Supplementary Material

Zhiqi Li[1,2,*], Wenhai Wang[2,*], Hongyang Li[2,*], Enze Xie[3]
Chonghao Sima[2], Tong Lu[1], Yu Qiao[2], and Jifeng Dai[2,]

[1] Nanjing University [2] Shanghai AI Laboratory [3] The University of Hong Kong

A Implementation Details

In this section, we provide more implementation details of the proposed method and experiments.

A.1 Traning Strategy

Following previous methods [10,11], we train all models with 24 epochs, a batch size of 1 (containing 6 view images) per GPU, a learning rate of 2×10^{-4}, learning rate multiplier of the backbone is 0.1, and we decay the learning rate with a cosine annealing [3]. We employ AdamW [4] with a weight decay of 1×10^{-2} to optimize our models.

A.2 VPN and Lift-Splat

We use VPN [5] and Lift-Splat [6] as two baselines in this work. The backbone and the task heads are the same as the BEVFomer for fair comparisons.
VPN. We employ the official codes[1] in this work. Limited by the huge amount of parameters of MLP, it is difficult for VPN to generate high-resolution BEV (*e.g.*, 200× 200). To compare with VPN, in this work, we transform the single-scale view features into BEV with a low resolution of 50×50 via two view translation layers.
Lift-Splat. We enhance the camera encoder of Lift-Splat[2] with two additional convolutional layers for a fair comparison with our BEVFormer under a comparable parameter number. Other settings remain unchanged.

A.3 Spatial Cross-Attention

Global Attention. Besides deformable attention [11], our spatial cross-attention can also be implemented by global attention (*i.e.*, vanilla multi-head attention) [9]. The most straightforward way to employ global attention is making each BEV query interact with all multi-camera features, and this conceptual implementation does not require camera calibration. However, the computational cost of this straightforward way is unaffordable. Therefore, we still utilize the

[1] https://github.com/pbw-Berwin/View-Parsing-Network
[2] https://github.com/nv-tlabs/lift-splat-shoot

camera intrinsic and extrinsic to decide the hit views that one BEV query deserves to interact. This strategy makes that one BEV query usually interacts with only one or two views rather than all views, making it possible to use global attention in the spatial cross-attention. Notably, compared to other attention mechanisms that rely on precise camera intrinsic and extrinsic, global attention is more robust to camera calibration.

A.4 Task Heads

Detection Head. We predict 10 parameters for each 3D bounding box, including the 3 parameters (l, w, h) for the scale of each box, 3 parameters (x_o, y_o, z_o) for the center location, 2 parameters $(\cos(\theta), \sin(\theta))$ for object's yaw θ, 2 parameters (v_x, v_y) for the velocity. Only L_1 loss and L_1 cost are used during training phase. Following [10], we use 900 object queries and keep 300 predicted boxes with highest confidence scores during inference.

Sementation Head. As shown in Fig. A1, for each class of the semantic map, we follow the mask decoder in [2] to use one learnable query to represent this class, and generate the final segmentation masks based on the attention maps from the vanilla multi-head attention.

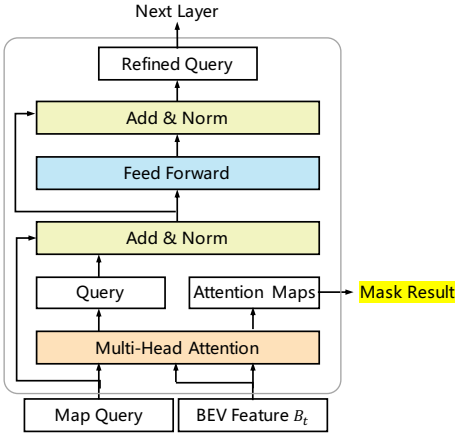

Fig. A1: **Detailed architecture of the mask decoder for map segmentation in this work.**

B Experiments on Waymo Open Dataset

Waymo Open Dataset [8] is a large-scale autonomous driving dataset with 798 training sequences and 202 validation sequences. Note that the five images at each frame provided by Waymo have only about 252° horizontal FOV, but the provided annotated labels are 360° around the ego car. We remove these bounding boxes that can not be visible on any images in training and validation sets. Due to the Waymo Open Dataset being large-scale and high-rate [7], we use a subset of the training split by sampling every 5^{th} frame from the training sequences and only detect the vehicle category. We use the thresholds of 0.5 and 0.7 for 3D IoU to compute the mAP on Waymo dataset. For experiments on Waymo, we change a few settings. Due to the camera system of Waymo can not capture the whole scene around the ego car [8], the default spatial shape of BEV queries is 300×220, the perception ranges are $[-35.0\text{m}, 75.0\text{m}]$ for the X-axis

Table A1: **3D detection results on Waymo `val` set under Waymo evaluation metric and nuScenes evaluation metric.** "L1" and "L2" refer "LEVEL_1" and "LEVEL_2" difficulties of Waymo [8]. *: Only use the front camera and only consider object labels in the front camera's field of view (50.4°). †: We compute the NDS score by setting ATE and AAE to be 1. "L" and "C" indicate LiDAR and Camera, respectively.

Method	Modality	Waymo Metrics				Nuscenes Metrics				
		IoU=0.5		IoU=0.7						
		L1/APH	L2/APH	L1/APH	L2/APH	NDS†↑	AP↑	ATE↓	ASE↓	AOE↓
PointPillars [1]	L	0.866	0.801	0.638	0.557	0.685	0.838	0.143	0.132	0.070
DETR3D [10]	C	0.220	0.216	0.055	0.051	0.394	0.388	0.741	**0.156**	0.108
BEVFormer	C	**0.280**	**0.241**	**0.061**	**0.052**	**0.426**	**0.440**	**0.679**	0.157	**0.101**
CaDNN* [7]	C	0.175	0.165	0.050	0.045	-	-	-	-	-
BEVFormer*	C	0.308	0.277	0.077	0.069	-	-	-	-	-

and $[-75.0\text{m}, 75.0\text{m}]$ for the Y-axis. The size of resolution s of each gird is 0.5m. The ego car is at (70, 150) of the BEV.

We also conduct experiments on Waymo, as shown in Tab. A1. Following [7], we evaluate the vehicle category with IoU criterias of 0.7 and 0.5. In addition, We also adopt the nuScenes metrics to evaluate the results since the IoU-based metrics are too challenging for camera-based methods. Due to a few camera-based works reported results on Waymo, we also use the official codes of DETR3D to perform experiments on Waymo for comparison. We can observe that BEVFormer outperforms DETR3D by Average Precision with Heading information (APH) [8] of 6.0% and 2.5% on LEVEL_1 and LEVEL_2 difficulties with IoU criteria of 0.5. On nuScenes metrics, BEVFormer outperforms DETR3D with a margin of 3.2% NDS and 5.2% AP. We also conduct experiments on the front camera to compare BEVFormer with CaDNN [7], a monocular 3D detection method that reported their results on the Waymo dataset. BEVFormer outperforms CaDNN with APH of 13.3% and 11.2% on LEVEL_1 and LEVEL_2 difficulties with IoU criteria of 0.5.

C Robustness on Camera Extrinsics

BEVFormer relies on camera intrinsics and extrinsics to obtain the reference points on 2D views. During the deployment phase of autonomous driving systems, extrinsics may be biased due to various reasons such as calibration errors, camera offsets, etc. As shown in Fig. A2, we show the results of models under different camera extrinsics noise levels. Compared to BEVFormer-S (point), BEVFormer-S utilizes the spatial cross-attention based on deformable attention [11] and samples features around the reference points rather than only interacting with the reference points. With deformable attention, the robustness of BEVFormer-S is stronger than BEVFormer-S (point). For example, with

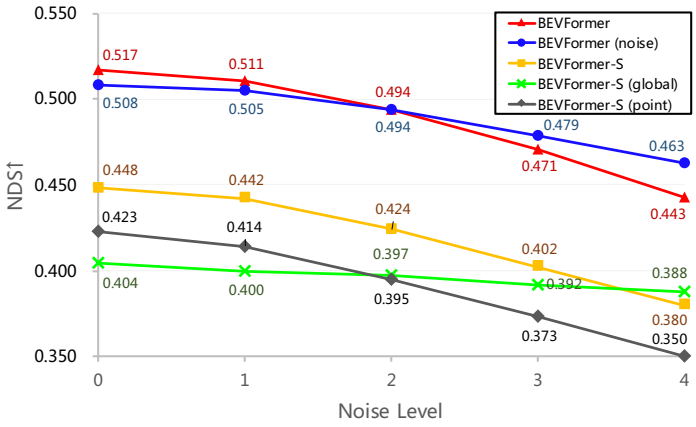

Fig. A2: **NDS of methods on nuScenes val set subjected to different levels of camera extrinsics noises.** For i-th level noises, the rotation noises are sampled from a normal distribution with mean equals 0 and variance equals i (rotation noise are in degrees, and the noise of each axis is independent), and the translation noises are sampled from a normal distribution with mean equals 0 and variance equals $5i$ (translation noises are in centimeters, and the noise of each direction is independent). "BEVFormer" is our default version. "BEVFormer (noise)" is trained with noisy extrinsics (noise level=1). "BEVFormer-S" is our static version of BEVFormer with the spatial cross-attention implemented by deformable attention [11]. "BEVFormer-S (global)" is BEVFormer-S with the spatial cross-attention implemented by global attention (*i.e.*, vanilla multi-head attention) [9]. "BEVFormer-S (point)" is BEVFormer-S with point spatial cross-attention where we degrade the interaction targets of deformable attention from the local region to the reference points only by removing the predicted offsets and weights.

the noise level being 4, the NDS of BEVFormer-S drops 15.2% (calculated by $1 - \frac{0.380}{0.448}$), while the NDS of BEVFormer-S (point) drops 17.3%. Compared to BEVFormer-S, BEVFormer only drops 14.3% NDS, which shows that temporal information can also improve robustness on camera extrinsics. Following [6], we show that when training BEVFormer with noisy extrinsics, BEVFormer (noise) has stronger robustness (only drops 8.9% NDS). With the spatial cross-attention based on global attention, BEVFormer (global) has a strong anti-interference ability (4.0% NDS drop) even under level 4 of the camera extrinsics noise. The reason is that we do not utilize camera extrinsics to select the RoIs for BEV queries. Notably, under the harshest noises, we see that BEVFormer-S (global) even outperforms BEVFormer-S (38.8% NDS *vs.* 38.0% NDS).

D Ablation Studies

Effect of the frame number during training. Tab. A2 shows the effect of the frame number during training. We see that the NDS on nuScenes val set keeps rising with the growth of the frame number and begins to level off the frame number ≥ 4. Therefore, we set the frame number during training to 4 by default in experiments.

Effect of some designs. Tab. A3 shows the results of several ablation studies. Comparing #1 and #4, we see that aligning history BEV features with ego-motion is important to represent the same geometry scene as current BEV queries (51.0% NDS *vs.* 51.7% NDS). Comparing #2 and #4, randomly sampling 4 frames from 5 frames is a effective data augment strategy to improve performance (51.3% NDS *vs.* 51.7% NDS). Compared to only using the BEV query to predict offsets and weights during the temporal self-attention module (see #3), using both BEV queries and history BEV features (see #4) contain more clues about the past BEV features and benefits location prediction (51.3% NDS *vs.* 51.7% NDS).

Table A2: **NDS of models on nuScenes val set using different frame numbers during training.** "#Frame" denotes the frame number during training.

#Frame	NDS↑	mAP↑	mAVE↓
1	0.448	0.375	0.802
2	0.490	0.388	0.467
3	0.510	0.410	0.423
4	**0.517**	**0.416**	0.394
5	**0.517**	0.412	**0.387**

Table A3: **Ablation Experiments on nuScenes val set.** "A." indicates aligning history BEV features with ego-motion. "R." indicates randomly sampling 4 frames from 5 continuous frames. "B." indicates using both BEV queries and history BEV features to predict offsets and weights.

#	A.	R.	B.	NDS↑	mAP↑
1	✗	✓	✓	0.510	0.410
2	✓	✗	✓	0.513	0.410
3	✓	✓	✗	0.513	0.404
4	✓	✓	✓	**0.517**	**0.416**

E Visualization

As shown in Fig. A3, we compare BEVFormer with BEVFormer-S. With temporal information, BEVFormer successfully detected two buses occluded by boards. We also show both object detection and map segmentation results in Fig. A4, where we see that the detection results and segmentation results are highly consistent. We provide more map segmentation results in Fig. A5.

Fig. A3: **Comparision of BEVFormer and BEVFormer-S on nuScenes val set.** We can observe that BEVFormer can detect highly occluded objects, and these objects are missed in the prediction results of BEVFormer-S (in red circle).

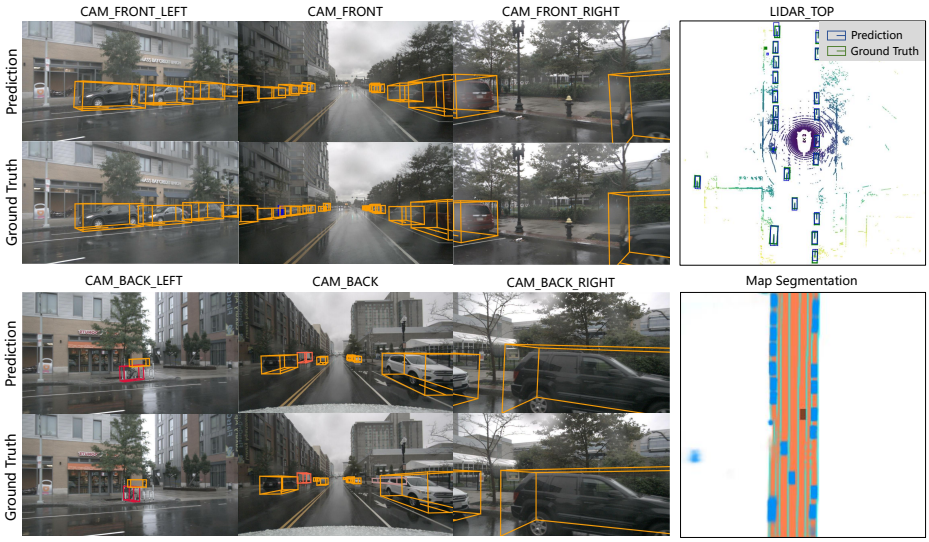

Fig. A4: **Visualization results of both object detection and map segmentation tasks.** We show vehicle, road, and lane segmentation in blue, orange, and green, respectively.

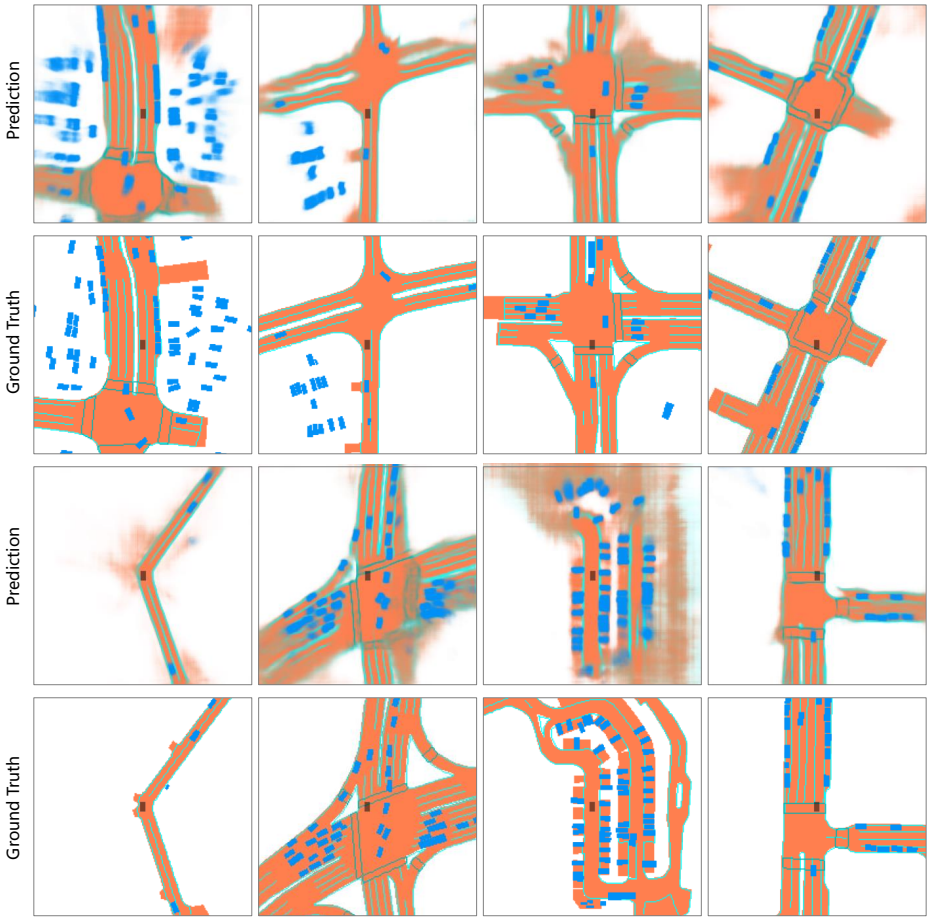

Fig. A5: **Visualization results of the map segmentation task.** We show vehicle, road, ped crossing and lane segmentation in blue, orange, cyan, and green, respectively.

References

1. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019) 3
2. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Lu, T., Luo, P.: Panoptic segformer: Delving deeper into panoptic segmentation with transformers. arXiv preprint arXiv:2109.03814 (2021) 2
3. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv: Learning (2017) 1
4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 1
5. Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-view semantic segmentation for sensing surroundings. IEEE Robotics and Automation Letters **5**(3), 4867–4873 (2020) 1
6. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision. pp. 194–210. Springer (2020) 1, 4
7. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021) 2, 3
8. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020) 2, 3
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 1, 4
10. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022) 1, 2, 3
11. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2020) 1, 3, 4