurately detecting and localizing traffic signs, even when they appear as small targets within the scene. This integration allows us to better capture the fine-grained details and spatial information crucial for effective traffic sign recognition. Specifically, the results of combining two methods with the baseline methods are shown in Table. 6.

Augmentation Strategy. Due to the lack of data on traffic elements, we further apply an augmentation strategy for object detection head. Specifically, throughout the training process, we employ a technique where we crop and store image patches containing traffic elements into an online memory bank. For images that do not initially contain any traffic elements, we randomly paste these cropped traffic element patches onto the images. This enables us to perform supervised training by utilizing the pasted patches. By utilizing the online memory bank and employing the technique of pasting traffic marker patches onto images, we can effectively augment our dataset and ensure comprehensive supervised training for traffic marker detection. This approach contributes to the model's improved performance and robustness in accurately identifying and localizing traffic markers in the autonomous driving challenge. In our efforts to enhance the generalization ability of the model, we explored additional techniques during the training process. Specifically, we incorporated random flip and GridMask operations to perturb the training of the model. For randomly flipping operation, we expose the model to a more diverse set of training samples, enabling it to learn invariant representations and improving its ability to handle variations in object orientations. By incorporating GridMask operations, we encourage the model to learn more robust and discriminative features, making it less sensitive to irrelevant details or distracting elements in the scene. This perturbation aids in enhancing the model's generalization capabilities and its ability to accurately detect and classify objects in varying conditions. Finally, we conducted an analysis of the traffic element locations in Fig. 3 and observed that the majority of traffic elements tended to appear in the middle and bottom regions of the image. Leveraging this observation, we devised a preprocessing technique to enhance the model's efficiency in handling logo detection. To reduce the area requiring positioning and focus on the relevant regions, we implemented a cropping strategy that targeted the sky area in the upper part of the image. By removing this portion, we minimized the unnecessary information and allocated more attention to the critical regions where the logos typically appeared. To ensure consistency in the aspect ratio of the image across all views, we added blank padding on both sides after the cropping process. This padding prevented any distortion or alteration of the image's original proportions. By maintaining a consistent aspect ratio, we enabled the model to process and analyze the cropped image effectively while reducing the computational burden. Specifically, all these augmentation strategies can bring an improvement of around 4% to the result of QueryDet in Table. 6.

3. Topology Prediction

To tackle the topology prediction task, we adopt a more appropriate structure, namely the Graph Convolution Network (GCN), instead of using several fully-connected layers to directly model the relationship between center-lines and traffic elements.In our experience, we have found that employing too many layers in the GCN can potentially lead to overfitting. Therefore, we have chosen to utilize a three-layer GCN for both of the topology heads.By using a GCN, we can effectively model the dependencies and interactions between the center-lines and traffic elements in a graph-like structure. The GCN leverages the graph's connectivity to propagate information and capture the complex relationships between different elements.

4. Training Process

During our experiments, we observed an interesting phenomenon when training the center-line prediction and traffic sign prediction together. Initially, the accuracy of the traffic sign prediction task increased, but then it started to decrease. We attribute this behavior to the relative simplicity of the traffic sign prediction task. As we continued training, the traffic sign prediction head tended to overfit to the training data, resulting in a decrease in its generalization ability and a subsequent decline in accuracy on the test set. To address this issue and ensure optimal performance, we made the decision to train the center-line and traffic sign prediction networks separately. This separation allowed us to mitigate the problem of overfitting in the traffic sign prediction task and maintain the generalization ability of the model.

Due to memory limitations, we adopted a strategy of training the center-line and traffic sign prediction networks independently. Subsequently, we utilized the prediction results and features obtained from both networks as inputs to the topology prediction headers. This approach enabled us to leverage the strengths of each network while avoiding potential interference and memory constraints that could arise from training all the structures together. By training the center-line and traffic sign prediction networks separately and utilizing their respective outputs for topology prediction, we were able to achieve improved performance and maintain the model's ability to generalize well to unseen data. This approach allowed us to address the challenges associated with overfitting and memory limitations, resulting in more accurate and reliable topology predictions for the Autonomous Driving Challenge.

5. Final Results

In our final submitted file, we have incorporated all the previously mentioned effective techniques to achieve the best performance. The finalized model architecture utilizes ResNet-101 as the backbone network, leveraging its strong feature extraction capabilities. To further enhance the model's performance, we load pre-trained weights that were specifically trained for depth-related tasks. In addition to the backbone network, we employ a Feature Pyramid Network (FPN) module to fuse the feature maps obtained from stages 1 to 4. This fusion enables the model to capture multi-scale representations, which are crucial for accurately detecting objects of varying sizes within the scene. The resulting feature representations from the FPN module are then fed into the BEVFormerV2 architecture. By utilizing this module, we can effectively extract discriminative features from the BEV views, further enhancing the model's ability to understand the scene from a top-down perspective. Moreover, We remove some duplicate lines by NMS between lines for the final center-line results. For the topology prediction module, a three-layer GCN module is utilized for both center-lines and traffic elements to model the topology structure. Results from two models optimized with different optimizers are ensembled as the final result. The final results of five metrics on the test set are shown in Table. 7.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

[2] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. *ArXiv*, abs/2203.11089, 2022.

[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *ArXiv*, abs/2001.04086, 2020.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[5] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang, and P. Torr. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):43, 2021.

[6] Cheng Han, Qichao Zhao, Shuyi Zhang, Yinzi Chen, Zhenlin Zhang, and Jinwei Yuan. Yolopv2: Better, faster, stronger for panoptic driving perception. *ArXiv*, abs/2208.11434, 2022.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023.

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

[9] Shilong Liu, Feng Li, Hao Zhang, Xiao Bin Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *ArXiv*, abs/2201.12329, 2022.

[10] Ruihao Wang, Jianbang Qin, Kai Li, Yaochen Li, Dongping Cao, and Jintao Xu. Bev-lanedet: a simple and effective 3d lane detection baseline. 2022.

[11] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiao hua Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Y. Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *ArXiv*, abs/2211.05778, 2022.

[12] Xinjiang Wang, Xingyi Yang, Shilong Zhang, Yijiang Li, Litong Feng, Shijie Fang, Chengqi Lyu, Kaibing Chen, and Wayne Zhang. Consistent-teacher: Towards reducing inconsistent pseudo-targets in semi-supervised object detection. 2022.

[13] Dongsheng Wu, Manwen Liao, Wei-Tian Zhang, Xinggang Wang, Xiang Bai, Wen-Qing Cheng, and Wen-Yu Liu. Yolop: You only look once for panoptic driving perception. *Machine Intelligence Research*, 19:550 – 562, 2021.

[14] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3040–3049, 2021.

[15] Chenyu Yang, Yuntao Chen, Haofei Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Y. Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *ArXiv*, abs/2211.10439, 2022.

[16] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13658–13667, 2021.

[17] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ArXiv*, abs/2010.04159, 2020.