FB-BEV: BEV Representation from Forward-Backward View Transformations

Zhiqi Li[1,2*] Zhiding Yu[2] Wenhai Wang[3] Anima Anandkumar[2,4] Tong Lu[1] Jose M. Alvarez[2]
[1]National Key Lab for Novel Software Technology, Nanjing University [2]NVIDIA
[3]The Chinese University of Hong Kong [4]Caltech

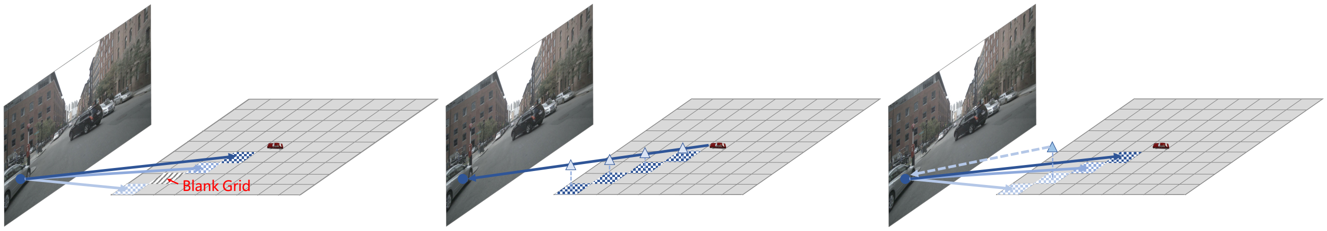


Figure 1: **Left**: Forward projection lifts the image features from 2D to BEV space and weight them based on depth (coded in different colors). However, forward projection tends to generate sparse BEV projection. **Middle**: Backward projection first defines the voxel locations in the 3D space and then projects these points onto the 2D image planes. Dense BEV features can be generated but the points on the projection ray fetch the same features without distinction. **Right**: Forward-backward projection proposed in this work. We use backward projection to refine the necessary BEV grids with reduced sparsity. We further introduce depth consistency into backward projection and assign each projection a different weight (dashed line).

Abstract

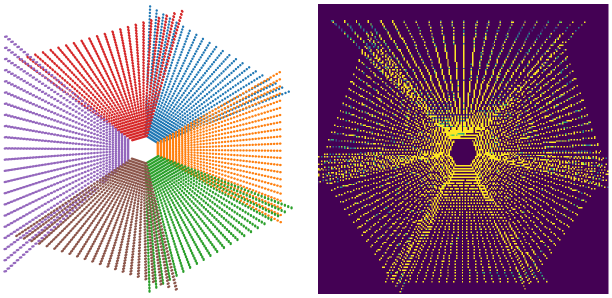
View Transformation Module (VTM), where transformations happen between multi-view image features and Bird-Eye-View (BEV) representation, is a crucial step in camera-based BEV perception systems. Currently, the two most prominent VTM paradigms are forward projection and backward projection. Forward projection, represented by Lift-Splat-Shoot, leads to sparsely projected BEV features without post-processing. Backward projection, with BEV-Former being an example, tends to generate false-positive BEV features from incorrect projections due to the lack of utilization on depth. To address the above limitations, we propose a novel forward-backward view transformation module. Our approach compensates for the deficiencies in both existing methods, allowing them to enhance each other to obtain higher quality BEV representations mutually. We instantiate the proposed module with FB-BEV, which achieves a new state-of-the-art result of 62.4% NDS on the nuScenes test set. Code and models are available at `https://github.com/NVlabs/FB-BEV`.

BEV-based 3D detection models have gained popularity due to their unified and comprehensive representation abilities for multi-camera inputs, enhancing the performance of both vision-only and multi-modality perception models for autonomous driving [1–9]. A typical BEV-based detection model comprises an image backbone, a View Transformation Module (VTM), and a detection head. The VTMs primarily function to project multi-view camera features onto the BEV plane. There are two main categories of existing mainstream VTMs based on the projection methods used: forward projection and backward projection.

1. Introduction

Forward projection. The most intuitive method for projecting camera features onto the BEV plane involves estimating the depth value of each pixel in the image and using the camera calibration parameters to determine the corresponding position of each pixel in 3D space [10], as shown in Figure 1 (left). We refer to this process as forward projection, where the 2D pixels take the initiative in projection and the 3D space passively accepts features from the images. The accuracy of the predicted depth for each pixel is critical to achieving high-quality BEV features. However, accurately estimating the depth value of each pixel is challenging [11]. To address this challenge, Lift-Splat-Shoot (LSS) pioneered the use of depth distribution to model the uncertainty of each pixel's depth [1]. One limitation of

[*] Work done during an internship at NVIDIA.



(a)

(b)

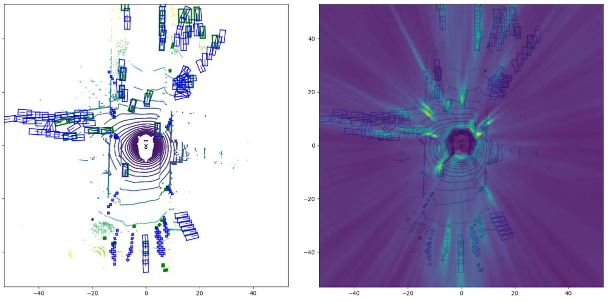
Figure 2: (a) Projection points on BEV plane with forward projection on nuScenes dataset, and different colors represent different cameras. (b) BEV feature map of LSS [1] with a shape of 200×200. We can observe that forward projection has an extremely low utilization rate for BEV space.

LSS is that it generates discrete and sparse BEV representation [1, 12]. As shown in Figure 2, the density of BEV features decreases with distance. When using the default settings of LSS on the nuScenes dataset, only 50% of the grids can receive valid image features through projection.

Backward projection. The motivation behind backward projection is opposite to that of forward projection. For the backward projection paradigm, the points in 3D space take the initiative [2, 3, 8, 13]. For instance, BEVFormer sets the coordinates of the 3D space to be filled in advance and then projects these 3D points back onto the 2D image [3], as shown in Figure 1 (middle). As a result, each predefined 3D space position can obtain its corresponding image features. The BEV representation obtained by this method is denser than that of LSS, with each BEV grid filled with the corresponding image features.

The drawbacks of backward projection are also apparent, as shown in Figure 3. Although yielding a denser BEV representation, it comes at the cost of establishing numerous false correspondences between 3D and 2D space due to occlusion and depth mismatch [14]. The absence of depth information during the projection process is the main cause. Without depth as a reference, each 3D coordinate on the ray is equally related to the same 2D coordinate, equivalent to having a uniform depth distribution for this pixel in forward projection. As a result, the distance prediction of the objects along the longitudinal direction become ambiguous. Backward projection thus tends to be inferior to forward projection in depth utilization. Recently, the advantage of forward projection has been further highlighted since more accurate depth distribution obtained from depth supervision is shown to improve 3D perception [15, 16].

Considering the pros and cons discussed above, we propose forward-backward view transformation to address the limitations of existing VTMs, as shown in Figure 1 (right).



(a)

(b)

Figure 3: (a) Detection of BEVFormer. (b) The corresponding BEV features of BEVFormer. Since BEVFormer cannot use depth for distinction, the features of each object on the BEV tend to be ray-shaped. The model thus predicts multiple boxes for one object along the longitudinal direction.

To address the issue of sparse BEV representations in forward projection, we leverage backward projection to refine the sparse region from forward projection. Meanwhile, backward projection is prone to false-positive features due to the lack of depth guidance. We thus propose a depth-aware backward projection design to suppress false-positive features by measuring the quality of each projection relationship through depth consistency. The depth consistency is determined by the distance of depth distributions between a 3D point and its corresponding 2D projection point. Using this depth-aware method, unmatched projections are given lower weights, which reduces the interference caused by false-positive BEV features. In addition, for the objection detection task, we only care about foreground objects, so we densify only the foreground regions of the BEV plane while using backward projection. This not only reduces the computational burden but also avoids the introduction of false-positive features in the background areas. With the sparse regions refined for forward projection and false-positives features reduced for backward projection, our forward-backward projection not only solves the defects of existing projection methods but also realizes the effective ensemble of existing projection methods. Our contributions can be summarized as follows:

- We propose a forward-backward projection strategy that generates dense BEV features with strong representation ability through bidirectional projection. Our approach addresses the limitations of existing projection methods, which result in either sparse BEV features or false-positive features caused by inaccurate projection.
- To address the pitfalls of existing forward projection methods for producing sparse BEV representations, we employ backward projection to refine the blank grid that not be activated by forward projection. This makes the model more suitable for large-scale BEVs.

- We propose a novel depth-aware backward projection method that overcomes the limitations of existing methods in effectively utilizing depth information. Our approach integrates depth consistency into the projection process to establish a more accurate mapping relationship between the 3D and 2D spaces.

- Our FB-BEV model has been extensively evaluated on the nuScenes dataset. The results demonstrate that it outperforms other methods for camera-based 3D object detection and achieves the state-of-the-art 62.4% NDS on the nuScenes *test* set.

2. Related work

We introduce related BEV perception works according to the VTMs they use.

2.1. Forward Projection Methods

The Lift-Splat-Shoot (LSS) [1] method is the archetypal technique of this category. LSS utilizes a depth distribution to model depth uncertainty and project multi-view features into the same Bird's Eye View (BEV) space. Subsequent methods have largely adhered to this paradigm. For instance, BEVDet [9] applies this forward projection approach to the field of multi-view 3D detection. CaDDN [17] and BEVDepth [15] proposes the use of LiDAR point clouds to generate depth ground truth for supervising the depth prediction module. BEVDepth demonstrates that an accurate depth prediction module can significantly enhance model performance. Similarly, BEVstereo [16] further underscores the importance of precise depth estimation to model performance. Furthermore, BEVFusion [4] extends this paradigm to the multi-modality perception domain and improves the projection efficiency of the LSS paradigm. The most notable disadvantage of VTM in LSS is low efficiency. Subsequent research has made significant progress in improving efficiency through engineering implementation [4, 9, 15]. In response to the sparseness of BEV features, MatrixVT [12] mainly focuses on improving the calculation efficiency in the process of BEV generation, rather than densely stressing BEV features.

2.2. Backward Projection Methods

OFT [13] is among the first methods to adapt the backward projection paradigm. This paradigm does not involve complex accumulation in 3D space [1], which is the least efficient step in forward projection. Subsequent works such as ImVoxelNet [8] and M^{2}BEV [18] extend this paradigm from monocular to multi-view perception where 3D space is divided into voxels. DETR3D [2] does not introduce dense BEV features, but performs end-to-end learning of object queries in 3D and projects object centers back to image space. BEVFormer [3] aggregates features at different heights on the BEV space without introducing voxelized representation, therefore reducing the resource consumption. BEVFormer also introduces deformable sampling points and temporal features, promoting further development of camera-based perception. For the perception heads, BEVFormer adopts Deformable DETR [19] and Panoptic SegFormer [20]. BEVFormerV2 [21] further exploits the potential of backward projection by adapting the modern image backbone via perspective supervision. PolarFormer [22] and PolarDETR [23] adopt polar coordinates rather than Cartesian coordinates to conduct the projection process. Methods [24–26] project 3D anchors onto 4D features rather than 3D features. PersFormer [27] uses Inverse Perspective Mapping (IPM) to guide the projection point on the image space. However, existing methods seldom consider introducing depth in the projection process or even consider getting rid of the dependence on depth as an advantage [2, 3, 18]. We argue that it increases ambiguity in the projection process without depth to measure the quality of the projection.

In addition to different view projection paradigms, researchers have also explored using longer temporal information to enhance the spatial perception capacity. [28–31].

2.3. Projection-Free Methods

In addition to the above two paradigms, some methods can generate BEV representations without relying on projections. PETR [32] and PETRv2 [33] implicitly learn the view transformation through global attention and use camera parameters to encode position features. CFT [34] uses view-aware attention to adaptively learn the BEV features required for each view, and even get rid of the dependence on camera calibration parameters. BEVSegFormer [35] automatically learns the correspondence between 3D and 2D space without relying on the projection process.

3. Method

To address the limitations of existing view transformation modules, we propose a novel Forward-Backward View Transformation method named FB-BEV. FB-BEV employs a two-pronged approach. Firstly, a VTM based on forward projection will generate an initial sparse BEV representation. To obtain a denser BEV representation while minimizing the computational burden, a foreground region proposal network is employed to select the foreground BEV grids. Subsequently, another VTM utilizes these foreground grids as BEV queries and refines them by projecting them back onto the images with a depth-aware mechanism.

3.1. Overall Architecture

As illustrated in Figure 4, FB-BEV mainly consists of three key modules: a view transformation module with for-

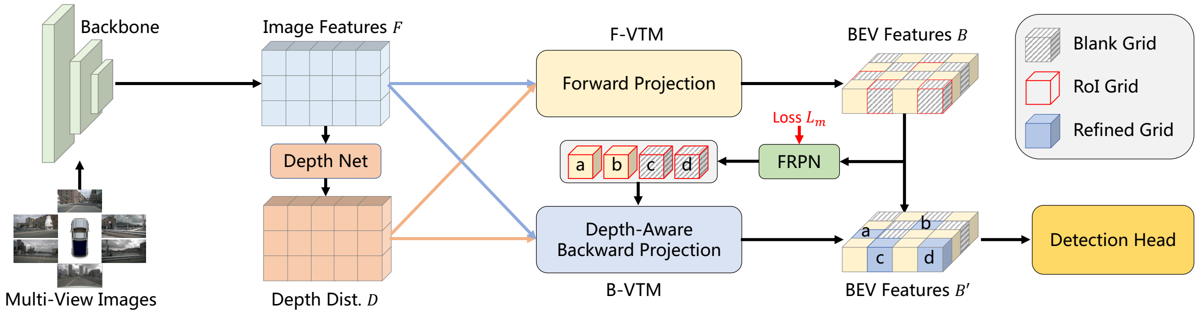


Figure 4: Overview of FB-BEV. We first extract multi-view features from the 2D backbone and generate the depth distribution using a depth network. We then employ a forward projection module to generate the BEV features B. Since BEV features B contain blank grids, FRPN generates a foreground mask and feeds foreground region of interest (RoI) grids to the next depth-aware backward projection module (Grids $\{a, b, c, d\}$ in the figure). Our depth-aware backward projection module uses RoI grids as BEV queries and refines these queries by projection them back onto images with a depth consistency mechanism. Finally, we obtain the BEV features B' by adding the refined grids and BEV features B.

ward projection denoted as F-VTM, a foreground region proposal network denoted as FRPN, and a view transformation module with depth-aware backward projection denoted as B-VTM. In addition, we have a depth net to predict the depth distributions, and the distributions will be utilized in both VTMs. F-VTM generates a complete BEV representation from the multi-view features by projecting each pixel feature into the 3D space based on the corresponding depth distribution. FRPN is a lightweight binarized mask predictor used to select the regions where the foreground object is located. B-VTM is only responsible for optimizing BEV grids located in the foreground region generated by FRPN.

During inference, we feed multi-view RGB images to the image backbone network and obtain the image features $F = \{F_i\}_{i=1}^{N_c}$, where F_i is the view features of i-th camera view and N_c is the total number of cameras. Then we obtain the depth distributions $D = \{D_i\}_{i=1}^{N_c}$ by feeding image feature F into depth net. Taking the view feature F and depth distribution D as input, the F-VTM will generate a BEV representation $B \in \mathbb{R}^{C\times H\times W}$, where C is the channel dimension, and $H \times W$ is the spatial shape of BEV. FRPN takes BEV features B as input and predicts a binary mask $M \in \mathbb{R}^{H\times W}$ to detect foreground regions. Only foreground grid $B[\text{sigmoid}(M) > t_f]$ will be fed in B-VTM to be further refined, where t_f is the foreground threshold. The final BEV features $B' \in \mathbb{R}^{C\times H\times W}$ are obtained by adding the refined BEV features from B-VTM to B back. Finally, we perform 3D detection task based on the BEV features B'.

3.2. Forward Projection

Our forward projection module F-VTM follows the paradigm of LSS [1]. Lift and Splat are two fundamental steps in modern forward projection techniques used for view transformation. The Lift step projects each pixel in the 2D image onto the 3D voxel space based on its corresponding depth distribution. The Splat step aggregates the feature values of pixels within each voxel by sum pooling. For specific implementation, our F-VTM is based on BEVDet [9, 36] and BEVDepth [15], which represent the current state-of-the-art design of forward projection. We denote the BEV features from F-VTM as B.

3.3. Foreground Region Proposal Network

The BEV features obtained from F-VTM are sparse, and there are BEV grids that are not activated and thus contain blank information. To obtain a stronger BEV representation, we expect to fill in these blank BEV grids. However, for 3D object detection, our interest lies only in the limited foreground objects that occupy a relatively small fraction of the BEV features. To locate these foreground objects within the BEV features, we utilize a simple segmentation network to generate a binary mask $M \in \mathbb{R}^{H\times W}$ from the BEV feature B. The FRPN employed in this process comprises a 3×3 convolutional layer followed by a sigmoid function, rendering it exceptionally lightweight. The ground truth for this binary mask, M^{gt}, is derived by projecting the foreground objects onto the BEV plane. In this paper, we use a combination of Dice loss [37] and cross-entropy loss to supervise the FRPN.

During the inference phase, with BEV feature B from F-VTM as input and the predicted binary mask M, we filter out unnecessary BEV grids with a mask logit lower than threshold t_f. Thus we obtain a set of discrete BEV grids $\{Q_{x,y} | M[(x, y)] > t_f\}$, where (x, y) is the location of each foreground BEV grid. Each BEV grid $Q_{x,y}$ can be seen as a BEV query that requires further refinement. To maintain feature consistency in the foreground area, we have selected BEV grids that contain both blank and non-blank grids.

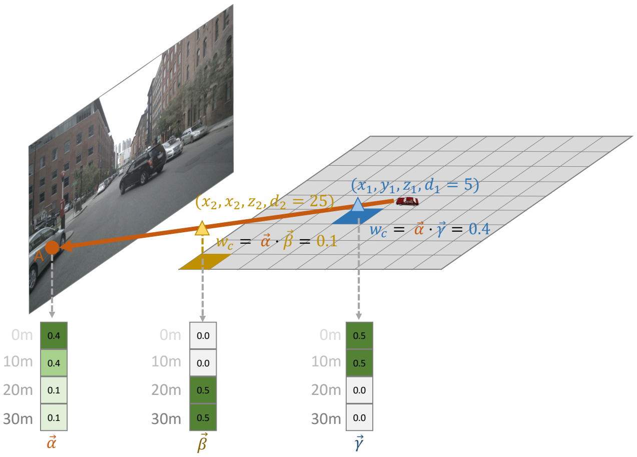

$(x_2, x_2, z_2, d_2 = 25)$
$(x_1, y_1, z_1, d_1 = 5)$
$w_c = \vec{\alpha} \cdot \vec{\beta} = 0.1$
$w_c = \vec{\alpha} \cdot \vec{\gamma} = 0.4$
A
0m
10m
20m
30m
0.4
0.4
0.1
0.1
$\vec{\alpha}$
0m
10m
20m
30m
0.0
0.0
0.5
0.5
$\vec{\beta}$
0m
10m
20m
30m
0.5
0.5
0.0
0.0
$\vec{\gamma}$

Figure 5: Depth-aware backward projection uses depth consistency to distinguish features on projection rays. For instance, points (x_1, y_1, z_1) and (x_2, y_2, z_2) are located on the same ray, and have the same projection point A on the image. The depth values of the two points are $d_1 = 5$m and $d_2 = 25$m, respectively. We then convert d_1 and d_2 to depth distribution $\vec{\beta}$ and $\vec{\gamma}$. Assuming the predicted depth distribution of A is $\vec{\alpha}$, the depth consistency can be computed as $\vec{\alpha} \cdot \vec{\gamma} = 0.4$ and $\vec{\alpha} \cdot \vec{\beta} = 0.1$. The closer point (x_1, y_1, z_1) thus owns a higher feature weight with a higher consistency.

3.4. Depth-Aware Backward Projection

The depth-aware backward projection module serves a dual purpose. Firstly, it effectively fills the BEV with arbitrary resolution and can choose only to generate BEV features of specified regions, thereby compensating for the sparse features generated by forward features. Secondly, when combined with a forward projection method, they provide a more comprehensive BEV representation. In this section, we first introduce the depth consistency used to improve the quality of backward projection in 3.4.1 and then introduce our detailed implementation in 3.4.2

3.4.1 Depth Consistency

The fundamental concept of backward projection involves projecting a 3D point (x, y, z) onto a 2D image point (u, v), based on the camera projection matrix $P \in \mathbb{R}^{3\times4}$. This process can be expressed mathematically as:

$$d \cdot \begin{bmatrix} u & v & 1 \end{bmatrix}^T = P \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T, \qquad (1)$$

where d represents the depth of the 3D point (x, y, z) on the image. Notably, for any 3D point $(\lambda x, \lambda y, \lambda z)$, where $\lambda \in \mathbb{R}^+$, they share the same projected point (u, v) on the 2D image. Consequently, these 3D points $(\lambda x, \lambda y, \lambda z)$ exhibit similar image features, as shown in Figure 3.

Forward projection alleviates this problem by predicting different weights for different depths. Specifically, for each point (u, v), it predict a weight w_i for each discrete depth $(d_0 + i\Delta)$, and $i \in \{0, 1, \cdots, |D|\}$, D is a set of discrete depths, d_0 is the initial depth and Δ is the depth interval. Thus, while considering two discrete depth $(d_0 + i\Delta)$ and $(d_0 + j\Delta)$ on point (u, v), it falls onto the 3D points (x_i, y_i, z_i) and (x_j, y_j, z_j) based on Equation 1. The forward projection method leverages predicted depth weights w_i and w_j to generate distinguishing features.

To incorporate depth in backward projection and enhance the projection quality, this paper introduces depth consistency w_c, as shown in Figure 5. Equation 1 shows that a 3D point (x, y, z) has a corresponding depth $d \in \mathbb{R}^+$ on the projected point (u, v). Since a discrete depth distribution vector $[w_0, w_1, \cdots, w_{|D|}]$ on point (u, v) is already available. The depth consistency w_c of the depth value d and this depth distribution vector can be computed by converting d into depth distribution vector $[w'_0, \cdots, w'_i, w'_{i+1}, \cdots w'_{|D|}]$, where only w'_i and w'_{i+1} are non zero, and $(d_0 + i\Delta) \leq d \leq (d_0 + (i+1)\Delta)$. The depth consistency w_c can be computed as:

$$\begin{aligned} w_c &= [w_0, w_1, \cdots, w_{|D|}] \cdot [w'_0, w'_1, \cdots, w'_{|D|}] \\ &= w_i w'_i + w_{i+1} w'_{i+1}, \end{aligned} \qquad (2)$$

where $w'_i = 1 - \frac{d - d_0 - i\Delta}{\Delta}$ and $w'_{i+1} = 1 - w'_i$. The depth consistency introduced in this paper serves a similar role as the depth weight in forward projection. It is worth mentioning that we obtain the depth distribution of point (u, v) via bilinear projection.

Forward projection employs discrete depth values to generate corresponding discrete 3D projection points in 3D space. While sacrificing continuity in depth, the accuracy of BEV features by forward projection is also affected. For our depth-aware backward projection, we guarantee the ability to densely fill 3D space at arbitrary resolutions, while leveraging depth consistency to guarantee projection quality.

3.4.2 Implementation

Depth consistency is a general mechanism that can be plugged into any existing backward projection method. In this paper, our depth-aware backward projection is based on the spatial cross-attention in BEVFormer [3]. The projection process of the original Spatial Cross-Attention (SCA) of BEVFormer can be formulated as:

$$\text{SCA}(Q_{x,y}, F) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_{\text{ref}}} \mathcal{F}_d(Q_{x,y}, \mathcal{P}_i(x, y, z_j), F_i), \quad (3)$$

where $Q_{x,y}$ is one BEV query that located at (x, y) and F are multi-view features. For each point (x, y) on the BEV plane, BEVFormer will lift this point up to N_{ref} 3D points with different heights z_i. Projection function will get the projection point (u_i, v_i) on i-th image based on Equation 1. $\mathcal{F}_d$ is the deformable attention function [19] that using query $Q_{x,y}$ to sample features of the projection point $\mathcal{P}_i(x,y,z_j)$ on image feature F_i.

By using the depth consistency of this paper, we can directly evolve SCA into Depth-Aware SCA (SCA_{da}) by:

$$\text{SCA}_{da}(Q_{x,y},F)=\sum_{i=1}^{N_c}\sum_{j=1}^{N_{\text{ref}}}\mathcal{F}_d(Q_{x,y},\mathcal{P}_i(x,y,z_j),F_i)\cdot\boldsymbol{w_c^{ij}},\tag{4}$$

where w_c^{ij} is the depth consistency between 3D point (x,y,z_j) and 2D point (u_i,v_i). Compared to the original SCA, our proposed SCA_{da} is capable of generating more discriminative BEV features along the longitudinal direction. Due to the high efficiency of our depth-aware SCA, we only use back projection once instead of stacking 6 layers used by the original BEVFormer [3].

4. Experiments

4.1. Experimental Setup

Dataset and Metrics. The nuScenes dataset [38] is a large-scale autonomous driving dataset, which contains 1000 diverse scenarios. Key samples of each scenario are annotated at 2Hz, and each sample includes RGB images from 6 cameras. NuScenes provides a series of metrics to measure the quality of 3D detection, including nuScenes Detection Score (NDS), mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE) and mean Average Attribute Error (mAAE). NDS is a composite metric and defines as $\text{NDS}=\frac{1}{10}[5\text{mAP}+\sum_{\text{mTP}\in\mathbb{TP}}(1-\min(1,\text{mTP}))]$.

Implementation Details. By adhering to common practices [9, 15, 36], we default to using ResNet-50 [39] and an image size of 256×704. During training, we adopt the CBGS strategy [40] and apply data augmentations at both the image and BEV levels, which include random scaling, flipping, and rotation as per BEVDet [9]. By default, our model is trained for 20 epochs using a batch size of 64 and the AdamW [41] optimizer with a learning rate of 2e-4. While training FB-BEV with V2-99 backbone for *test* set, we train the model with 30 epochs without CBGS. For training the depth net with temporal information, we use the camera-aware Depth Net in BEVDepth [15] with a total of 118 depth categories ($|D|$), $d_0=1$ and $\Delta=0.5$. Incorporating depth-aware spatial cross-attention, we sample the predefined heights uniformly from [-5m, 3m], use 8 attention heads, and set $N_{\text{ref}}=4$. The spatial shape of BEV grids is, by default, 128×128 with a channel dimension of 256. The threshold for the foreground mask is set to $t_f=0.4$. When introducing temporal information, we stack the BEV features of two adjacent keyframes, as done in BEVDet4D [36] for *val* set, and 9 previous keyframes for *test* set.

4.2. Baselines

To assess the efficacy of our novel approach, FB-BEV, we conduct comparisons with two types of baselines that solely rely on forward and backward projection techniques, respectively. Notably, for these baselines, we maintain consistency with FB-BEV in terms of backbone, detection head, and training strategy, with the exception of the view transformation module. We reduce the number of channels and layers of FB-BEV to match the computational cost.

Forward Projection. For forward projection methods, we adopt BEVDet [9, 36] and BEVDepth [15] as our baseline. Compared to BEVDet, BEVDepth uses point clouds to generate the ground truth of depth and train the depth net with the ground truth of depth.

Backward Projection. For backward projection methods, we choose BEVFormer [3] as the baseline. Considering the difference in implementation details, we ported the view transformation module of BEVFormer to BEVDet for a fair comparison. It is worth mentioning that we discard the temporal self-attention module in BEVFormer. In this paper, we note the BEVFormer that with temporal information as BEVFormer-T.

4.3. Benchmark Results

Table 1 shows the 3D detection results on the nuScenes `val` set for our proposed FB-BEV method, as well as the two baseline methods BEVDet [9] and BEVFormer [3], and other previous state-of-the-art 3D detection methods. Without using temporal information or depth supervision, our method outperforms BEVDet and BEVFormer by a significant margin of 2.4% NDS and 2.7% NDS. When introducing temporal information by stacking historical BEV features, our proposed FB-BEV still outperforms BEVDet and BEVFormer by 1.3 points. With depth supervision, our method achieves a lead of more than 1.5 points over BEVDepth. However, as the previous backward projection cannot use depth information, BEVFormer-T only brings a marginal improvement of 0.2% NDS when only using depth supervision as an auxiliary task. This confirms the limitations of existing backward projection methods. Despite achieving higher performance, our method still maintains a comparable or even lower computational cost than our baselines. As shown in Table 2, our model obtains a new state-of-the-art 62.4% NDS and outperforms previous SOLOFusion [28] with a clear margin of 0.5 points.

4.4. Ablation Studies

Depth-aware Backward Projection. In Table 3, we compare the results of adopting depth-aware backward projection in FB-BEV and BEVFormer. BEVFormer obtains an improvement of 0.9% NDS with depth-aware projection. When using depth supervision as an auxiliary task,

Table 1: Comparison on the nuScenes `val` set. *: Baseline methods for a fair comparison. While using depth supervision in BEVFormer, we set training depth estimation as an auxiliary task. †: Since our channel dimension from the backbone is 256, which is half of BEVDet4D*/BEVDepth*. Thus our model is much lighter than BEVDet4D* and BEVDepth* while using the camera-aware depth net.

Methods	Backbone	Image Size	Temporal	Depth Sup.	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	Param.	Flops
PETR [32]	R50	384 × 1056	✗	✗	0.313	0.381	0.768	0.278	0.564	0.923	0.225	-	-
BEVDet [9]	R50	256 × 704	✗	✗	0.298	0.379	0.725	0.279	0.589	0.860	0.245	-	
BEVDet*	R50	256 × 704	✗	✗	0.307	0.382	0.722	0.278	0.606	0.876	0.235	55.7	184
BEVFormer*	R50	256 × 704	✗	✗	0.297	0.379	0.739	0.281	0.601	0.833	0.242	59.7	216
FB-BEV (ours)	R50	256 × 704	✗	✗	0.312	0.406	0.702	0.275	0.518	0.777	0.227	58.4	192
BEVDet4D [36]	R50	256 × 704	✓	✗	0.322	0.457	0.703	0.278	0.495	0.354	0.206	-	-
BEVDet4D*	R50	256 × 704	✓	✗	0.344	0.466	0.670	0.273	0.523	0.400	0.194	83.4†	296
BEVFormer-T*	R50	256 × 704	✓	✗	0.330	0.459	0.686	0.272	0.482	0.417	0.201	66.9	249
FB-BEV (ours)	R50	256 × 704	✓	✗	0.350	0.479	0.642	0.275	0.459	0.391	0.193	65.7	225
STS [42]	R50	256 × 704	✓	✓	0.377	0.489	0.601	0.275	0.450	0.446	0.212	-	-
BEVDepth [15]	R50	256 × 704	✓	✓	0.351	0.475	0.639	0.267	0.479	0.428	0.198	-	-
BEVDepth*	R50	256 × 704	✓	✓	0.370	0.484	0.611	0.271	0.493	0.423	0.211	83.4†	292
BEVFormer-T*	R50	256 × 704	✓	✓	0.343	0.461	0.680	0.274	0.519	0.426	0.204	66.9	249
FB-BEV (ours)	R50	256 × 704	✓	✓	0.378	0.498	0.620	0.273	0.444	0.374	0.200	65.7	225

Table 2: Comparison on the nuScenes `test` set. Extra data is depth pertaining. V2-99 [11, 43] uses extra data for depth training. Swin-B [44] and ConvNeXt-B [45] are trained with ImageNet-22K [46].

Methods	Backbone	Image Size	Test-Time Aug	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
FCOS3D [47]	R101-D	900×1600	✓	0.358	0.428	0.690	0.249	0.452	1.434	0.124
DETR3D [2]	V2-99	900×1600	✓	0.412	0.479	0.641	0.255	0.394	0.845	0.133
UVTR [48]	V2-99	900×1600	✗	0.472	0.551	0.577	0.253	0.391	0.508	0.123
BEVFormer [3]	V2-99	900×1600	✗	0.481	0.569	0.582	0.256	0.375	0.378	0.126
BEVDet4D [36]	Swin-B	900×1600	✓	0.451	0.569	0.511	**0.241**	0.386	0.301	0.121
PolarFormer [22]	V2-99	900×1600	✗	0.493	0.572	0.556	0.256	0.364	0.439	0.127
PETRv2 [33]	V2-99	640×1600	✗	0.490	0.582	0.561	0.243	0.361	0.343	**0.120**
BEVStereo [16]	V2-99	640×1600	✗	0.525	0.610	**0.431**	0.246	**0.358**	0.357	0.138
BEVDepth [15]	V2-99	640×1600	✗	0.503	0.600	0.445	0.245	0.378	0.320	0.126
SOLOFusion [28]	ConvNeXt-B	640×1600	✗	**0.540**	0.619	0.453	0.257	0.376	0.276	0.148
FB-BEV (ours)	V2-99	640×1600	✗	0.537	**0.624**	0.439	0.250	**0.358**	**0.270**	0.128

BEVFormer-T achieves a larger gain of 1.1% NDS. Without depth-aware backward projection in FB-BEV, the performance drops by about 0.9% NDS. In the past, only forward projection methods could benefit from more accurate depth prediction. With depth consistency, backward methods can also improve performance by leveraging accurate depth prediction.

Figure 6 presents visual results of FB-BEV with and without depth-aware backward projection. When the depth-aware projection is not employed, the model tends to produce incorrect results along the longitudinal direction due to depth ambiguities, as seen in the yellow boxes in Figure 6 (b). In addition, Figure 6 (c) and (d) show the depth consistency on the BEV plane for FB-BEV with and without depth-aware projection. Foreground grids exhibit higher depth consistency, which prevents background regions from erroneously identifying false foreground features. Moreover, Figure 6 (d) shows that the depth consistency varies with height for the same location (x, y) on the BEV plane. Prior backward projection methods aggregated features at all heights, resulting in feature interference. However, with our proposed depth-aware backward projection, the model selectively aggregates features based on depth consistency at different heights. These visualizations provide compelling evidence for the effectiveness of our method.

Effect of FRPN. We employ FRPN to selectively optimize the foreground grids in BEV feature B through B-VTM. To study its effectiveness, we conduct an experiment where we exclude FRPN and instead feed all BEV features into B-VTM. Results in Table 4 demonstrate that FRPN not only improves the inference efficiency but also improves the detection accuracy. In Figure 7, we present the depth consistency map of FB-BEV with and without FRPN. Without using FRPN, depth-aware backward projectionmay focus on some background regions due to imprecise depth predictions. On the other hand, using the foreground mask provided by FRPN, the model can selectively concentrate only on the foreground objects, thus avoiding interference from the background regions.

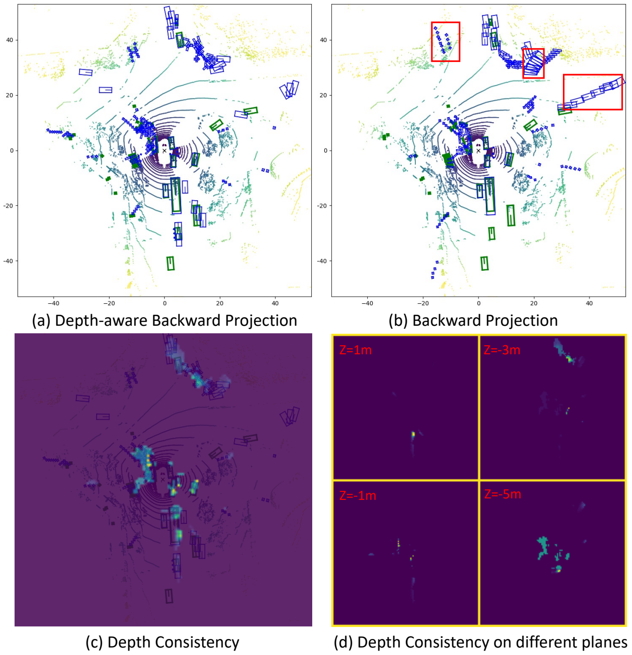
Effect of Reducing Sparsity. Due to the fixed discrete depth values, the forward projection method generates fixed discrete 3D projection points (projection matrix P remains unchanged). As the BEV scale increases, the proportion of blank grids on BEV generated by forward projection will also increase. The rate of blank gird of BEVDet with a BEV scale 400×400 and input shape 256×704 is 80.5%. Thus we can observe that BEVDet performance drops on

Table 3: Effect of depth in backward projection.

Methods	Backbone	Image Size	Temporal	Depth Sup.	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
BEVFormer	R50	256×704	✗	✗	0.297	0.379	0.739	0.281	0.601	0.833	0.242
w/ depth-aware	R50	256×704	✗	✗	0.291	0.387	0.740	0.282	0.548	0.806	0.225
BEVFormer	R50	256×704	✓	✓	0.343	0.461	0.680	0.274	0.519	0.426	0.204
w/ depth-aware	R50	256×704	✓	✓	0.350	0.472	0.665	0.281	0.499	0.390	0.194
FB-BEV	R50	256×704	✗	✗	0.312	0.406	0.702	0.275	0.518	0.777	0.227
w/o depth-aware	R50	256×704	✗	✗	0.305	0.397	0.726	0.278	0.552	0.779	0.227
FB-BEV	R50	256×704	✓	✓	0.378	0.498	0.620	0.273	0.444	0.374	0.200
w/o depth-aware	R50	256×704	✓	✓	0.367	0.489	0.629	0.273	0.458	0.382	0.196

Table 4: Effect of FRPN. With FRPN, FB-BEV obtains higher accuracy and faster inference. The latency of VTM is smaller due to only refining the foreground BEV grids.

Methods	Temporal	Depth Sup.	mAP↑	NDS↑	Latency
FB-BEV	✗	✗	0.312	0.406	2.6ms
w/o FRPN	✗	✗	0.308	0.400	3.4ms
FB-BEV	✓	✓	0.378	0.498	2.6ms
w/o FRPN	✓	✓	0.373	0.494	3.4ms



(a) Depth-aware Backward Projection

(b) Backward Projection

(c) Depth Consistency

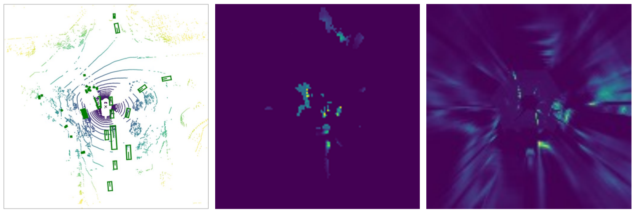
(d) Depth Consistency on different planes

Figure 6: (a)/(b) Comparison of FB-BEV back projection with and without depth consistency, respectively. The red boxes in (b) indicate erroneous results produced by the model along the longitudinal direction due to depth ambiguities. (c) shows the depth consistency map on the BEV plane, where each value is the sum of depth consistency at different heights. In (d), we observe the depth consistency map at different heights.

large-scale BEV. Our forward-backward projection fixes it by filling these blank grids, then obtains continuous performance gains. In addition, we can observe that current VTMs are highly efficient and are not considered a potential bottleneck against inference efficiency.

Table 5: Ablation on the Effect of BEV scale. All results are trained for 12 epochs with CBGS. We also show the latency of VTM during inference.

Methods	BEV Size	mAP↑	NDS↑	Latency
BEVDet	128×128	0.304	0.370	0.7ms
FB-BEV	128×128	0.309	0.396	2.6ms
BEVDet	256×256	0.309	0.375	0.8ms
FB-BEV	256×256	0.322	0.404	3.4ms
BEVDet	400×400	0.302	0.368	1.5ms
FB-BEV	400×400	0.325	0.406	6.6ms



(a) ground truth (b) depth consistency w/ FRPN (c) depth consistency w/o FRPN

Figure 7: We compare the depth consistency map of FB-BEV with FRPN or not in (b) and (c). In the absence of FRPN, depth-aware backward projection may still attend to certain background regions due to the presence of inaccurate depth prediction. On the other hand, FRPN utilizes the foreground mask to enable the model to concentrate solely on the foreground objects.

5. Conclusion

We present a forward-backward projection paradigm to address the limitations of current projection schemes. Our approach addresses the issue of sparse features generated by forward projection and introduces depth into backward projection to establish a more precise projection relationship. This two-stage VTM strategy is suitable for higher-resolution BEV perception and has application prospects for ultra-long-distance object detection or high-resolution occupancy perception.

References

[1] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, 2020. 1, 2, 3, 4

[2] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 2022. 1, 2, 3, 7

[3] 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 preprint arXiv:2203.17270*, 2022. 1, 2, 3, 5, 6, 7

[4] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 3

[5] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. BEVFusion: A simple and robust lidar-camera fusion framework. *arXiv preprint arXiv:2205.13790*, 2022. 1

[6] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust lidar-camera fusion for 3d object detection with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1

[7] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022. 1

[8] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022. 1, 2, 3

[9] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. BEVDet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 3, 4, 6, 7

[10] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[11] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE International Conference on Computer Vision*, 2021. 1, 7

[12] Hongyu Zhou, Zheng Ge, Zeming Li, and Xiangyu Zhang. Matrixvt: Efficient multi-camera to bev transformation for 3d perception. *arXiv preprint arXiv:2211.10593*, 2022. 2, 3

[13] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. In *British Machine Vision Conference*, 2019. 2, 3

[14] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020. 2

[15] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. BEVDepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 2, 3, 4, 6, 7

[16] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. 2, 3, 7

[17] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[18] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M^{2}BEV: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 3

[19] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 3, 5

[20] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1280–1289, 2022. 3

[21] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022. 3

[22] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformers. *arXiv preprint arXiv:2206.15398*, 2022. 3, 7

[23] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv preprint arXiv:2206.10965*, 2022. 3

[24] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022. 3

[25] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. UniFormer: Unified multi-view fusion transformer

for spatial-temporal representation in bird's-eye-view. *arXiv preprint arXiv:2207.08536*, 2022. 3

[26] Zhipeng Luo, Changqing Zhou, Gongjie Zhang, and Shijian Lu. Detr4d: Direct multi-view 3d object detection with sparse attention. *arXiv preprint arXiv:2212.07849*, 2022. 3

[27] 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 preprint arXiv:2203.11089*, 2022. 3

[28] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022. 3, 6, 7

[29] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023. 3

[30] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model. *arXiv preprint arXiv:2305.14018*, 2023. 3

[31] Jiahao Wang, Guo Chen, Yifei Huang, Limin Wang, and Tong Lu. Memory-and-anticipation transformer for online action understanding, 2023. 3

[32] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 3, 7

[33] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETRv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 3, 7

[34] Hongxiang Jiang, Wenming Meng, Hongmei Zhu, Qian Zhang, and Jihao Yin. Multi-camera calibration free bev representation for 3d object detection. *arXiv preprint arXiv:2210.17252*, 2022. 3

[35] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5935–5943, 2023. 3

[36] Junjie Huang and Guan Huang. BEVDet4D: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 4, 6, 7

[37] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International conference on 3D vision (3DV)*, 2016. 4

[38] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[40] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[42] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. Sts: Surround-view temporal stereo for multi-view 3d detection. *arXiv preprint arXiv:2208.10145*, 2022. 7

[43] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 7

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision*, 2021. 7

[45] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 7

[46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[47] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *IEEE International Conference on Computer Vision*, pages 913–922, 2021. 7

[48] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 7