A. Further Implementation Details

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

A.1. Training Settings

In Tab. 7, we provide the hyper-parameters and training recipes of BEVformer v2 used for InternImage-B [34] and InternImage-XL backbones in Tab. 1.

Table 7. Training settings of BEVformer v2 with InternImage backbones for the main results.

backbone	InternImage-B	InternImage-XL
training epochs	24	24
batch size	16	32
optimizer	AdamW	AdamW
base learning rate	4e-4	5e-4
weight decay	0.01	0.01
lr schedule	step decay	step decay
layer-wise lr decay	0.96	0.94
warmup iters	2000	2000
warmup schedule	linear	linear
gradient clip	35	35
image size	640 × 1600	640 × 1600
image-level aug	✓	✓
temporal interval	4 seconds	4 seconds
bi-directional	✓	✓

In Tab. 3, we also construct our BEVFormer v2 detector on other backbones, including ResNet-50 [8], DLA-34 [39], ResNet-101 [8], and VoVNet-99 [13]. We list their training settings in Tab. 8.

Table 8. Training settings of BEVformer v2 with other backbones.

backbone	R50	DLA34	R101	V2-99
batch size	16			
optimizer	AdamW			
base lr	4e-4			
backbone lr	2e-4	2e-4	4e-5	4e-5
weight decay	0.01			

A.2. Network Architecture

In BEVformer v2, the image backbone yields 3 levels of feature maps of stride 8, 16, and 32. We employ FPN following the backbone to produce 5-level features of stride 8, 16, 32, 64, and 128. The perspective head takes all 5 levels of features, while the BEV head takes the first 4 levels (with stride of 8, 16, 32, and 64).

A.2.1 Perspective Head.

We adopt the single-stage anchor-free monocular 3D detector implemented by DD3D [26], which consists of three independent heads: a classification head, a 2D detection head, and a 3D detection head. The classification head produces the logit of each object category. The 2D head yields class-agnostic bounding boxes by 4 offsets from the feature location to the sides and generates the 2D center-ness. The 2D detection loss $\mathcal{L}_{2D}$ derives from FCOS [31]. The 3D head predicts the 3D bounding boxes with the following coefficients: the quotation of allocentric orientation, the depth of the box center, the offset from the feature location to the projected box center, and the size deviation from the class-specific canonical sizes. Besides, the 3D head generates the confidence of the predicted 3D box relative to the 2D confidence. It adopts the disentangles L1 loss for 3D bounding box regression and the self-supervised loss for 3D confidence in [30], denoted as $\mathcal{L}_{3D}$ and $\mathcal{L}_{conf}$ respectively. The perspective loss for BEVFormer v2 is the summation of the 2D detection loss, the 3D regression loss, and the 3D confidence loss:

$$\mathcal{L}_{pers} = \mathcal{L}_{2D} + \mathcal{L}_{3D} + \mathcal{L}_{conf} \tag{2}$$

We refer the readers to [26] for more details of the perspective detection head.

A.3. Post-Process of the First-Stage Proposals

In this section, we describe the post-processing pipeline for proposals from the perspective detection head. We start with the raw predictions of all camera views provided by the perspective head. For the i-th view in all views $\mathcal{V}$, the predicted 3D bounding boxes and their scores are denoted as $\{(\mathbf{B}_{i,j}, s_{i,j})\}_j$. We filter out the candidates with the highest score (probability) through the following post-processing pipeline. Firstly, we perform non-maximum suppression (NMS) on the proposals of each view i to obtain candidates C_i without overlapping in the perspective view:

$$C_i := \text{NMS}_{pers}\left(\{(\mathbf{B}_{i,j}, s_{i,j})\}_j\right) \tag{3}$$

The threshold of NMS is set as 2D IoU = 0.75. To ensure that objects in all camera views can be detected, we balance the numbers of proposals from different views by taking the top-k_1 of each view i after NMS:

$$C := \bigcup_{i \in \mathcal{V}} \text{top-}k_1\left(C_i\right) \tag{4}$$

We set $k_1 = 100$ in our experiments. All the 3D boxes in C are projected to the bird's-eye-view coordinate with corresponding camera extrinsics. To avoid objects that appear in multiple views causing overlapped proposals, another NMS is applied on the BEV plane with BEV IoU = 0.3:

$$C := \text{NMS}_{bev}(C) \tag{5}$$

Table 9. 3D detection results on the nuScenes *test* set of BEVFormer v2 with a COCO pre-trained VoVNet-99 and a depth pre-trained VoVNet-99. We omit the pre-training epochs for COCO since COCO pre-trained backbones are widely available and both settings use the same COCO pre-training. To isolate the effectiveness of perspective supervision, we use the non-temporal version of both detectors.

Method	Epoch	Pre-train	NDS	mAP	mATE	mASE	mAOE	mAVE	mAAE
BEVFormer v2	72	COCO	0.494	0.444	0.610	0.263	0.400	0.869	0.135
BEVFormer	15M Img + 120 + 24	COCO → DDAD15M → nuScenes	0.495	0.447	0.602	0.257	0.407	0.888	0.130

Finally, we select the top $k_2 = 100$ proposals:

$$C := \text{top-}k_2(C) \tag{6}$$

For every 3D bounding box $\mathbf{B}$ in the final set of proposals C, we use its projected center on the BEV plane, $(c_x(\mathbf{B}), c_y(\mathbf{B}))$, as the reference points for Deformable DETR in the object decoder.

B. Comparison with Depth Pre-trained Image Backbones

To demonstrate that BEVFormer v2 does not rely on domain-specific pre-training like DDAD15M [26] to achieve state-of-the-art results with large-scale image backbones, we compare BEVFormer v2 with the original BEVFormer [17] using the VoVNet-99 [13] backbone with different pre-training settsings in Tab. 9. The pre-trained VoVNet-99 backbone in the original BEVFormer consists of a COCO object detection pre-training phase, a DDAD15M depth pre-training phase as described in DD3D [26], and a nuScenes monocular 3D object detection pre-training phase. Instead of relying on this cumbersome chain of pre-trainings, our BEVFormer v2 could achieve the same detection results with an off-the-shelf COCO pre-trained backbone.

Table 10. 3D Detection Results on Waymo val set of BEVFormer v2 and other methods. We use ResNet-50 as the backbone and 1/5 split of the $train$ set. *Frames* denotes the range of frames input in the form of [−past, +future].

Method	Frames	LET-3D-APL	LET-3D-AP
BEVFormer	[0, 0]	0.331	0.474
BEVFormer v2	[0, 0]	0.347	0.495
BEVFormer	[-4, 0]	0.358	0.499
BEVFormer v2	[-4, 0]	0.377	0.523

C. Results on the Waymo Open dataset.

To further confirm the generalization to different datasets and the robustness to hyper parameter selection, we train BEVFormer v2 on the Waymo dataset *with architecture and hyper-parameters identical to the nuScenes dataset* and compare it with the original BEVFormer [17]. As shown in Tab. 10, BEVFormer v2 improves on BEVFormer significantly for both the single- and the multi-frame settings.

D. Visualization

We demonstrate visualization for 3D object detection results of our BEVFormer v2 detector in Fig. 4. Our model predicts accurate 3D bounding boxes for the target objects, even for the hard cases in the distance or with occlusion. For instance, our model successfully detects the distant pedestrian in the front-right camera, the truck overlapped with multiple cars in the back camera, and the bicycle occluded by the tree in the back-right camera.

Figure 4. Visualization of BEVFormer v2 3D object detection predictions.