

# RecurrentBEV: A Long-term Temporal Fusion Framework for Multi-view 3D Detection

Ming Chang<sup>1</sup>, Xishan Zhang<sup>1,2\*</sup>, Rui Zhang<sup>2</sup>, Zhipeng Zhao<sup>1</sup>, Guanhua He<sup>1,2,3</sup>, and Shaoli Liu<sup>1</sup>

<sup>1</sup> Cambricon Technologies {changming,zhaozhipeng,liushaoli}@cambricon.com

<sup>2</sup> SKL of Processors, Institute of Computing Technology, CAS, Beijing, China  
{zhangxishan,zhangrui}@ict.ac.cn, heguanhua22@mailsucas.ac.cn

<sup>3</sup> University of Chinese Academy of Sciences, UCAS, Beijing, China

**Abstract.** Long-term temporal fusion is frequently employed in camera-based Bird’s-Eye-View (BEV) 3D object detection to improve detection of occluded objects. Existing methods can be divided into two categories, parallel fusion and recurrent fusion. Recurrent fusion reduces inference latency and memory consumption but fails to exploit the long-term information as well as parallel fusion. In this paper, we first find two shortcomings of recurrent fusion paradigm: (1) Gradients of previous BEV features cannot directly contribute to the fusion module. (2) Semantic ambiguity are caused by coarse granularity of the BEV grids during aligning BEV features. Then based on the above analysis, we propose RecurrentBEV, a novel recurrent temporal fusion method for BEV based 3D object detector. By adopting RNN-style back-propagation and new-designed inner grid transformation, RecurrentBEV improves the long-term fusion ability while still enjoying efficient inference latency and memory consumption during inference. Extensive experiments on the nuScenes benchmark demonstrate its effectiveness, achieving a new state-of-the-art performance of 57.4% mAP and 65.1% NDS on the test set. The real-time version (25.6 FPS) achieves 44.5% mAP and 54.9% NDS without external dataset, outperforming the previous best method StreamPETR by 1.3% mAP and 0.9% NDS. The code is available at <https://github.com/lucifer443/RecurrentBEV>.

**Keywords:** Autonomous Driving · Bird’s-Eye-View · 3D Object Detection · Temporal Fusion

## 1 Introduction

Camera-only 3D detection presents a formidable challenge for autonomous driving and has garnered significant attention in recent years owing to its low cost of deployment. The Bird’s-Eye-View (BEV) representation-based approach has recently demonstrated notable improvement by leveraging temporal information [8, 11, 21, 22, 33].

---

\* Corresponding Author.

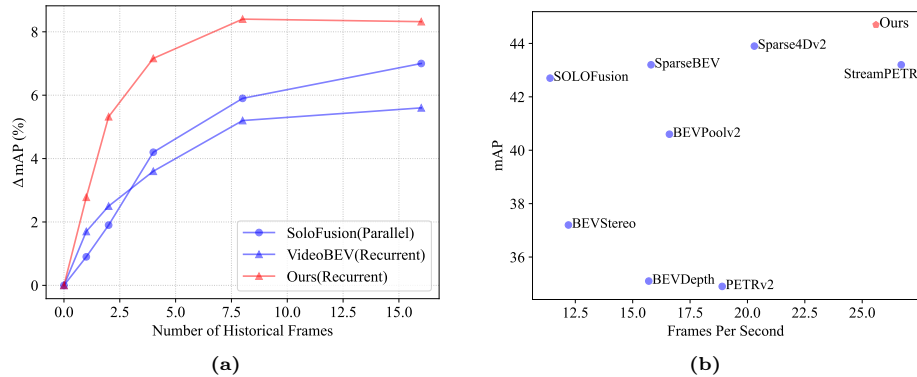


**Fig. 1:** (a) Comparison of improvement ( $\Delta mAP\uparrow$ ) from long-term fusion between parallel paradigm (SOLOFusion [33]) and recurrent paradigm (VideoBEV [8], Ours). (b) Performance comparison on the validation set of nuScenes [2]. All methods use ResNet50 [9] pre-trained on ImageNet [6] as backbone with  $256 \times 704$  as input size. Frames per second (FPS) is measured on a single RTX 3090 with the PyTorch fp32.

The temporal fusion mechanisms utilised in the BEV-based 3D detector can be broadly categorised into parallel fusion [11, 33, 39], and recurrent fusion [8, 22]. Parallel fusion [11, 33, 39] caches and updates multiple historical features in learning the visual features for current frames. Therefore, the overhead of computing and storage grows as the length of the historical frame feature increases. Since the superior performance requires longer historical information, this fusion paradigm faces a dilemma between effectiveness and efficiency in deployment. In contrast, recurrent fusion [22] encodes all historical features into a single recurrent feature to cache and update, so it largely reduces inference latency and memory consumption. For instance, when employing 16 historical frames, the recurrent fusion method VideoBEV [8] exhibits  $6\times$  times faster in latency and uses  $12\times$  less memory compared to the state-of-the-art parallel fusion method SOLOFusion [33]. However, we observe that recurrent fusion fails to exploit the long-term information as well as parallel fusion. As depicted in Fig. 1a, VideoBEV benefits less ( $\Delta mAP\uparrow$ ) from utilizing historical frames than that of SOLOFusion ( $\Delta mAP\uparrow$ ). This phenomenon becomes even more pronounced, when utilizing more than four historical frames.

Why is the performance of recurrent fusion not as good as parallel fusion? We conducted a detailed analysis of the differences between the two paradigms and found two reasons: (1) Gradient propagation. During the training process, only the recent frames contribute to gradients of fusion module in the recurrent paradigm, while all historical frames contribute equally in the parallel paradigm. This prevents the temporal fusion module of recurrent paradigm from learning fusion pattern from long-term temporal information of all historical frames. (2) Semantic ambiguity. Due to the coarse granularity of the BEV grids, the recurrent paradigm generates errors when aligning historical features multiple

times, resulting in semantic ambiguity for the same object. As a result, recurrent paradigm may fuse different objects on the singular BEV grid, confusing the recognition of the same object. For the above two reasons, the previous recurrent method cannot fully and correctly utilize the long-term temporal information.

Motivated by the above analysis, in this paper, we propose RecurrentBEV, a novel temporal fusion framework for BEV-based 3D object detector to improve the long-term fusion ability of recurrent paradigm while still has efficient latency and memory consumption in inference. First, RecurrentBEV adopts an RNN-style back-propagation during training, enabling historical frames to contribute gradients to the fusion module directly. This improvement solves the first problem while still maintaining inference efficiency. Second, we design an inner grid transformation before fusion, which can learn fine-grained semantic changes from ego-motion matrix. This effectively reduces semantic differences in the inter-frame alignment process.

Based on the RNN-style back-propagation and the inner grid transformation, the proposed RecurrentBEV effectively improves long-term fusion in the recurrent paradigm, as shown in Fig.1a. Extensive experiments on the nuScenes 3D detection dataset evaluate the effectiveness of the proposed RecurrentBEV. As shown in Fig.1b, it outperforms both existing BEV-based algorithms such as SOLOFusion, VideoBEV and query-based algorithms such as StreamPETR [35], SparseBEV [27], Sparse4Dv2 [26] in both accuracy and efficiency. To summarize, our contributions are:

- We provide an in-depth analysis for two temporal fusion paradigms and point out two factors that render the recurrent fusion less effective: issues with gradient propagation and semantic ambiguity.
- We propose RecurrentBEV, a novel temporal fusion framework for BEV-based 3D object detector. It adopts RNN-style back-propagation and inner grid transformation, attaining state-of-the-art results (57.4% mAP and 65.1% NDS) on the nuScenes benchmark. It represents the first recurrent fusion method with explicit BEV features that surpasses parallel fusion performance.

## 2 Related Works

### 2.1 Multi-view 3D Object Detection

Multi-view 3D object detection is a fundamental task for autonomous driving. Previous works mainly focus on how to transform the 2D features into a unified 3D space. This transformation can be divided to two categories, BEV-based methods [10, 13, 18, 22, 23, 41, 42] and query-based methods [25, 28, 36, 37], depending on whether an explicit 3D representation is employed.

BEVDet [13] is one typical method of BEV-based multi-view 3D object detection. This method involves converting 2D features to BEV features based on the Lift-Splat-Shoot [34] paradigm. BEVDepth [21] is a further example, which notably refines the transformation process via explicit depth supervision.

BEVFormer [22] further introduces a novel attention mechanism to gather 2D features for BEV grids.

Query-based methods typically define several object queries in 3D space to gather features from multi-view image features. DETR3D [36] projects object queries to image coordinates to sample features then predicts 3D bounding boxes directly. DETR4D [31] and Sparse4D [25] enhance it by leveraging historical information. PETR [28] encodes the position information of 3D coordinates into image feature, facilitating the detection of 3D objects similar to DETR [3].

## 2.2 Temporal Fusion for BEV-based Detectors

Recent works find that incorporating temporal information can enhance the stability in perception results and improve overall performance. BEVFormer [22] first proposes the temporal self-attention mechanism to fuse the historical BEV features in a recurrent paradigm. BEVDet4D [11] extends BEVDet [13] to a temporal version by employing a concatenation operation to integrate previous BEV features, usually called parallel paradigm. SOLOFusion [33] extends these methods by increasing the number of previous frames to 16, resulting in significant benefits. BEVFormerv2 [39] switches to parallel paradigm same as SOLOFusion. However, the inference latency and memory consumption of the parallel paradigm increase linearly as the historical frame lengthens. VideoBEV [8] applies recurrent fusion to the LSS-based 3D detector, resulting in significant reduction of inference latency and memory consumption. However, the improvement of long-term fusion in VideoBEV is limited in comparison to SOLOFusion. In order to combine the advantages of the two paradigms, we construct a novel temporal fusion framework based on efficient recurrent paradigm.

## 3 Method

Converting multi-camera image features to bird’s-eye-view (BEV) features can provide a unified surrounding environment representation for autonomous driving perception tasks. It involves transforming image data into a top-down representation of the environment, which often denoted as BEV features. The BEV detectors contain an image encoder, a 2D-3D view transformation, and a temporal fusion module. In this work, we present a new temporal fusion module, which can effectively aggregate spatio-temporal BEV features.

### 3.1 Analysis of BEV-based Temporal Fusion Paradigms

There are two kinds of temporal fusion paradigms in BEV detection, which are the effective parallel fusion and efficient recurrent fusion. The parallel fusion caches and updates multiple historical BEV features in learning the visual features for current frames. The recurrent fusion encodes all historical features into a single recurrent feature to cache and update. We conduct a detailed analysis to identify the disparities among them and pinpoint two crucial factors that cause

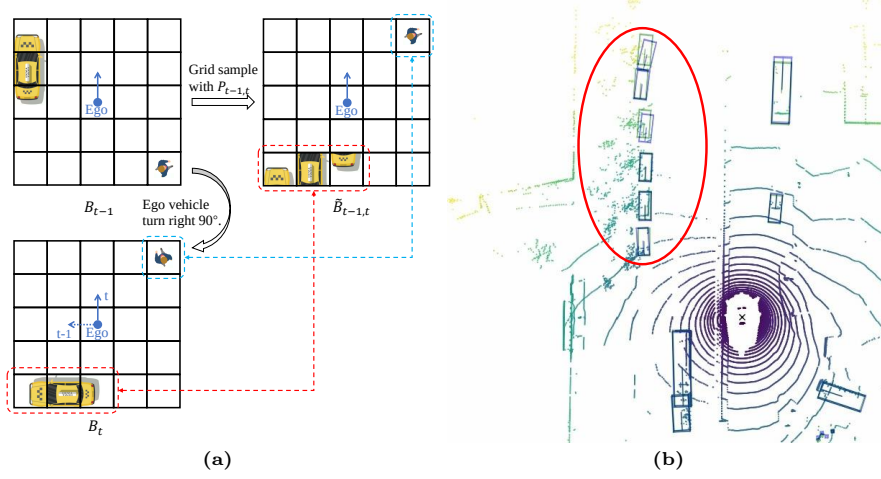


**Fig. 2:** (a) Semantic ambiguity caused by grid sample operation. When ego vehicle turning right, BEV feature change from  $B_{t-1}$  to  $B_t$ . In fusion, the grid sample operation used for aligning  $B_{t-1}$  and  $B_t$  only aligns the position but ignores the orientation as shown in the blue and red dash lines. (b) Visualization of the semantic ambiguity. The green boxes represent the BEV feature change from  $B_{t-1}$ , which have been transformed to align with the  $B_t$  and decoded into bounding boxes. The blue boxes depict the actual bounding boxes in the  $B_t$ . Notably, for static objects, there is a mismatch in the orientation of the green and blue boxes.

the sub-optimal performance of recurrent fusion: issues with gradient propagation and semantic ambiguity.

**Gradient propagation** Aligning historical frames to current frame is essential for BEV-based temporal fusion. Given the previous BEV feature  $B_j$ , this process can be written as:

$$\tilde{B}_{j,i} = \zeta(B_j, P_{j,i}) \quad (1)$$

where  $\tilde{B}_{j,i}$  is the  $j$ -th frame BEV feature in the ego coordinate of  $i$ -th frame,  $P_{j,i}$  is the ego-motion matrix between the  $j$ -th frame and  $i$ -th frame and  $\zeta$  refers to the ego-motion transformation function, specifically grid sample operation [15]. Based on Eq.(1), We can express the parallel fusion of the  $i$ -th frame as:

$$\bar{B}_i = \varphi(\tilde{B}_{i-k,i}, \dots, \tilde{B}_{i-1,i}, B_i, W) \quad (2)$$

where  $\bar{B}_i$  is the fused BEV feature of  $i$ -th frame,  $\varphi$  is the temporal fusion function (convolution [11, 21, 33, 39] or deformable attention [22]) and  $W$  refers the parameters of it. Similarly, the recurrent paradigm for the  $i$ -th frame can be written as:

$$\bar{B}_i = \varphi(\zeta(\bar{B}_{i-1}, P_{i-1,i}), B_i, W) \quad (3)$$

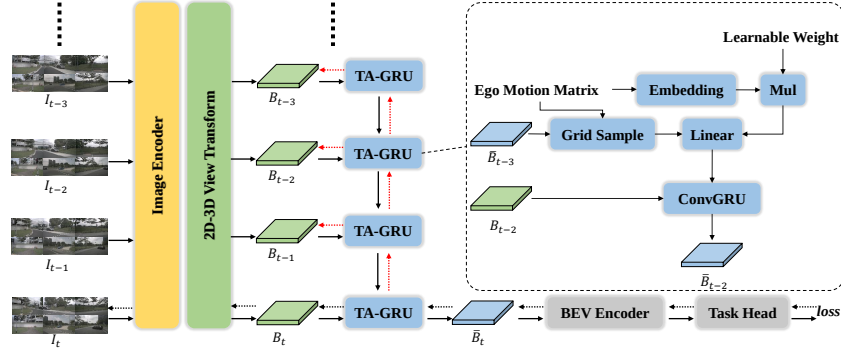


**Fig. 3:** Overall architecture of the proposed RecurrentBEV. It shows a training process in the  $t$ -th frame, while arrows indicate forward propagation and backward propagation. The red arrows indicate the additional back-propagation introduced by our method. The details of the Transformation-aware Gate Recurrent Unit (TA-GRU) is illustrated in dotted box.

Based on Eq.(3), we can calculate the gradient for the parameters in the temporal fusion function for parallel fusion.

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \bar{B}_i} \cdot \varphi'(\bar{B}_{i-k,i}, \dots, \bar{B}_{i-1,i}, B_i) \quad (4)$$

In recurrent fusion,  $B_{i-1}$  is detached so that the gradient of  $W$  in recurrent fusion can be written as:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \bar{B}_i} \cdot \varphi'(\zeta(\bar{B}_{i-1}, P_{i-1,i}), B_i) \quad (5)$$

Comparing Eq.(4) and Eq.(5), it is evident that only the current BEV feature  $B_i$  and one previous BEV feature  $\bar{B}_{i-1}$  affect the gradient of  $W$  directly in recurrent paradigm while all previous BEV features within a window size  $k$  contribute to gradient of  $W$  in parallel paradigm. This is one reason that recurrent paradigm make limited progress in long-term fusion.

**Semantic Ambiguity** To optimize memory usage, the recurrent fusion approach only caches a single historical 3D BEV feature instead of storing a sequence of 2D features. Consequently, this approach encounters challenges with temporal alignment in the 3D space.

In the current recurrent fusion method, we have observed semantic ambiguity issues arising during the temporal alignment process in the 3D space. As illustrated in Fig.2a, the coarse granularity of BEV grids results in inaccurate transformation of orientation information within each grid during the temporal alignment process represented by Eq.(1). Consequently, this process introduces incorrect object orientations in real-world detection scenarios, as evident in Fig.2b.

In the parallel fusion counterpart, temporal alignment in the 3D space is avoided by directly projecting a sequences of 2D view feature onto the current 3D BEV feature space [33, 39]. Assuming that the past perspective 2D view feature at time step  $j$  is denoted as  $C_j$ , the formulation for projecting it onto the BEV feature at time step  $i$  is as follows:

$$\tilde{B}_{j,i} = ViewTrans(C_j, I_j, P_{j,i} \cdot E_j) \quad (6)$$

where  $I_j$  and  $E_j$  denote camera intrinsic and extrinsic respectively. This additional view transformation in 2D space effectively aligns objects between frames, subsequently enhancing the performance of parallel fusion. Nevertheless, this approach is not feasible for recurrent fusion due to the unavailability of 2D features during the recurrent fusion process. This discrepancy serves as another contributing factor to the performance gap between the two fusion methods.

### 3.2 RecurrentBEV

Based on the two above-mentioned possible reasons, we design a novel temporal fusion framework called RecurrentBEV. Similar to existing BEV-based 3D object detectors [8, 11, 21, 33], RecurrentBEV contains an image encoder, a 2D-3D view transformation, a temporal fusion module, a BEV encoder and a task head, as illustrated in Fig.3. The main difference between RecurrentBEV and previous recurrent fusion is the temporal fusion module which can learn from all historical frames directly and learn fine-grained ego-motion transform.

**RNN-style Back-propagation** In prior recurrent fusion methods [22], the  $\bar{B}_{i-1}$  in Eq.(3) is solely computed for forward propagation, but not for backward propagation. Thus, historical BEV features do not contribute to the gradient of temporal fusion module in Eq.(5), preventing optimizing the temporal fusion module with all the relevant historical BEV features. To address this issue, RecurrentBEV adopts RNN-style Back-propagation that all the relevant historical features back-propagate gradients to adjacent layers. In this context, the gradients of parameters in temporal fusion function is written as:

$$\frac{\partial L}{\partial W} = \sum_{j=i-k}^i \frac{\partial L}{\partial \bar{B}_j} \cdot \varphi'(\zeta(\bar{B}_{j-1}, P_{j-1,j}), B_j) \quad (7)$$

where  $k$  is the number of total historical frames using in a iteration of training process. As a result, each historical BEV feature makes an equal contribution to the gradient of the temporal fusion module, akin to the parallel paradigm.

Computing the back-propagation along the temporal dimension for the entire network places significant demands on GPU memory. As shown in Table 1 (Setting 1), if all modules, including image encoders, undergo back-propagation along the temporal dimension, memory overflow issues arise even when the frame number reaches only 2. However, it is imperative for the training process to incorporate long-term frames to effectively learn temporal information (refer to

| Training frames | 0      | 1      | 2      | 4      | 8      | 16     |
|-----------------|--------|--------|--------|--------|--------|--------|
| Setting 1       | 15,067 | 30,105 | OOM    | OOM    | OOM    | OOM    |
| Setting 2       | 15,067 | 17,661 | 18,735 | 19,975 | 23,579 | 30,941 |

**Table 1:** The memory usage (MB) during training. OOM means out of memory on A100-40G with FP32 precision.

Sec.4.6 for details). Therefore, we need to strike a balance between GPU memory usage (determined by the modules involved in back-propagation) and the length of the time sequence. In our design, as depicted by the red arrows in Figure 3, RecurrentBEV further computes all the gradients of the temporal fusion module (TA-GRU in the following) via back-propagation along the temporal dimension. Consequently, RecurrentBEV (Setting 2 in Table 1) can be trained with a batch size of 8 with FP32 precision on a single A100-40G GPU, effectively utilizing up to 16 consecutive frames.

**Transformation-aware Gated Recurrent Unit** Because of the coarse granularity of the BEV grids, the semantic feature in a grid can not project correctly when model transforms the previous BEV features to the coordinate system of the current ego. We design an inner grid transformation to solve this problem. For simplicity, we use a linear layer to model this transformation:

$$\tilde{B}_{j,i} = \text{Linear}(\zeta(B_j, P_{j,i}), W_p) \quad (8)$$

where  $W_p$  is the parameter of *Linear*. For simplicity, we assume different channels in a projected feature share same motion embedding, so that the  $W_p$  can be decoupled as:

$$W_p = E_{j,i} \odot \tilde{W}_p \quad (9)$$

where  $E_{j,i} \in \mathbb{R}^{C_{in} \times 1}$  is ego motion embedding from timestamp  $j$  to timestamp  $i$ ,  $\tilde{W}_p \in \mathbb{R}^{C_{in} \times C_{out}}$  is a learnable weight and  $\odot$  denotes a Hadamard Product. Motion embedding  $E_{j,i}$  is a high dimensional vector converted from the given ego motion matrix  $P_{j,i}$ :

$$E_{j,i} = \sigma(\psi(P_{j,i})) \quad (10)$$

where  $\sigma$  is a sigmoid activation and  $\psi$  is a multi-layer perception (MLP). Combining Eq.(8), Eq.(9) and Eq.(10), the complete inner grid transformation can be written as:

$$\tilde{B}_{j,i} = \text{Linear}(\zeta(\bar{B}_j, P_{j,i}), \sigma(\psi(P_{j,i})) \odot \tilde{W}_p) \quad (11)$$

For the RNN modules, we adopted the typical Convolutional Gated Recurrent Unit (ConvGRU) [1]. By combining the aforementioned inner grid transformation, we introduce a novel temporal fusion module named the Transformation-Aware Gated Recurrent Unit (TA-GRU). The entire process of temporal fusion is formulated as follows:

$$W_p = \sigma(\psi(P_{i-1,i})) \odot \widetilde{W}_p \quad (12)$$

$$\widetilde{B}_{i-1,i} = \text{Linear}(\zeta(\overline{B}_{i-1}, P_{i-1,i}), W_p) \quad (13)$$

$$Z_i = \sigma(\text{Conv}([\widetilde{B}_{i-1,i}, B_i], W_z)) \quad (14)$$

$$R_i = \sigma(\text{Conv}([\widetilde{B}_{i-1,i}, B_i], W_r)) \quad (15)$$

$$O_i = \tanh(\text{Conv}([R_i \odot \widetilde{B}_{i-1,i}, B_i], W_o)) \quad (16)$$

$$\overline{B}_i = (1 - Z_i) \odot \widetilde{B}_{i-1,i} + Z_i \odot O_i \quad (17)$$

where  $[\cdot, \cdot]$  represents the concatenation of feature maps and  $W_z, W_r, W_o$  are 2D-convolution kernels. Because of inner grid transformation, TA-GRU can learn fine-grained semantic changes from ego-motion matrix, thus removing wrong orientation.

**Inference Efficiency** Comparing Eq.(2) and Eq.(3), parallel fusion requires  $k$  historical frames to be cached while recurrent fusion requires only one. Additionally, the transformation process described in Eq.(1) or Eq.(6) also needs to be computed  $k$  times in parallel fusion, whereas in recurrent fusion it only needs to be computed once. As a result, recurrent fusion, which has lower inference latency and smaller memory consumption, is more suitable for deployment in the industry.

Similar to previous recurrent fusion methods [8, 22], RecurrentBEV adopts sequential inference which only needs to cache a single historical BEV feature  $\overline{B}$ , initialized with zero and updated at the end of the fusion.

## 4 Experiments

### 4.1 Dataset and Metrics

We evaluate our method on the challenging nuScenes dataset [2]. The dataset comprises 1000 driving scenes, each lasting about 20 seconds, collected in Boston and Singapore, which are divided into 700, 150 and 150 scenes respectively for training, validation, and testing. The vehicle is equipped with 6 cameras for a complete 360-degree field of view. The nuScenes dataset annotates 23 object classes with accurate 3D bounding boxes for each scene at 2Hz. For evaluation purposes, 10 classes are selected, namely: car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, and traffic cone. As for evaluation, we use the following metrics, mean average precision (mAP), the nuScenes detection score (NDS), and five true positive metrics <sup>4</sup> including average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), average attribute error (AAE).

<sup>4</sup> details of evaluation metrics: <https://www.nuscenes.org/object-detection>

| Methods             | Backbone  | Image Size        | ED           | mAP $\uparrow$ | NDS $\uparrow$ | mATE $\downarrow$ | mASE $\downarrow$ | mAOE $\downarrow$ | mAVE $\downarrow$ | mAAE $\downarrow$ |
|---------------------|-----------|-------------------|--------------|----------------|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| BEVDet [13]         | ResNet50  | 256 $\times$ 704  | $\times$     | 0.298          | 0.379          | 0.725             | 0.279             | 0.589             | 0.860             | 0.245             |
| BEVDet4D [11]       | ResNet50  | 256 $\times$ 704  | $\times$     | 0.322          | 0.457          | 0.703             | 0.278             | 0.495             | 0.354             | 0.206             |
| BEVDepth [21]       | ResNet50  | 256 $\times$ 704  | $\times$     | 0.351          | 0.475          | 0.639             | <b>0.267</b>      | 0.479             | 0.428             | 0.198             |
| BEVStereo [20]      | ResNet50  | 256 $\times$ 704  | $\times$     | 0.372          | 0.500          | 0.598             | 0.270             | 0.438             | 0.367             | 0.190             |
| BEVPoolv2 [12]      | ResNet50  | 256 $\times$ 704  | $\times$     | 0.406          | 0.526          | 0.572             | 0.275             | 0.463             | 0.275             | 0.188             |
| SOLOFusion [33]     | ResNet50  | 256 $\times$ 704  | $\times$     | 0.427          | 0.534          | 0.567             | 0.274             | 0.511             | 0.252             | 0.181             |
| VideoBEV [8]        | ResNet50  | 256 $\times$ 704  | $\times$     | 0.422          | 0.535          | 0.564             | 0.276             | 0.440             | 0.286             | 0.198             |
| PETrv2 [29]         | ResNet50  | 256 $\times$ 704  | $\times$     | 0.349          | 0.456          | 0.700             | 0.275             | 0.580             | 0.437             | 0.187             |
| Sparse4Dv2 [26]     | ResNet50  | 256 $\times$ 704  | $\times$     | 0.439          | 0.539          | 0.598             | 0.270             | 0.475             | 0.282             | <b>0.179</b>      |
| SparseBEV [27]      | ResNet50  | 256 $\times$ 704  | $\times$     | 0.432          | 0.545          | 0.606             | 0.274             | <b>0.387</b>      | <b>0.251</b>      | 0.186             |
| StreamPETR [35]     | ResNet50  | 256 $\times$ 704  | $\times$     | 0.432          | 0.540          | 0.581             | 0.272             | 0.413             | 0.295             | 0.195             |
| <b>RecurrentBEV</b> | ResNet50  | 256 $\times$ 704  | $\times$     | <b>0.445</b>   | <b>0.549</b>   | <b>0.555</b>      | 0.272             | 0.451             | 0.256             | 0.204             |
| BEVDepth [13]       | ResNet101 | 512 $\times$ 1408 | $\times$     | 0.412          | 0.535          | 0.565             | 0.266             | 0.358             | 0.331             | 0.190             |
| SOLOFusion [33]     | ResNet101 | 512 $\times$ 1408 | $\times$     | 0.483          | 0.582          | 0.503             | <b>0.264</b>      | 0.381             | <b>0.246</b>      | 0.207             |
| Sparse4Dv2 [26]     | ResNet101 | 512 $\times$ 1408 | $\times$     | 0.485          | 0.580          | 0.555             | 0.272             | 0.367             | 0.256             | <b>0.182</b>      |
| <b>RecurrentBEV</b> | ResNet101 | 512 $\times$ 1408 | $\times$     | <b>0.509</b>   | <b>0.599</b>   | <b>0.487</b>      | 0.267             | <b>0.358</b>      | 0.247             | 0.199             |
| SparseBEV [27]      | ResNet101 | 512 $\times$ 1408 | $\checkmark$ | 0.501          | 0.592          | 0.562             | 0.265             | 0.321             | 0.243             | 0.195             |
| StreamPETR [35]     | ResNet101 | 512 $\times$ 1408 | $\checkmark$ | 0.504          | 0.592          | 0.569             | 0.262             | <b>0.315</b>      | 0.257             | 0.199             |
| Sparse4Dv2 [26]     | ResNet101 | 512 $\times$ 1408 | $\checkmark$ | 0.505          | 0.594          | 0.548             | 0.268             | 0.348             | 0.239             | <b>0.184</b>      |
| <b>RecurrentBEV</b> | ResNet101 | 512 $\times$ 1408 | $\checkmark$ | <b>0.528</b>   | <b>0.612</b>   | <b>0.481</b>      | <b>0.256</b>      | 0.355             | <b>0.226</b>      | 0.201             |

**Table 2:** Comparison results on 3D detection on the nuScenes validation set. ED is external data which indicates to use pre-trained weights from the external dataset i.e. nuImages.

## 4.2 Implementation Details

We conduct experiments based on the BEVDet4D-Depth [11, 21] with the same learning rate, optimizer, and data augmentation. Drawing upon previous methods [21, 33, 35], we utilize ResNet50 [9], ResNet101, ConNeXt-B [30] and V2-99 [17] as backbones. On the nuScenes validation set, we report results of ResNet50 and ResNet101 models with pre-trained weights ImageNet [6] and nuImages [35]. On the nuScenes test set, we provide the performance of V2-99 model initialized from DD3D [32] and ConvNeXt-B model benefited from COCO [24] pre-training. The models in the ablation study are trained for 24 epochs without CBGS [43]. When compared with others, we follow the training setting of StreamPETR [35], extending the training epochs to 60 and using 2D auxiliary supervision for a fair comparison.

To achieve real-time inference speed, we optimize the BEVDet4D-Depth baseline by making several key modifications. Firstly, we reduce the number of depth bins. Secondly, we eliminate the ASPP [5] module from the DepthNet to minimize the latency associated with view transformation. Additionally, we merge 6 detection tasks performed by CenterHead [40] into a single task, significantly reducing post-processing time. These optimizations combined enable us to achieve a more efficient inference speed.

| Methods             | Backbone   | mAP $\uparrow$ | NDS $\uparrow$ | mATE $\downarrow$ | mASE $\downarrow$ | mAOE $\downarrow$ | mAVE $\downarrow$ | mAAE $\downarrow$ | Conference |
|---------------------|------------|----------------|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------|
| DETR3D* [36]        | V2-99      | 0.412          | 0.479          | 0.641             | 0.255             | 0.394             | 0.845             | 0.133             | CoRL2021   |
| PETR [28]           | V2-99      | 0.441          | 0.504          | 0.593             | 0.249             | 0.383             | 0.808             | 0.132             | ECCV2022   |
| UVTR [19]           | V2-99      | 0.472          | 0.551          | 0.577             | 0.253             | 0.391             | 0.508             | 0.123             | NIPS2022   |
| BEVFormer [22]      | V2-99      | 0.481          | 0.569          | 0.582             | 0.256             | 0.375             | 0.378             | 0.126             | ECCV2022   |
| PETrv2 [29]         | V2-99      | 0.490          | 0.582          | 0.561             | 0.243             | 0.361             | 0.343             | 0.120             | ICCV2023   |
| PolarFormer [16]    | V2-99      | 0.493          | 0.572          | 0.556             | 0.256             | 0.364             | 0.439             | 0.127             | AAAI2023   |
| BEVDepth [21]       | V2-99      | 0.503          | 0.600          | 0.445             | 0.245             | 0.378             | 0.320             | 0.126             | AAAI2023   |
| VEDet [4]           | V2-99      | 0.505          | 0.585          | 0.545             | 0.244             | 0.346             | 0.421             | 0.123             | CVPR2023   |
| MV2D [37]           | V2-99      | 0.511          | 0.596          | 0.525             | 0.243             | 0.357             | 0.357             | 0.120             | ICCV2023   |
| CAPE [38]           | V2-99      | 0.525          | 0.610          | 0.503             | 0.242             | 0.361             | 0.306             | <b>0.114</b>      | CVPR2023   |
| BEVStereo [20]      | V2-99      | 0.525          | 0.610          | 0.431             | 0.246             | 0.358             | 0.357             | 0.138             | AAAI2023   |
| SA-BEV [41]         | V2-99      | 0.533          | 0.624          | 0.430             | 0.241             | 0.338             | 0.282             | 0.139             | ICCV2023   |
| FB-BEV [23]         | V2-99      | 0.537          | 0.624          | 0.439             | 0.250             | 0.358             | 0.270             | 0.128             | ICCV2023   |
| StreamPETR [35]     | V2-99      | 0.550          | 0.636          | 0.479             | <b>0.239</b>      | <b>0.317</b>      | <b>0.241</b>      | 0.119             | ICCV2023   |
| SparseBEV [27]      | V2-99      | 0.556          | 0.636          | 0.485             | 0.244             | 0.332             | 0.246             | 0.117             | ICCV2023   |
| Sparse4Dv2 [26]     | V2-99      | 0.557          | 0.638          | 0.462             | 0.238             | 0.328             | 0.264             | 0.115             | -          |
| <b>RecurrentBEV</b> | V2-99      | <b>0.573</b>   | <b>0.651</b>   | <b>0.410</b>      | 0.247             | 0.323             | 0.249             | 0.123             | -          |
| BEVDepth [21]       | ConvNeXt-B | 0.520          | 0.609          | 0.445             | 0.243             | 0.352             | 0.347             | 0.127             | AAAI2023   |
| AeDet* [7]          | ConvNeXt-B | 0.531          | 0.620          | 0.439             | 0.247             | 0.344             | 0.292             | 0.130             | CVPR2023   |
| SOLOFusion [33]     | ConvNeXt-B | 0.540          | 0.619          | 0.453             | 0.257             | 0.376             | 0.276             | 0.148             | ICLR2023   |
| VCD-A [14]          | ConvNeXt-B | 0.548          | 0.631          | 0.436             | 0.244             | 0.343             | 0.290             | 0.120             | NIPS2023   |
| StreamPETR [35]     | ConvNeXt-B | 0.550          | 0.631          | 0.493             | <b>0.241</b>      | 0.343             | <b>0.243</b>      | 0.123             | ICCV2023   |
| VideoBEV [8]        | ConvNeXt-B | 0.554          | 0.629          | 0.457             | 0.249             | 0.381             | 0.266             | 0.132             | -          |
| <b>RecurrentBEV</b> | ConvNeXt-B | <b>0.574</b>   | <b>0.651</b>   | <b>0.411</b>      | 0.248             | <b>0.333</b>      | 0.257             | <b>0.116</b>      | -          |

**Table 3:** Comparison results on 3D detection on the nuScenes test set. \* indicates using test-time augmentation in test process. No future frames are used in any of the methods.

### 4.3 Main Results

We compare the proposed RecurrentBEV with previous state-of-the-art multi-view 3D detectors on the nuScenes validation and test set.

On the nuScenes validation set, as shown in Fig.1b and Tab.2, the real-time version of RecurrentBEV (25.6 FPS), using ResNet50 as backbone and  $256 \times 704$  as the input size, outperforms the previous state-of-the-art method StreamPETR by 1.3% mAP and 0.9% NDS on the nuScenes validation set. To scale up our method, we adopt ResNet101 backbone with nuImages pre-training and extend input size to  $512 \times 1408$ , outperforming the same setting StreamPETR by 2.4% mAP and 2.0% NDS. It’s worth noting that RecurrentBEV makes remarkable progress on the mAP and mATE metrics, indicating the improvement of detection of occlusion objects. We’ll delve into the reasons behind this in Sec.4.6.

On the nuScenes test set in Tab.3, we compare with all the recent methods in multi-view 3D object detection. Following the settings in StreamPETR, we adopt V2-99 and ConvNeXt-B as backbone and  $640 \times 1600$  as the input size. Using V2-99 as backbone, RecurrentBEV can surpass the previous state-of-the-art method Sparse4Dv2 by 1.6% mAP and 1.3% NDS. When the backbone is scaled up to ConvNeXt-B, RecurrentBEV demonstrates comparable performance, surpassing other methods.

| method                    | mAP $\uparrow$ | NDS $\uparrow$ | memory usage $\downarrow$ | latency $\uparrow$ |
|---------------------------|----------------|----------------|---------------------------|--------------------|
| parallel fusion [33]      | 0.427          | 0.534          | 945MB                     | 96ms               |
| recurrent fusion baseline | 0.396          | 0.515          | 516MB                     | 34ms               |
| RecurrentBEV              | 0.445          | 0.549          | 549MB                     | 39ms               |

**Table 4:** Comparison of time and memory in inference. All data are measured on a single RTX 3090 with PyTorch fp32. Implementation details are the same as Tab.2 ResNet50 Backbone. For more latency analysis, please refer to supplementary.

#### 4.4 Time and Memory Comparison

To demonstrate the time and memory efficiency in inference, we compare the proposed RecurrentBEV with parallel fusion methods and previous recurrent fusion baselines. As shown in Tab.4 RecurrentBEV is 2.5 times faster than the normal parallel fusion method SOLOFusion [33], and the memory usage is reduced by 42%. When compared to the normal recurrent baseline <sup>5</sup>, RecurrentBEV requires only an additional 33MB of memory while significantly improving the mean Average Precision (mAP) from 0.396 to 0.445.

#### 4.5 Ablation Study

In the ablation study, we delve into three aspects: the components of RecurrentBEV, the receptive field of TA-GRU, and the length of frame sequence in training. We conduct the ablation experiments with ResNet-50 backbone.

**Ablation of Each Component.** We conduct a component-wise ablation in Tab.5. Exp.1 starts with a ConvGRU as temporal fusion module, achieving similar performance to VideoBEV [8]. In Exp.2 and Exp.3, we enable the backward propagation of historical BEV features in training process. In Exp.3 and Exp.4, we replace the temporal fusion module with TA-GRU. The overall components (BP and TA-GRU) brings significant performance gains (+1.7% mAP, +1.6% NDS, -4.7% mAOE) over baseline.

Specifically, the TA-GRU demonstrates its ability to enhance the mAP and mAOE metrics, regardless of whether backward propagation of historical BEV features is enabled. This proves the TA-GRU’s consistent ability to alleviate semantic ambiguity for identical objects. Notably, the mean average orientation error (mAOE) has a notable increase, indicating that the inner grid transformation, facilitated by the alignment of historical BEV features, effectively mitigates semantic ambiguity by refining the orientation of objects.

As demonstrated in Exp.4, when the historical BEV feature is detached, the performance suffers, highlighting the importance of RNN-style back-propagation. However, relying solely on back-propagation of historical BEV features during training results in limited improvement. This shows the necessity of employing

<sup>5</sup> similar performance to VideoBEV [8] without stereo matching (VideoBEV-D) with 60 training epochs.

| Exp | BP | TG | mAP $\uparrow$ | NDS $\uparrow$ | mAOE $\downarrow$ | mAVE $\downarrow$ |
|-----|----|----|----------------|----------------|-------------------|-------------------|
| 1   |    |    | 0.371          | 0.476          | 0.646             | 0.317             |
| 2   | ✓  |    | 0.375          | 0.479          | 0.634             | 0.319             |
| 3   | ✓  | ✓  | <b>0.388</b>   | <b>0.492</b>   | <b>0.596</b>      | <b>0.281</b>      |
| 4   |    | ✓  | 0.378          | 0.484          | 0.602             | 0.316             |

**Table 5:** Ablation of each component. BP means the historical BEV feature backward propagate. TG means TA-GRU.

| Kernel size  | mAP $\uparrow$ | NDS $\uparrow$ | mATE $\downarrow$ | mAVE $\downarrow$ | Training frames | mAP $\uparrow$ | NDS $\uparrow$ | mATE $\downarrow$ | mAVE $\downarrow$ |
|--------------|----------------|----------------|-------------------|-------------------|-----------------|----------------|----------------|-------------------|-------------------|
| 1 $\times$ 1 | 0.388          | 0.492          | 0.647             | 0.281             | 0               | 0.318          | 0.380          | 0.742             | 0.835             |
| 3 $\times$ 3 | 0.397          | 0.500          | 0.632             | 0.284             | 1               | 0.346          | 0.457          | 0.696             | 0.389             |
| 5 $\times$ 5 | <b>0.403</b>   | <b>0.506</b>   | <b>0.615</b>      | 0.276             | 2               | 0.372          | 0.479          | 0.665             | 0.310             |
| 7 $\times$ 7 | 0.398          | 0.501          | 0.630             | <b>0.273</b>      | 4               | 0.390          | 0.493          | 0.646             | 0.283             |
| 9 $\times$ 9 | 0.394          | 0.496          | 0.633             | 0.280             | 8               | <b>0.403</b>   | <b>0.506</b>   | <b>0.615</b>      | <b>0.276</b>      |
|              |                |                |                   |                   | 16              | 0.402          | <b>0.49</b>    | 0.628             | 0.280             |

**Table 6:** Ablation of receptive field.

**Table 7:** Ablation of sequence length.

the correct temporal fusion module for effective back-propagation, as the combination of these two components can achieve a synergistic effect that surpasses the sum of their individual contributions.

**Ablation of Receptive Field.** Receptive field is critical for temporal fusion of moving objects. We explore the kernel size of convolutions in TA-GRU in Tab.6. As we can see, an excessively large or small receptive field can have a detrimental effect on performance. Our suspect is that a fusion module with an overlarge receptive field gathers noisy information, while a small receptive field leads to mismatches for moving objects. Consequently, we use a  $5 \times 5$  kernel size in TA-GRU, unless otherwise specified.

**Ablation of Training Sequence Length.** We use the BEVDet-Depth with no temporal fusion as baseline and apply our method with different training sequence length in Tab.7. The fusion of a single frame greatly enhances velocity prediction (mAVE). The fusion of four to eight frames mainly enhances localisation metrics mAP and mATE, with an increase of 8.5% and 12.7% respectively. Further increasing the number of frames to 16 offers gains limited performance improvement. Therefore, we train our models on 8 frames.

#### 4.6 Effect of Recurrent Long-term Fusion

**Detection of Occlusion Objects.** Intuitively, long-term fusion has the potential to aid in the detection of occluded objects. Therefore, we conducted an analysis of object occlusions on the nuScenes validation set. Firstly, we define a ‘distinct object’ as one where the visibility of the entire object exceeds 60%. Subsequently, we determined that if an object is classified as a ‘distinct object’ in any frame of the video, it is considered visible throughout the video. Finally, we divided the number of visible objects by the total number of objects present to get the overall visibility of the video. As shown in Fig.4a, the visibility of the video has the same trend with mAP, proving the improvement of mAP in long-term fusion originates from the detection of occlusion objects.

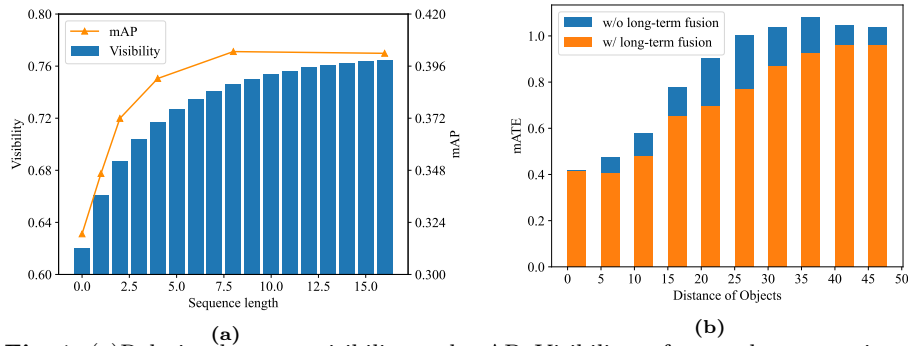


**Fig. 4:** (a)Relation between visibility and mAP. Visibility refers to the proportion of distinct objects in a video, expressed as a percentage of the total number of objects present. The Pearson correlation coefficient between visibility and mAP is 0.977.(b)The mATE metrics with objects at different location on the nuScenes validation set. The blue one is a result from single frame baseline and the orange one is RecurrentBEV. The objects between 25 meters and 30 meters improve the mATE metric by 24% mostly.

**Localization of Distant Objects.** The localization of objects is a major challenge for camera-based 3D detector. due to the uncertainty of depth estimation. We speculate long-term fusion can more accurately localize objects that move from near to far as the ego vehicle moves. Therefore, we calculate the mATE metrics of objects at different distances before and after long-term fusion, as in Fig.4b. Notably, objects between 25 and 30 meters exhibit the most significant improvement in mATE through long-term fusion, whereas nearby objects show marginal gains. This finding demonstrates the efficacy of long-term fusion in accurately localizing distant objects.

## 5 Conclusion and Future Work

In this paper, we investigate the differences between recurrent fusion and parallel fusion of BEV-based 3D object detectors, and propose RecurrentBEV, a simple and effective recurrent temporal fusion method. By incorporating RNN-style back-propagation and a newly designed inner grid transformation, RecurrentBEV enhances long-term fusion capabilities while maintaining latency and memory advantages. Rigorous experiments conducted on the nuScenes dataset demonstrate that RecurrentBEV outperforms all prior methods, while maintaining competitive inference speeds, thereby facilitating deployment. To our knowledge, it represents the first recurrent fusion method with explicit BEV features that surpasses parallel fusion performance.

In the future, we will generalize recurrent fusion idea to more BEV architectures such as different BEV view transform backbones and query-based methods. We will also try to extend RecurrentBEV to other 3D perceptual tasks and datasets such as object tracking and trajectory prediction.

## Acknowledgements

This work is partially supported by the National Key R&D Program of China (under Grant 2023YFB4502200), the NSF of China (under Grants U22A2028, 61925208, 62102399, 62222214, 62341411, 62102398, U20A20227, 62372436, 623-02478, 62302482, 62302483, 62302480), Strategic Priority Research Program of the Chinese Academy of Sciences, (Grant No.XDB0660200, XDB0660201, XDB0660202), CAS Project for Young Scientists in Basic Research(YSBR-029), Youth Innovation Promotion Association CAS and Xplore Prize.

## References

1. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. In: ICLR (2016)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
4. Chen, D., Li, J., Guizilini, V., Ambrus, R.A., Gaidon, A.: Viewpoint equivariance for multi-view 3d object detection. In: CVPR (2023)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI (2017)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
7. Feng, C., Jie, Z., Zhong, Y., Chu, X., Ma, L.: Aedet: Azimuth-invariant multi-view 3d object detection. In: CVPR (2023)
8. Han, C., Sun, J., Ge, Z., Yang, J., Dong, R., Zhou, H., Mao, W., Peng, Y., Zhang, X.: Exploring recurrent long-term temporal fusion for multi-view 3d perception. arXiv preprint arXiv:2303.05970 (2023)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
10. Huang, B., Li, Y., Xie, E., Liang, F., Wang, L., Shen, M., Liu, F., Wang, T., Luo, P., Shao, J.: Fast-bev: Towards real-time on-vehicle bird’s-eye view perception. arXiv preprint arXiv:2301.07870 (2023)
11. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
12. Huang, J., Huang, G.: Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. arXiv preprint arXiv:2211.17111 (2022)
13. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
14. Huang, L., Li, Z., Sima, C., Wang, W., Wang, J., Qiao, Y., Li, H.: Leveraging vision-centric multi-modal expertise for 3d object detection. In: NeurIPS (2023)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NeurIPS (2015)
16. Jiang, Y., Zhang, L., Miao, Z., Zhu, X., Gao, J., Hu, W., Jiang, Y.G.: Polarformer: Multi-camera 3d object detection with polar transformer. In: AAAI (2023)

17. Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: CVPRW (2019)
18. Li, H., Zhang, H., Zeng, Z., Liu, S., Li, F., Ren, T., Zhang, L.: Dfa3d: 3d deformable attention for 2d-to-3d feature lifting. In: ICCV (2023)
19. Li, Y., Chen, Y., Qi, X., Li, Z., Sun, J., Jia, J.: Unifying voxel-based representation with transformer for 3d object detection. In: NeurIPS (2022)
20. Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., Li, Z.: Bevstereo: enhancing depth estimation in multi-view 3d object detection with temporal stereo. In: AAAI (2023)
21. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: AAAI (2023)
22. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022)
23. Li, Z., Yu, Z., Wang, W., Anandkumar, A., Lu, T., Alvarez, J.M.: Fb-bev: Bev representation from forward-backward view transformations. In: ICCV (2023)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. arXiv preprint arXiv:2211.10581 (2022)
26. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
27. Liu, H., Teng, Y., Lu, T., Wang, H., Wang, L.: Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In: ICCV (2023)
28. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: ECCV (2022)
29. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: ICCV (2023)
30. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR (2022)
31. Luo, Z., Zhou, C., Zhang, G., Lu, S.: Detr4d: Direct multi-view 3d object detection with sparse attention. arXiv preprint arXiv:2212.07849 (2022)
32. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: ICCV (2021)
33. Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In: ICLR (2023)
34. Phillion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV (2020)
35. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: ICCV (2023)
36. 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 (2022)
37. Wang, Z., Huang, Z., Fu, J., Wang, N., Liu, S.: Object as query: Lifting any 2d object detector to 3d detection. In: ICCV (2023)
38. Xiong, K., Gong, S., Ye, X., Tan, X., Wan, J., Ding, E., Wang, J., Bai, X.: Cape: Camera view position embedding for multi-view 3d object detection. In: CVPR (2023)
39. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird’s-eye-view recognition via perspective supervision. In: CVPR (2023)

- 40. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
- 41. Zhang, J., Zhang, Y., Liu, Q., Wang, Y.: Sa-bev: Generating semantic-aware bird's-eye-view feature for multi-view 3d object detection. In: ICCV (2023)
- 42. Zhou, H., Ge, Z., Li, Z., Zhang, X.: Matrixvt: Efficient multi-camera to bev transformation for 3d perception. In: ICCV (2023)
- 43. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)