2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Efficient and Hybrid Decoder for Local Map Construction in Bird's-Eye-View

Kun Tian[1], Yun Ye[1], Zheng Zhu[1], Peng Li[1], and Guan Huang[1]

Abstract— **High-definition maps are crucial perception elements for autonomous robot navigation systems, which can provide accurate scene layout and environment information for downstream motion prediction and planning control tasks. Traditional methods based on manual annotation or SLAM algorithms require massive labor efforts and time costs, which hinders the deployment of practical applications. Online construction of local maps from on-board cameras offers an alternative solution. Aiming at the problems of unsatisfying precision and redundant computation of HDMapNet, we propose an efficient and hybrid decoder (EHD) that consists of a CNN-based segmentation (Seg) head and a query-based lane detection head (QLD). Specifically, the Seg head outputs pixel-level semantic maps, and QLD predicts instance mask for each lane object through learnable query embeddings. The designed decoding method eliminates the cumulative error caused by inaccurate semantic maps and does not require additional clustering algorithm for post-processing. Through combining with a variety of bird's-eye-view (BEV) encoders, the effectiveness and efficiency of our EHD is demonstrated by extensive experiments. For segmentation task, the mIoU scores of semantic map can be improved by 1.3%∼2.9%. Additionally, the accuracy of lane detection is also significantly increased (more than 10.2% mAP) under all evaluation criteria. Since our method discards redundant post-processing, the inference speed is up to 22.71 FPS, which is 32 times faster than HDMapNet.**

I. Introduction

In the past few years, visual perception in bird's-eye-view (BEV) has attracted much attention from academia and industry, which is more convenient to plan and control the autonomous robot navigation system. For example, DETR3D [1] uses camera transformation matrices to establish the association between 3D object queries and 2D image features for 3D object detection. Except for moving objects, this paper focuses on local map construction [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] in BEV space. It is a vital task that provides positioning information such as scene layout for the downstream motion prediction and behavior control.

According to the granularity of perception, local map construction can be divided into two sub-tasks: semantic-level segmentation and instance-level detection. Although many works [2], [3], [4], [5], [6], [7], [8], [9], [10] have discussed how to perform local semantic segmentation, few studies have explored the fine-grained lane detection task. Since the shape of lane instances is irregular, it is difficult to perform detection by defining rectangular boxes as for general objects. Therefore, recognizing the sparse distributed lane instances in a top-down manner is also not feasible. To achieve a fine-grained description of lane contours and locations, an alternative solution is to use a bottom-up strategy. For example, HDMapNet [11] segments the lanes at pixel level in advance, and then generates multiple cluster centers as identities of different lane instances through a post-processing algorithm. However, inaccurate semantic maps will lead to cumulative errors in the subsequent clustering process, which impairs the accuracy of lane detection task.

As shown in Figure 1, we have observed several apparent deficiencies in the prediction of HDMapNet (surrounding). For segmentation maps in the first row, HDMapNet failed to predict complete lane dividers and pedestrian crossings. In addition, the localization of lane boundaries is not precise. For lane instances in the second row, although HDMapNet has predicted connected segmentation results, pixels representing the same object are not well clustered. Instead, a lane divider is discarded as background noise, which is equivalent to missing detection in the general object detection task. Moreover, HDMapNet can aggregate the features of neighboring pixels, but it is not good at modeling the correlation between long-distance embeddings. This results in the spatially connected lanes being decomposed into many sub-classes. The redundant predicted identities will be judged as false positives, which also harm the detection accuracy. In terms of design pattern, the clustering process of lane instances relies on the segmentation results of each category, which leads to the upper bound of detection performance being limited by the segmentation accuracy. We argue that such propagation error should be avoided during the algorithm design stage.

To alleviate the above problems, we propose an efficient and hybrid decoder (EHD) for semantic segmentation and more fine-grained instance-level lane detection tasks in BEV. Specifically, the segmentation head decodes pixel-level semantic maps and the query-based lane decoder (QLD) outputs lane instance for each query with the help of Transformer decoder. Whether the input is images from surrounding cameras, point clouds from LiDAR scans, or a mixture of them, our method could produce high performance perceptual results. Moreover, since the encoder and decoder are designed independently, EHD can be easily combined with various known encoding units, such as IPM [12], [13], VPN [14], LSS [8], etc. In order to improve the utilization of the network structure, we discard the embedding branch, and share the same convolution blocks and an upsampling module for both perception tasks. Therefore, our method has less parameters and computations. Of note, QLD does not require additional clustering algorithm to predict lane

[1]Kun Tian, Yun Ye, Zheng Zhu, Peng Li, and Guan Huang are with PhiGent Robotics, Beijing, China. {kun.tian, yun.ye, peng.li, guan.huang}@phigent.ai, zhengzhu@ieee.org

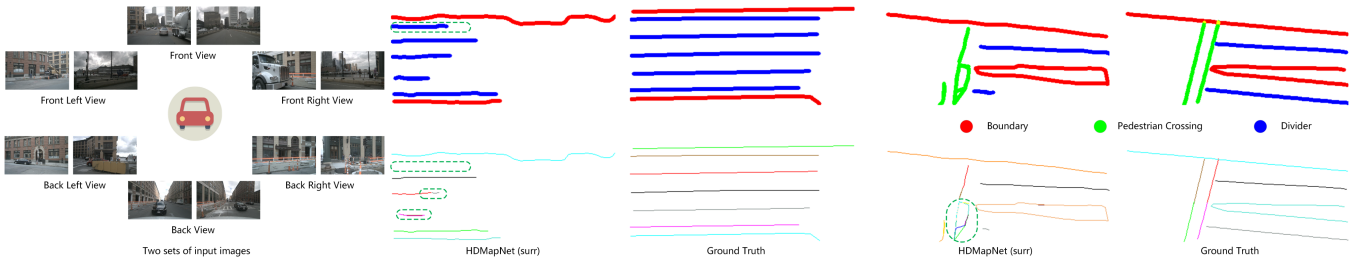

Fig. 1. Local maps constructed by HDMapNet for two sets of input images. We have marked some false predictions with green dashed ellipses. Please zoom in for more details.

instances, so it has a significant advantage in reasoning speed. To summarize, the major contributions of this paper are as follows:

- We propose a parallel hybrid decoder (EHD) for local map construction. Combined with seven classic and advanced bird's-eye-view encoders, EHD can steadily improve the performance of both semantic segmentation and lane detection tasks, which demonstrates its versatility.
- Different from the paradigm of obtaining instance-level predictions based on embedding branch and clustering algorithm, our method has less model parameters and computations. Besides, EHD accelerates the inference speed (up to 22.71 FPS) by more than 20 times, reaching a real-time level.
- The designed EHD refreshes the state-of-the-art lane detection performance in BEV, which can be regarded as a faster and stronger baseline for learning local map construction.

II. Related Work

SLAM based methods. Most High-definition maps (HD maps) are obtained by manual annotation on the collected LiDAR point clouds. There are some SLAM algorithms [15], [16], [17], [18], [19] that are able to merge LiDAR scans into consistent point clouds, thereby improving the precision and generation efficiency of HD maps. However, LiDARs have higher deployment and maintenance cost than on-board cameras. On the other hand, optical images captured by camera sensors can provide color and texture features, and with the help of deep neural networks, higher-level semantic information can also be extracted. Considering the time and labeling cost, constructing real-time local maps from cameras is an economical alternative.

Local semantic segmentation. There are many deep learning based solutions [12], [13], [14], [8], [11], [20] dedicated to semantic segmentation in bird's-eye-view (BEV). And researchers focus on obtaining BEV feature representations, which are the basis for model prediction. IPM (CB) is a method that conducts feature extraction in the perspective view and then maps them to BEV using inverse projective mapping (IPM) [12], [13]. Pan et al. [14] utilized two fully connected layers to learn the transformation of camera image features to the BEV. Lift-Splat-Shoot [8] introduces implicit depth probability estimation and projects depth-weighted image features into new feature representations that can be processed by 2D convolution operators, via voxel pooling. Zhou et al. [20] presented cross-view transformers that implicitly learn a mapping from individual camera features to a common BEV representation. Depending on the input modality, there are three versions of HDMapNet [11], which also design MLPs to transform image features into camera coordinate system and then to BEV with camera extrinsics. It can be seen that many researchers [21], [14], [22], [23], [24], [25], [26], [27], [28], [29] have explored how to design encoders based on learnable neural networks, camera models, and a mixture of these two paradigms. However, the decoder for specific downstream task is also important, which has been studied by few works [11].

Lane detection. Lane detection in perspective view [30], [31], [32] and bird's-eye-view [11], [33], are two different tasks. The curvature of lanes in BEV is larger. Roundabouts and other lane boundaries also exacerbate the difficulty of detection, while the shape of lanes in perspective view is relatively more regular. To the best of our knowledge, HDMapNet is the current state-of-the-art solution for BEV lane detection task. It draws on the design of LaneNet [34] and introduces an embedding branch and density-based spatial clustering of applications with noise (DBSCAN) to detect lane instances. We elaborate on its defects in Section I, which reflects the design motivation of this paper.

III. Method

A. Overview

The overview of the proposed perception framework is illustrated in Figure 2. In this paper, we disassemble the local map construction process into two parts, i.e., encoder and decoder, so that a modular configuration design paradigm is adopted in the specific implementation. The raw input of our method can be surrounding images and/or point clouds captured by on-board sensors. Taking Figure 2 as an example, a set of images from 6 cameras (front left, front, front right, back left, back, back right) are fed into the framework at each timestamp. Then, parallel feature extraction is performed for multi-view images with a shared backbone. The function of BEV encoder is to project and aggregate the outputs of different camera coordinate systems into a common BEV coordinate system. We have introduced

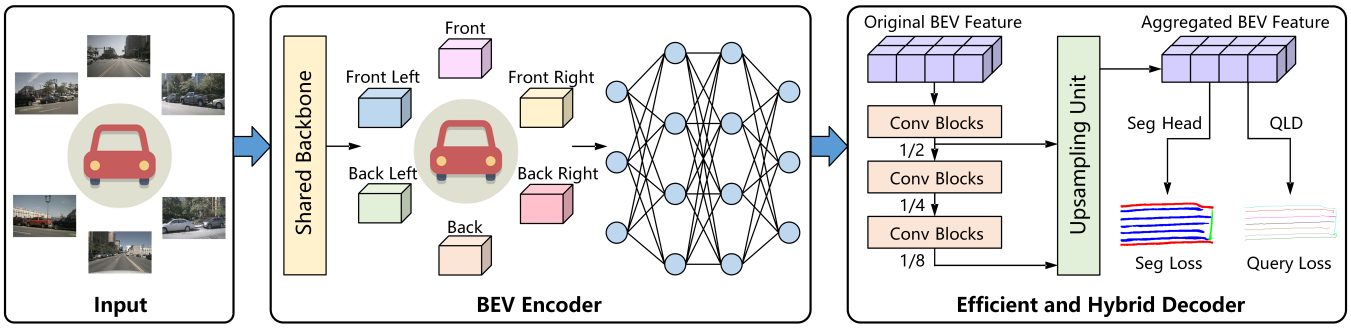

Fig. 2. The overview of our perception framework, which can be divided into a BEV encoder and a hybrid decoder. First, images from different perspectives are fed into a shared backbone and feature transformation network. Second, the original BEV features output from the encoder are aggregated through three stacked convolution layers and an upsampling unit. Finally, the task-specific decoding heads return the desired prediction results.

several encoding modules in the related work. As this paper focuses on designing a decoder with high performance and efficiency, we recommend that readers refer to the corresponding literature [14], [8], [20], [11] for implementation details of different BEV encoders.

The proposed hybrid decoder can be regarded as a combination of CNN [35], [36] and Transformer [37], [38], [39], which consists of three parts. First, three stacked convolutional blocks further aggregate the original BEV features output by the encoder, thereby enhancing the local association in the BEV coordinate system. The serial feature extraction process reduces the resolution of feature maps to 1/2, 1/4, and 1/8 of the original input. Second, we upsample and fuse low-resolution feature representations at different semantic levels, which aims to model the context information in BEV. This design is inspired by the feature pyramid network (FPN) [40], but has a simpler structure. Considering the computation and inference efficiency of the framework, only one upsampling unit is adopted. After that, the aggregated BEV features are fed into task-specific decoding modules, namely segmentation head (Seg head) and query-based lane decoder (QLD). The final output and training labels in the BEV coordinate system are used to compute segmentation loss and detection loss, supervising the update of model parameters. The detailed decoding implementation and training process are introduced in the next subsection.

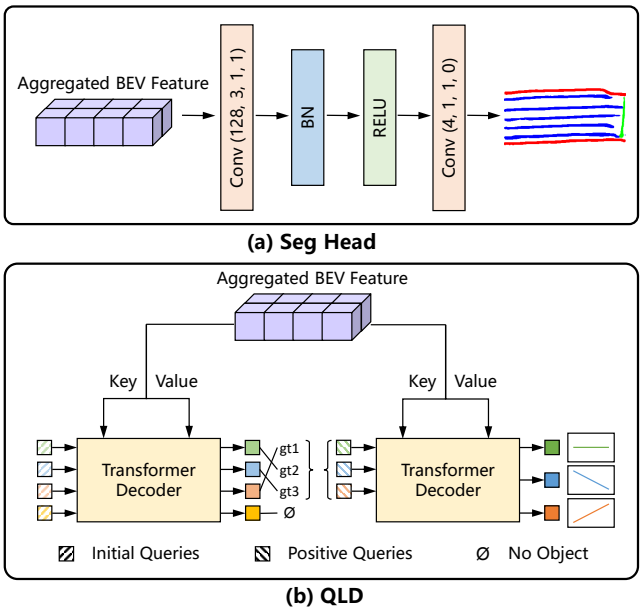

Fig. 3. The structure of the hybrid decoder. (a) Simple but efficient semantic segmentation head. (b) Two serially connected Transformer decoders. The first one is used to refine and filter the initial queries, and the second one decodes instance predictions associated with the positive queries.

B. Efficient and Hybrid Decoder

Figure 3 (a) shows a lightweight CNN head for pixel-level semantic segmentation. Convolution-Batch Normalization-RELU-Convolution is a classic decoding combination. (c, k, s, p) denotes the output channel, kernel size, stride and padding of the convolution layer. The shape of final output tensor is $B \times 4 \times H \times W$, where B, H, W represent the batch size, height, width of the predicted BEV maps. The channel dimension is 4, corresponding to three kinds of lanes (divider, pedestrian crossing, boundary) and the background category. The segmentation loss is calculated using the ground truth and classification map activated by softmax function, which can be written as:

$$L_{seg} = -\frac{1}{K} \sum_{i=1}^{K} \sum_{c=1}^{4} y_{ic} \log \left(p_{ic} \right), \tag{1}$$

where p_{ic} represents the predicted probability that the i-th pixel belongs to the c-th category, y_{ic} denotes the class label in one-hot form, and $K = H \times W$ is the number of pixel features in the BEV coordinate system. Following the setting of HDMapNet, H and W are set to 200 and 400.

Figure 3 (b) shows the structure of query-based decoder, whose working pattern consists of two steps. First, defining the initial query embeddings (IQ), which are used to store the statistics of the latent lane instances. The first Transformer decoder updates IQ based on the BEV features of each timestamp. This preliminary decoding can be viewed as a refinement of the initial query embeddings. After that, the output queries are highly correlated with dynamic inputs,

e.g., images from surrounding cameras. The second Transformer decoder queries the aggregated BEV features with the positive embeddings (PQ) that are assigned with ground truth lanes (GTs), and outputs the corresponding instance-level perception results.

The training process of Transformer decoders. We uses bipartite matching to adaptively establish the *sample-label* assignment strategy. In the first stage, it is assumed that the model initializes M IQs and there are N lane objects (GTs) in the current timestamp. Each IQ will calculate the classification loss of C categories. According to the class labels of N lanes, we construct a classification cost matrix with the of shape $M \times N$. Finally, the association between IQs and GTs can be obtained with the minimum matching cost through bipartite matching algorithm. Next, only N positive queries (PQs) assigned with GT lanes will be passed to the second Transformer decoder. In this stage, the decoder outputs the classification activation and instance mask prediction for each PQ. The cost matrix with shape of $N \times N$ comprehensively considers the classification accuracy and positioning accuracy of all PQs.

The overall optimization objective consists of three items, which can be expressed as:

$$L_{all} = L_{seg} + L_{IQ} + L_{PQ}. \tag{2}$$

L_{seg} teaches model to predict accurate semantic segmentation results. L_{IQ} is used to supervise classification predictions from M initial queries, and L_{PQ} guides N positive queries to generate accurate lane instance masks. Considering the imbalance of positive and negative samples in lane detection task, L_{IQ} and L_{PQ} are implemented as:

$$\begin{aligned} L_{IQ} &= -\frac{1}{M}\sum_{i=1}^{M}(1-p_i)^{\gamma}\log(p_i), \\ L_{PQ} &= -\frac{1}{K*N}\sum_{i=1}^{N}\sum_{j=1}^{K}(1-p_{ij})^{\gamma}\log(p_{ij}), \end{aligned} \tag{3}$$

where p_i is the probability that the i-th initial query is correctly classified and p_{ij} is the probability that the j-th pixel of the i-th positive query is correctly located. γ is set to 2, which is consistent with [41]. In summary, the proposed parallel decoding paradigm prevents the accuracy of lane detection from being compromised by the accumulated error of semantic segmentation, and the end-to-end framework discards the post-processing of identity clustering, which significantly improves the inference speed.

IV. Experiments

A. Implementation Details

Dataset and Evaluation Metrics. The experiments are conducted on nuScenes Dataset [42], which consists of 1,000 video sequences collected in Boston and Singapore for autonomous driving task. Specifically, the training and validation sets consist of 28,130 and 6,019 samples, storing images from 6 on-board cameras and the raw point clouds. To model road structures and driving rules, we focus on the local construction of three static map elements: lane divider, pedestrian crossing, and lane boundary. Following the previous approach [11], the performance of local map construction is evaluated on the tasks of semantic map segmentation and lane instance detection. For semantic metrics, we compute the intersection-over-union (IoU) and Chamfer distance (CD) for each class. For instance-level metric, average precision (AP) is adopted to trade off detection precision and recall.

Experimental settings. The BEV coordinates are constructed based on ego-vehicle system, where the ranges of X-axis and Y-axis are $[-30.0m, 30.0m]$ and $[-15.0m, 15.0m]$. As sampling interval is set to $0.15m$, the resolution of ground truth maps is 200 $\times$ 400 pixels, which follows the predecessor [11]. The image backbone network in BEV encoder is unified as EfficientNet-B0 [43] pre-trained on ImageNet [44]. And for LiDAR point clouds, PointPillars [45] and PointNet [46] are used to perform feature extraction. The network structure for perspective feature transformation, such as VPN [14], LSS [8], CVT [20] etc., is consistent with the original paper. As for BEV decoder, the number of initial queries, attention heads, and decoding layers are 25, 8, and 4. The embedding dimension is set to 64 for computational efficiency.

Other details. For the training process, four NVIDIA GeForce RTX 3090 are employed and the mini-batch per GPU is set to 6 images with a resolution of 128 $\times$ 352 pixels, which is the same setting as [11] for fair comparison. More concretely, we utilize AdamW [47] as the optimizer, with initial learning rate and weight decay as 2e-4 and 0.01. All models are trained for 30 epochs, and one-cycle learning policy [48] is applied with the maximum learning rate as 2e-3. In order to obtain a pure benchmark, we do not use any data augmentation strategies. For the testing process, the inference rates of models are evaluated on an idle machine, and the batch size of input sample is 1, representing 6 images and/or LiDAR point clouds within a timestamp. All experiments are implemented using the PyTorch framework [49] and follow the same code-base.

B. Comparison with Existing Methods

In this section, the segmentation and detection performance of EHD are mainly compared with the decoder (S-CNN) of HDMapNet. S-CNN is a CNN-based decoder for segmenting semantic maps and extracting instance embeddings, which consists of two separate upsampling units and several convolutional blocks. Moreover, a clustering algorithm (DBSCAN) is utilized to generate lane instance identities. To further explore the scalability of EHD, we conduct experiments with seven advanced BEV encoders.

Table I presents the evaluation results at semantic level. The calculation of IoU score is consistent with traditional segmentation task. $\mathrm{CD_p}$ is computed by the Chamfer distance from ground truth to predictions, and $\mathrm{CD_L}$ is calculated by the Chamfer distance from predictions to ground truth. ↑/↓ indicates that larger/smaller values are better. The perceptual accuracy of IPM (CB)+S-CNN is significantly lower than

TABLE I

COMPARISON ON SEMANTIC-LEVEL EVALUATION RESULTS. S-CNN REPRESENTS SEPARATE CNN, AND EHD IS THE ABBREVIATION OF THE PROPOSED HYBRID DECODER. FOLLOWING TABLES SHARE THESE ANNOTATIONS.

Method			Divider				Pedestrian Crossing				Boundary			
Encoder	**S-CNN**	**EHD**	IoU↑	CD_p↓	CD_L↓	CD↓	IoU↑	CD_p↓	CD_L↓	CD↓	IoU↑	CD_p↓	CD_L↓	CD↓
IPM (CB)	✓		23.6	2.138	4.251	3.375	3.9	1.968	5.000	5.000	19.9	1.266	4.558	3.364
		✓	**44.1**	**1.257**	**1.288**	**1.271**	**24.3**	**1.734**	**1.670**	**1.709**	**45.2**	**0.906**	**0.898**	**0.902**
CVT	✓		38.6	1.065	1.917	1.502	18.8	1.243	2.993	2.211	37.4	0.729	1.555	1.160
		✓	**40.8**	1.636	**1.266**	**1.468**	18.1	2.081	**2.094**	**2.085**	**39.9**	1.228	**0.997**	**1.125**
LSS	✓		40.7	0.862	1.756	1.321	25.1	1.057	2.167	1.639	42.1	0.666	1.067	0.866
		✓	**43.0**	1.195	**1.289**	**1.240**	24.2	1.720	**1.590**	1.670	**44.7**	0.884	**0.932**	0.907
VPN	✓		41.8	0.814	1.541	1.184	25.0	1.148	2.192	1.699	43.0	0.653	1.046	0.850
		✓	**44.2**	1.128	**1.296**	1.209	24.4	1.593	**1.618**	**1.603**	**44.5**	0.893	**0.871**	0.882
HDMNet (Surr)	✓		37.1	0.780	2.878	1.919	20.8	0.941	2.998	2.139	39.3	0.644	1.394	1.027
		✓	**39.4**	1.134	**2.716**	1.979	**21.2**	1.671	**2.368**	**1.990**	**41.9**	0.872	**1.311**	1.092
HDMNet (LiDAR)	✓		45.6	0.867	1.734	1.316	33.4	0.790	2.574	1.814	54.8	0.432	1.106	0.773
		✓	**48.7**	1.338	**1.178**	**1.265**	32.8	1.812	**1.735**	**1.781**	**59.3**	0.711	**0.787**	**0.747**
HDMNet (Fusion)	✓		43.8	0.632	2.110	1.403	33.6	0.722	2.544	1.705	53.2	0.401	0.924	0.657
		✓	**48.1**	0.911	**1.867**	**1.400**	31.3	1.353	**2.139**	1.726	**59.9**	0.516	**0.801**	**0.657**

TABLE II

COMPARISON ON INSTANCE-LEVEL EVALUATION RESULTS. THE CALCULATION OF AP AND MAP IS SIMILAR TO GENERAL DETECTION TASK.

Method			Divider				Pedestrian Crossing				Boundary			
Encoder	**S-CNN**	**EHD**	AP@.2	AP@.5	AP@1.	mAP	AP@.2	AP@.5	AP@1.	mAP	AP@.2	AP@.5	AP@1.	mAP
IPM (CB)	✓		2.5	6.5	10.9	6.7	0.1	1.0	4.1	1.7	2.4	6.2	11.9	6.8
		✓	**19.1**	**37.6**	**54.0**	**36.9**	**10.8**	**30.1**	**50.1**	**30.4**	**23.0**	**47.5**	**65.5**	**45.3**
CVT	✓		10.1	22.3	31.8	21.4	1.5	9.8	26.0	12.4	13.7	29.0	42.3	28.3
		✓	**15.3**	**35.0**	**49.2**	**33.2**	**5.4**	**20.3**	**42.2**	**22.6**	**16.7**	**36.6**	**57.4**	**36.9**
LSS	✓		9.5	21.2	33.0	21.2	2.4	13.7	33.6	16.5	17.2	35.4	50.3	34.3
		✓	**19.9**	**40.9**	**55.1**	**38.7**	**9.5**	**29.2**	**50.3**	**29.7**	**23.8**	**46.6**	**65.2**	**45.2**
VPN	✓		11.6	25.0	36.6	24.4	2.9	13.9	34.3	17.0	19.5	39.8	53.8	37.7
		✓	**21.6**	**44.2**	**57.8**	**41.2**	**11.4**	**32.6**	**52.4**	**32.1**	**24.1**	**48.6**	**65.6**	**46.1**
HDMNet (Surr)	✓		8.9	20.1	27.7	18.9	3.4	11.6	27.1	14.0	14.8	33.0	47.3	31.7
		✓	**18.5**	**33.8**	**45.1**	**32.5**	**9.2**	**25.3**	**42.8**	**25.8**	**22.4**	**44.2**	**59.3**	**42.0**
HDMNet (LiDAR)	✓		9.2	21.0	30.9	20.4	4.1	17.9	36.4	19.5	25.6	42.4	54.6	40.8
		✓	**20.6**	**42.6**	**55.9**	**39.7**	**14.3**	**34.1**	**50.9**	**33.1**	**35.6**	**57.5**	**70.1**	**54.4**
HDMNet (Fusion)	✓		13.0	26.8	36.5	25.4	9.1	23.8	39.1	24.0	25.7	46.3	59.4	43.8
		✓	**24.6**	**42.4**	**53.5**	**40.2**	**15.3**	**33.3**	**47.3**	**32.0**	**39.7**	**61.2**	**72.1**	**57.7**

other baselines, which is because its training is unstable and the model does not converge in the end. Compared with S-CNN, the proposed EHD could adapt to all BEV encoders. For IoU scores, our method has obvious advantages on lane divider and boundary. For example, HDMapNet (Fusion)+EHD improves the HDMapNet (Fusion)+S-CNN by 4.3% and 6.7% IoU scores. Since Transformer decoder pays more attention to the global correlation, and the BEV space is dominated by these two categories, our method performs mediocrely for pedestrian crossings with only a few samples. Furthermore, EHD usually has better CD_L and worse CD_p, which indicates that our method is able to identify more foreground pixels, but also produces some false positives.

Table II reports the evaluation results at instance level. It can be seen that our method achieves significant performance gains on all evaluation criteria (AP@.2, AP@.5, AP@1., mAP). For example, LSS+EHD surpasses LSS+S-CNN by 17.5%, 13.2%, and 10.9% mAP on lane divider, pedestrian crossing, and lane boundary classes. This is because EHD alleviates the identity redundancy problem directly from the decoding mode. For each timestamp, the number of predicted lane instances is less than or equal to the number of initial queries. Although the number of false positive pixels increases, from the perspective of detection accuracy, the number of false positive instances decreases, thus improving AP and mAP for all categories.

Furthermore, we conduct ablation studies on two hyperparameters, i.e., the number of query Q and the dimension of query embeddings D. For convenience, VPN and EHD are utilized as the BEV encoder and decoder. The comparison results are reported in Table III. Specifically, when we fix the dimension of query embeddings, reducing the number of queries ($25 \rightarrow 15$) impairs the detection accuracy, because the number of preset lanes is insufficient. When increasing the number of queries, there is a slight performance fluctuation. For example, the mAP of All Classes is increased from 39.8% to 40.4%, when Q is set to 125. Moreover, if the number of queries is fixed to 25, reducing the dimension of embeddings ($64 \rightarrow 32$) mainly decreases the mAP of pedestrian crossing. When D is increased to 512, the mAP of All Classes is increased by 1.0%. Overall, EHD is robust to the changes of the above two hyperparameters.

In Table IV, we comprehensively evaluate the performance of the model on all categories, as well as the computational efficiency. When the input modality is single image data, VPN+EHD achieves the second highest mIoU, the best CD and the best mAP. Despite the simple structure of VPN consisting of only two fully-connected layers, our method still achieves considerable perceptual accuracy, illustrating the importance of decoder design. When the input data contains point clouds, both mIoU and mAP are significantly imcreased (+12.2% and +9.9%), which demonstrates the ability of multi-modal data to improve the perception accuracy. As shown in the last three columns of Table IV, separate upsampling unit and embedding branch bring extra model parameters and calculations. In contrast, our QLD and segmentation head shares the same aggregated BEV features, and the added parameters and GFLOPs are negligible. What's more, our EHD also speeds up the inference.

TABLE III

ABLATION EXPERIMENTS OF DIFFERENT Q AND D, WHICH REPRESENT THE NUMBER AND DIMENSION OF QUERIES.

VPN+EHD		Divider				Ped Crossing				Boundary				All Classes			
Q	D	AP@.2	AP@.5	AP@1.	mAP	AP@.2	AP@.5	AP@1.	mAP	AP@.2	AP@.5	AP@1.	mAP	AP@.2	AP@.5	AP@1.	mAP
15	64	21.5	42.6	55.6	39.9	10.4	29.2	48.1	29.2	24.0	46.5	64.5	45.0	18.6	39.4	56.1	38.0
25	64	21.6	44.2	**57.8**	41.2	11.4	32.6	52.4	32.1	24.1	48.6	65.6	46.1	19.1	41.8	58.6	39.8
50	64	21.8	43.4	56.9	40.7	12.9	33.5	52.8	33.0	23.3	48.6	64.3	45.4	19.3	41.8	58.0	39.7
75	64	22.1	42.1	56.4	40.2	11.6	31.3	50.5	31.1	22.8	46.5	63.6	44.3	18.8	40.0	56.8	38.5
100	64	21.0	43.0	56.7	40.2	**13.2**	32.9	52.1	32.7	24.2	48.8	65.2	46.1	19.5	41.6	58.0	39.7
125	64	**23.3**	44.3	57.4	**41.6**	12.5	33.8	52.6	33.0	24.5	49.2	66.0	46.6	**20.1**	42.4	58.7	40.4
25	32	22.7	44.1	57.2	41.3	11.8	31.6	51.2	31.6	25.0	49.1	65.4	46.5	19.8	41.6	57.9	39.8
25	128	22.1	44.5	57.4	41.3	12.3	32.9	52.9	32.7	**25.1**	48.3	66.1	46.5	19.8	41.9	58.8	40.2
25	256	22.4	43.2	56.5	40.7	12.6	31.9	50.1	31.6	24.8	50.2	66.1	47.0	20.0	41.8	57.6	39.8
25	512	22.6	**44.7**	57.2	41.5	12.8	**34.1**	**53.9**	**33.6**	24.8	**50.3**	**66.9**	**47.3**	20.0	**43.0**	**59.3**	**40.8**

TABLE IV

COMPARISON ON THE COMPREHENSIVE PERFORMANCE OF ALL CATEGORIES AND MODEL'S COMPUTATIONAL EFFICIENCY.

Method			All Classes								Model efficiency		
Encoder	S-CNN	EHD	mIoU↑	CD_p↓	CD_L↓	CD↓	AP@.2	AP@.5	AP@1.	mAP	#param.	GFLOPs	FPS
IPM (CB)	✓		15.8	1.791	4.603	3.913	1.7	4.6	8.9	5.1	12.10M	124.02	1.26
		✓	**37.9**	**1.299**	**1.285**	**1.294**	**17.6**	**38.4**	**56.5**	**37.5**	**10.89M**	**75.24**	**18.95**
CVT	✓		31.6	1.012	2.155	1.624	8.4	20.3	33.4	20.7	12.59M	130.27	0.63
		✓	**32.9**	1.648	**1.452**	**1.559**	**12.4**	**30.6**	**49.6**	**30.9**	**11.38M**	**81.48**	**17.72**
LSS	✓		36.0	0.862	1.663	1.275	9.7	23.4	39.0	24.0	12.10M	124.02	0.62
		✓	**37.3**	1.266	**1.271**	**1.272**	**17.8**	**38.9**	**56.9**	**37.8**	**10.89M**	**75.24**	**18.33**
VPN	✓		36.6	0.872	1.593	1.245	11.3	26.2	41.6	26.4	12.47M	124.04	0.62
		✓	**37.7**	1.205	**1.262**	**1.231**	**19.1**	**41.8**	**58.6**	**39.8**	**11.26M**	**75.26**	**21.35**
HDMNet (Surr)	✓		32.4	0.788	2.423	1.695	9.0	21.6	34.0	21.5	76.96M	128.17	0.77
		✓	**34.2**	1.226	**2.132**	**1.687**	**16.7**	**34.4**	**49.1**	**33.4**	**75.74M**	**79.39**	**20.02**
HDMNet (LiDAR)	✓		44.6	0.696	1.805	1.301	13.0	27.1	40.6	26.9	10.56M	313.44	0.7
		✓	**46.9**	1.287	**1.233**	**1.264**	**23.5**	**44.7**	**58.9**	**42.4**	**9.34M**	**264.65**	**22.71**
HDMNet (Fusion)	✓		43.5	0.585	1.859	1.255	15.9	32.3	45.0	31.1	81.40M	326.65	0.72
		✓	**46.4**	0.927	**1.602**	1.261	**26.5**	**45.7**	**57.6**	**43.3**	**80.18M**	**277.87**	**14.88**

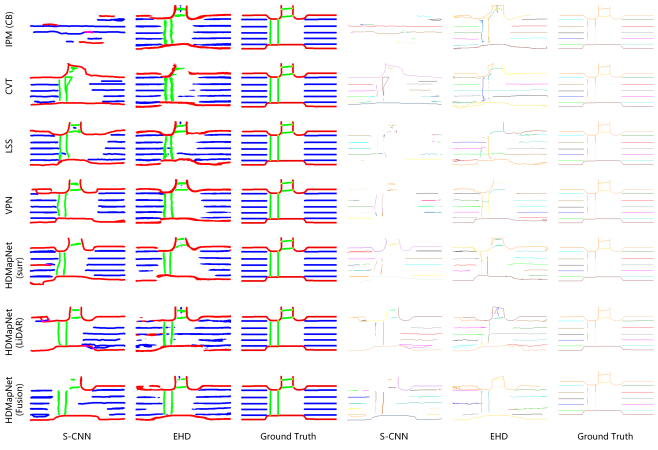

Fig. 4. Comparison of decoding output of S-CNN and EHD with seven BEV encoders. Blue, green, and red elements in segmentation maps represent lane divider, pedestrian crossing, and boundary divider, respectively. Different instances in detection maps are distinguished by different colors.

C. Visualization and Analysis

As mentioned above, we combined S-CNN and EHD with seven BEV encoders, and trained 14 models for comparison. The predicted semantic segmentation and instance detection results are visualized in Figure 4. There are some observations worth noting. First, for all encoders, the semantic segmentation results decoded by S-CNN have the problem of missing detection and incomplete prediction, while EHD has a higher recall for foreground pixels. We argue that this is because the Transformer decoder has a natural global modeling capability, i.e., a larger receptive field. Distant pixel features can be referenced as a supplement to local information, which alleviates missing perception to some extent. Second, for instance detection task, the clustering accuracy of all encoders+S-CNN is not satisfactory. In one case, the entire lane is discarded due to failure clustering, which produces false negative. Another case is that multiple sub-classes are generated due to inaccurate clustering, i.e., redundant identities. The main advantage of EHD is that there is a one-to-one correspondence between the predicted lane instances and queries, thus alleviating the challenge of false positives.

V. CONCLUSION

In this paper, we propose a faster and stronger local map construction baseline in bird's-eye-view, which is attributed to the efficient and hybrid decoder. Experiments on seven state-of-the-art BEV encoders confirm the generality of our method. The designed EHD reduces the model parameters and improves the computational efficiency through sharing global feature extraction and local feature aggregation. Moreover, the query-based lane decoder alleviates the problem of redundant identities generated by the embedding branch and clustering algorithm, which effectively improves detection accuracy. And the multi-task joint training also increases mIoU scores of semantic segmentation. Finally, this parallel and end-to-end decoding pattern significantly accelerates the inference speed, demonstrating the scalability and applicability of our method.

REFERENCES

[1] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning, 8-11 November 2021, London, UK*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 2021, pp. 180–191.

[2] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11219. Springer, 2018, pp. 815–831.

[3] S. A. Abbas and A. Zisserman, "A geometric approach to obtain a bird's eye view from an image," in *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*. IEEE, 2019, pp. 4095–4104.

[4] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder-decoder networks," *IEEE Robotics Autom. Lett.*, vol. 4, no. 2, pp. 445–452, 2019.

[5] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," in *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*. BMVA Press, 2019, p. 285.

[6] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 11 135–11 144.

[7] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3d scene reconstruction from posed images," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12352. Springer, 2020, pp. 414–431.

[8] J. Philion and S. 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*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12359. Springer, 2020, pp. 194–210.

[9] L. Reiher, B. Lampe, and L. Eckstein, "A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view," in *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*. IEEE, 2020, pp. 1–7.

[10] A. Loukkal, Y. Grandvalet, T. Drummond, and Y. Li, "Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, 2021, pp. 51–60.

[11] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4628–4634.

[12] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, "Restricted deformable convolution-based road scene semantic segmentation using surround view cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4350–4362, 2020.

[13] T. Sämann, K. Amende, S. Milz, C. Witt, M. Simon, and J. Petzold, "Efficient semantic segmentation for visual bird's-eye view interpretation," in *Intelligent Autonomous Systems 15 - Proceedings of the 15th International Conference IAS-15, Baden-Baden, Germany, June 11-15, 2018*, ser. Advances in Intelligent Systems and Computing, M. Strand, R. Dillmann, E. Menegatti, and S. Ghidoni, Eds., vol. 867. Springer, 2018, pp. 679–688.

[14] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics Autom. Lett.*, vol. 5, no. 3, pp. 4867–4873, 2020.

[15] R. Dubé, A. Gawel, H. Sommer, J. I. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3d lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*. IEEE, 2017, pp. 1004–1011.

[16] E. Mendes, P. Koch, and S. Lacroix, "Icp-based pose-graph SLAM," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2016, Lausanne, Switzerland, October 23-27, 2016*. IEEE, 2016, pp. 195–200.

[17] T. Shan, B. J. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*. IEEE, 2020, pp. 5135–5142.

[18] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A robust pose graph approach for city scale lidar mapping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*. IEEE, 2018, pp. 1175–1182.

[19] J. Jiao, "Machine learning assisted high-definition map creation," in *2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC 2018, Tokyo, Japan, 23-27 July 2018, Volume 1*, S. Reisman, S. I. Ahamed, C. Demartini, T. M. Conte, L. Liu, W. R. Claycomb, M. Nakamura, E. Tovar, S. Cimato, C. Lung, H. Takakura, J. Yang, T. Akiyama, Z. Zhang, and K. Hasan, Eds. IEEE Computer Society, 2018, pp. 367–373.

[20] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 760–13 769.

[21] M. H. Ng, K. Radia, J. Chen, D. Wang, I. Gog, and J. E. Gonzalez, "Bev-seg: Bird's eye view semantic segmentation using geometry and semantic point cloud," *CoRR*, vol. abs/2006.11436, 2020.

[22] N. Hendy, C. Sloan, F. Tian, P. Duan, N. Charchut, Y. Xie, C. Wang, and J. Philbin, "FISHING net: Future inference of semantic heatmaps in grids," *CoRR*, vol. abs/2006.09917, 2020.

[23] K. Mani, S. Daga, S. Garg, N. S. Shankar, K. M. Jatavallabhula, and K. M. Krishna, "Monolayout: Amodal scene layout from a single image," *CoRR*, vol. abs/2002.08394, 2020.

[24] Y. Zhang, Z. Zhu, W. Zheng, J. Huang, G. Huang, J. Zhou, and J. Lu, "Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," *arXiv preprint arXiv:2205.09743*, 2022.

[25] W. Yang, Q. Li, W. Liu, Y. Yu, Y. Ma, S. He, and J. Pan, "Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 15 536–15 545.

[26] A. Saha, O. M. Maldonado, C. Russell, and R. Bowden, "Translating images into maps," *CoRR*, vol. abs/2110.00966, 2021.

[27] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "FIERY: future instance prediction in bird's-eye view from surround monocular cameras," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 15 253–15 262.

[28] A. Saha, O. M. Maldonado, C. Russell, and R. Bowden, "Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation," in *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*. IEEE, 2021, pp. 5133–5139.

[29] K. Chitta, A. Prakash, and A. Geiger, "NEAT: neural attention fields for end-to-end autonomous driving," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 15 773–15 783.

[30] H. Abualsaud, S. Liu, D. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, "Laneaf: Robust multi-lane detection with affinity fields," *IEEE Robotics Autom. Lett.*, vol. 6, no. 4, pp. 7477–7484, 2021.

[31] J. Han, X. Deng, X. Cai, Z. Yang, H. Xu, C. Xu, and X. Liang, "Laneformer: Object-aware row-column transformers for lane detection," *CoRR*, vol. abs/2203.09830, 2022.

[32] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on local: Detecting lane marker from bottom up via key point," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 14 122–14 130.

[33] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, and T. E. Choe, "Gen-lanenet: A generalized and scalable approach for 3d lane detection," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXI*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12366. Springer, 2020, pp. 666–681.

[34] Z. Wang, W. Ren, and Q. Qiu, "Lanenet: Real-time lane detection networks for autonomous driving," *CoRR*, vol. abs/1807.01726, 2018.

[35] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. W. III, and A. F. Frangi, Eds., vol. 9351. Springer, 2015, pp. 234–241.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12346. Springer, 2020, pp. 213–229.

[40] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 936–944.

[41] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 2999–3007.

[42] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 11 618–11 628.

[43] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[45] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 12 697–12 705.

[46] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 77–85.

[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[48] Y. Yan, Y. Mao, and B. Li, "SECOND: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.