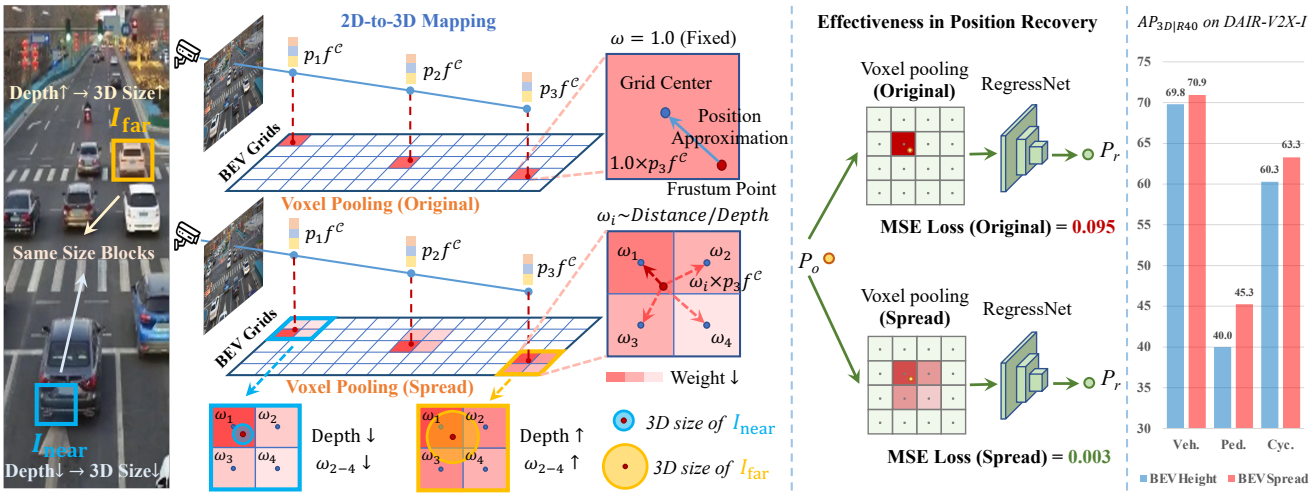
BEVSpread: Spread Voxel Pooling for Bird's-Eye-View Representation in Vision-based Roadside 3D Object Detection

Wenjie Wang[1]*, Yehao Lu[1]*, Guangcong Zheng[1], Shuigen Zhan[2], Xiaoqing Ye[3], Zichang Tan[3]
Jingdong Wang[3], Gaoang Wang[1], Xi Li[1,2,4]†
[1]College of Computer Science and Technology, Zhejiang University
[2]Polytechnic Institute, Zhejiang University [3]Baidu
[4]Zhejiang – Singapore Innovation and AI Joint Research Lab
{wenjie_wang, luyehao, guangcongzheng, shuigenzhan, xilizju}@zju.edu.cn
gaoangwang@intl.zju.edu.cn {yexiaoqing, tanzichang, wangjingdong}@baidu.com



(a) Comparison of Voxel Pooling Methods (Direct) and (Spread) (b) Proof Experiment (c) Performance

Figure 1. **(a)** Original voxel pooling strategy approximately accumulates the image features contained in a frustum point to the single corresponding BEV grid center, leading to an irrecoverable position approximation error. We discover that spread operation can reduce this error, where the weights ω assigned to surrounding BEV grids should be related to the distance and depth. First, weight decay with distance can effectively retain more location information, which is beneficial for subsequent network learning. Second, same size image blocks with deeper depth represent objects of larger 3D scales, which results in distant objects containing few image features. Therefore, it is reasonable to assign larger weights to the surrounding BEV grids for distant targets. **(b)** We have designed an intuitive experiment to demonstrate that network can learn accurate position coordinates from the BEV features obtained by spread voxel pooling. **(c)** Results on DAIR-V2X-I dataset show that BEVSpread outperforms state-of-the-art method by a significant margin of (1.12, 5.26, 3.01) AP in vehicle, pedestrian and cyclist categories, respectively.

Abstract

Vision-based roadside 3D object detection has attracted rising attention in autonomous driving domain, since it encompasses inherent advantages in reducing blind spots and expanding perception range. While previous work mainly focuses on accurately estimating depth or height for 2D-to-3D mapping, ignoring the position approximation error in the voxel pooling process. Inspired by this insight, we propose a novel voxel pooling strategy to reduce such error, dubbed BEVSpread. Specifically, instead of bringing the image features contained in a frustum point to a single BEV grid, BEVSpread considers each frustum point as a source and spreads the image features to the surrounding BEV grids with adaptive weights. To achieve superior propagation performance, a specific weight function is designed

*Equal contribution.
†Corresponding author.

to dynamically control the decay speed of the weights according to distance and depth. Aided by customized CUDA parallel acceleration, BEVSpread achieves comparable inference time as the original voxel pooling. Extensive experiments on two large-scale roadside benchmarks demonstrate that, as a plug-in, BEVSpread can significantly improve the performance of existing frustum-based BEV methods by a large margin of (1.12, 5.26, 3.01) AP in vehicle, pedestrian and cyclist. The source code will be made publicly available at BEVSpread.

1. Introduction

Vision-centric 3D object detection plays a critical role in autonomous driving perception, which helps accurately estimate the state of the surrounding environment and provide reliable observations for forecasting and planning at a low cost. Most existing work focuses on the ego vehicle system [16, 17, 23, 26, 36, 38], facing safety challenges due to a lack of global perspective and the limitation of long-range perception capacity. To this end, roadside 3D object detection has attracted rising attention in recent years [3, 33, 34, 43, 44]. Since roadside cameras are mounted on poles a few meters above the ground, they have inherent advantages in reducing blind spots, improving occlusion robustness, and expanding global perception capability [1, 45–47]. Therefore, it is promising to improve roadside perception performance as a complement to improve the safety of autonomous driving.

Recently, bird's eye view (BEV) has become the mainstream paradigm for handling the 3D object detection task [13, 22], among which frustum-based method [16, 23, 27, 44] is a significant branch and its pipeline is shown in Fig. 1a. It first maps image features to 3D frustums by estimating depth or height, and then pools frustums onto BEV grids by reducing the Z-axis degree of freedom. Extensive work focuses on improving the precision of depth estimation [8–10, 16, 23] or height estimation [17, 39, 44] to improve the performance of 2D-to-3D mapping. However, the approximation error caused by the voxel pooling process is rarely considered. As shown in Fig. 1a, the predicted point is usually not located in a BEV grid center. To improve the computational efficiency, previous work approximately accumulates the image features contained in the predicted point to the single corresponding BEV grid center, leading to a position approximation error, and this error is irrecoverable. Augmenting the density of BEV grids can alleviate this error, but results in a notable increase in computational workload. Especially in roadside scenarios, due to the large perception range and limited computing resources, BEV grids can only be designed relatively sparse to ensure real-time detection, which exactly exacerbates the impact of this error. Thus, the question is raised: How can we reduce this error while maintaining computational complexity?

In this work, we propose a novel voxel pooling strategy to reduce such position approximation error, dubbed BEVSpread. Instead of adding the image features contained in a frustum point to a single BEV grid, BEVSpread considers each frustum point as a source and spreads the image features to the surrounding BEV grids with adaptive weights. We discover that the weights assigned to surrounding BEV grids should be related to distance and depth. First, weight decay with distance can effectively retain more location information, which is beneficial for subsequent network learning. Second, we notice that same size image blocks with deeper depth represent objects of larger 3D scales, which results in distant objects containing few image features. Therefore, it is reasonable to assign larger weights to the surrounding BEV grids for distant targets. Inspired by this insight, a specific weight function is designed to achieve superior spread performance, where weights and distances follow a Gaussian distribution. The variance of this Gaussian distribution is positively related to the depth information, which controls the decay speed. In particular, BEVSpread is a plug-in that can be directly deployed on existing frustum-based BEV methods.

To validate the effectiveness of BEVSpread, extensive experiments are conducted on two challenging benchmarks for vision-based roadside perception, DAIR-V2X-I [46] and Repo3D [45]. After deploying spread voxel pooling strategy, the 3D average precision ($AP_{3D|R40}$) of BEVHeight [44] and BEVDepth [16] increases by a large margin of 3.1 and 4.0 on average across three major categories.

Our contributions can be summarized as:

- We point out a position approximation error existed in current voxel pooling approach, which seriously affects the performance of 3D object detection in roadside scenarios, while this issue is ignored in previous works.
- We propose a novel spread voxel pooling approach, namely BEVSpread, which considers both distance and depth effects during the spread process to reduce the position approximation error while maintaining comparable inference time through CUDA parallel acceleration.
- Extensive experiments demonstrate that, as a plug-in, BEVSpread significantly enhances the performance of existing frustum-based BEV methods by a large margin of (1.12, 5.26, 3.01) AP in vehicle, pedestrian, and cyclist categories, respectively.

2. Related Work

Recently, bird's eye view (BEV) has become the mainstream paradigm for 3D object detection in autonomous driving, as it provides a unified feature space for multi-sensor and clearly presents the location and scale of objects. In this section, we introduce BEV perception, roadside BEV perception and voxel pooling strategy in detail.

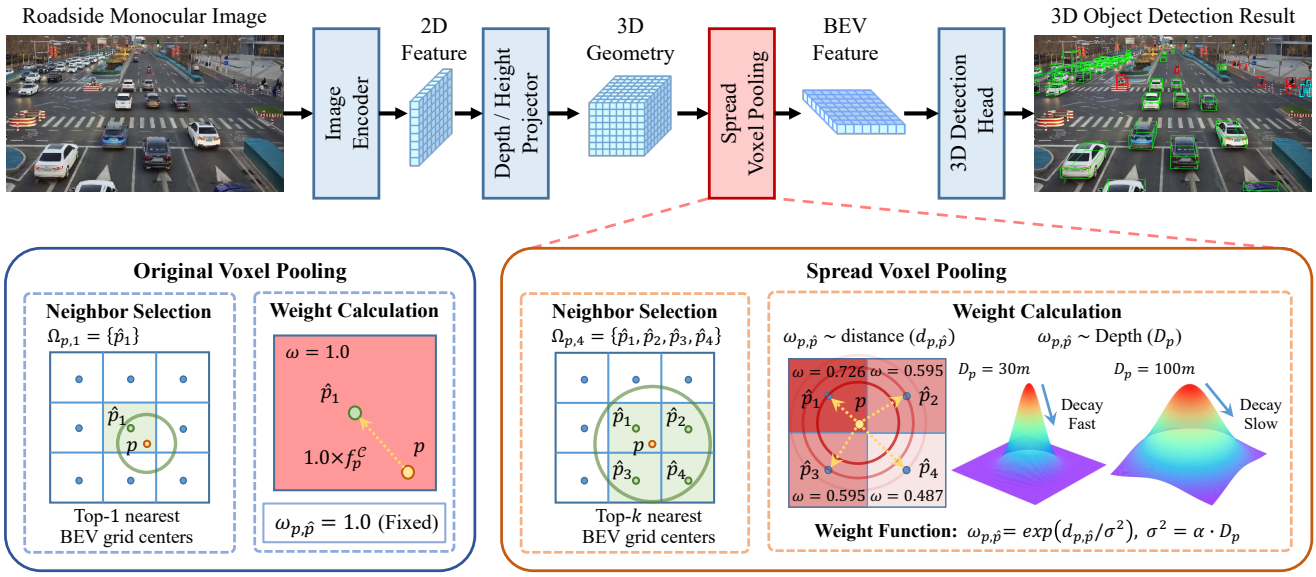


Figure 2. **The overall framework of BEVSpread.** Spread voxel pooling consists of two main steps, Neighbor Selection and Weight Calculation. First, each 3D geometry point p is mapped to BEV space, where $top - k$ nearest BEV grid centers are selected as its neighbors $\Omega_{p,k}$. Correspondingly, the original voxel pooling selects the $top - 1$ nearest BEV grid center as its neighbor $\Omega_{p,1}$. Second, the weights are calculated for the neighbors by Weight Function, where the weights $\omega_{p,\hat{p}}$ and the distances $d_{p,\hat{p}}$ follow a Gaussian distribution with $(0, \sigma^2)$. Furthermore, the variance σ^2 is positively related to depth D_p, which controls the decay speed of $\omega_{p,\hat{p}}$. Ultimately, the image features contained in each 3D geometry point are accumulated to its neighbors according to the calculated weights.

BEV Perception. Based on the sensor types, BEV approaches can be mainly divided into three parts including vision-based [16, 17, 23, 26, 37, 44], LiDAR-based [2, 5, 7, 12, 30, 35] and fusion-based [15, 18, 24, 40, 41] methods. Benefits from its low cost for deployment, vision-based BEV methods have been a topic of great significance, which are further divided into transformer-based and frustum-based schema. Transformer-based methods [11, 17, 19, 20, 36, 37] introduce 3D object queries or BEV grid queries to regress 3D bounding boxes. Frustum-based methods [14, 16, 23, 25, 27, 44] first map image features to 3D frustums by estimating depth or height and then generate BEV features by voxel pooling. This work focuses on the voxel pooling process in frustum-based methods, which has rarely been explored but is critical.

Roadside BEV Perception. Roadside BEV perception is an emerging field, which has been under-explored. BEVHeight [43, 44] first concentrates on roadside perception, which predicts the height distribution to replace the depth distribution. CBR [3] focuses on device robustness, which generates BEV features without extrinsic calibration, while accuracy is limited. CoBEV [29] fuses geometry-centric depth and semantic-centric height cues to further improve performance. MonoGAE [34] considers the prior knowledge of the ground plane. Different from these methods, this paper proposes a plug-in to improve the performance of existing frustum-based BEV methods.

Voxel Pooling Strategy. LSS [23] is the pioneering work of frustum-based BEV methods, where voxel pooling is proposed for the first time. Extensive work follows this setting [14, 16, 25, 44]. SA-BEV [27] proposes a novel voxel pooling strategy, SA-BEVPool, which filters out background information. While the unfiltered out frustum points adopt the same voxel pooling method as LSS. In this work, we focus on eliminating the position approximation error in the voxel pooling process of LSS.

3. Methods

In this section, we first give a brief problem formulation of vision-based roadside 3D object detection. Next, an overall architecture of BEVSpread network is presented. Finally, the core designs of BEVSpread are described in detail.

3.1. Problem Formulation

In this work, we aim to detect 3D bounding boxes of traffic objects from roadside monocular images. Formally, a 3D object detector can be defined as:

$$B = M_\theta(I, E, K) \tag{1}$$

where M_θ is the detection model with the learnable parameters θ, $I \in \mathbb{R}^{H \times W \times 3}$ is the input monocular image, (H, W) represent the height and width of the image, $E \in \mathbb{R}^{3 \times 4}$ and $K \in \mathbb{R}^{3 \times 3}$ are the extrinsic and intrinsic matrix of the roadside camera, respectively. We denote the set of predicted

3D bounding boxes as:

$$B = \{\hat{B}_1, \hat{B}_2, ..., \hat{B}_n\} \tag{2}$$

where n is the number of predicted objects and $\hat{B}$ can be formulated as a vector with 7 degrees of freedom:

$$\hat{B} = (x, y, z, l, w, h, r) \tag{3}$$

where (x, y, z) is the location of the 3D bounding box, (l, w, h) is the length, width and height of the 3D bounding box, and r is the yaw angle relative to one specific axis.

3.2. BEVSpread

Overall Architecture. As shown in Fig. 2, the overall framework consists of four main stages. The image encoder is composed of a ResNet [6] and a SECONDFPN [42], aiming to extract the 2D high-dimensional multi-scale image features $f^{\mathcal{I}} \in \mathbb{R}^{C_{\mathcal{I}} \times \frac{H}{16} \times \frac{W}{16}}$ from a monocular roadside image I, where $C_{\mathcal{I}}$ denotes the channel number. The depth/height projector first takes the 2D image features $f^{\mathcal{I}}$ and camera parameters $(E,\ K)$ as input to predict the depth/height distribution $f^{\mathcal{D}} \in \mathbb{R}^{C_{\mathcal{D}} \times \frac{H}{16} \times \frac{W}{16}}$ and the context features $f^{\mathcal{C}} \in \mathbb{R}^{C_{\mathcal{C}} \times \frac{H}{16} \times \frac{W}{16}}$, where $C_{\mathcal{D}}$ represents the number of depth/height bins and $C_{\mathcal{C}}$ stands for the channels of the context features. These two are further fused through an outer product operation to obtain 2.5D frustum features $f^{2.5D} \in \mathbb{R}^{C_{\mathcal{C}} \times C_{\mathcal{D}} \times \frac{H}{16} \times \frac{W}{16}}$. Then, the projector push the 2.5D frustum features $f^{2.5D}$ into 3D geometry features $f^{3D} \in \mathbb{R}^{X \times Y \times Z \times C_{\mathcal{C}}}$ using camera parameters $(E,\ K)$. The proposed spread voxel pooling strategy splattes the 3D geometry features f^{3D} into an unified BEV features f^{BEV}. Finally, the 3D detection head utilizes the generated BEV features to produce the 3D bounding boxes B.

Top-k Nearest BEV Grids. Define P^{BEV} to represent the set of arbitrary positions in BEV grids, $\dot{P}^{\text{BEV}} \in P^{\text{BEV}}$ to represent the set of BEV grid centers, $\Omega_{p,k} \subseteq \dot{P}^{\text{BEV}}$ to represent the set of top-k neareast BEV grid centers to $p = (x, y) \in P^{\text{BEV}}$. For $\forall \hat{p} = (\hat{x}, \hat{y}) \in \Omega_{p,k}$ and $\bar{p} = (\bar{x}, \bar{y}) \in \{\dot{P}^{\text{BEV}} \backslash \Omega_{p,k}\}$, it should satisfies:

$$\begin{cases} |\Omega_{p,k}| = k, \\ d_{p,\hat{p}} \leq d_{p,\bar{p}} \end{cases} \tag{4}$$

$$d_{p,p'} = \sqrt{(x - x')^2 + (y - y')^2} \tag{5}$$

where $|\cdot|$ denotes the cardinality of a set, k represents the neighbors number of $\forall p \in P^{\text{BEV}}$, $\{\dot{P}^{\text{BEV}} \backslash \Omega_{p,k}\}$ denotes the relative complement of $\Omega_{p,k}$ in $\dot{P}^{\text{BEV}}$, and $d_{p,p'}$ represents the Euclidean distance between $p = (x, y) \in P^{\text{BEV}}$ and $p' = (x', y') \in P^{\text{BEV}}$.

Spread Voxel Pooling. In the spread voxel pooling stage, we first calculate the corresponding positions $p = (x, y) \in P^{\text{BEV}}$ in BEV space for each point $(x, y, z) \in P^{3D}$ in 3D geometry by reducing the Z-axis degree of freedom. Instead of accumulating the included context feature $f^{\mathcal{C}} \in \mathbb{R}^{C_{\mathcal{C}}}$ of p into the corresponding single BEV gird center, we propagate $f^{\mathcal{C}}$ with certain weights to its neighbors Ω, which are the n nearest BEV grids center around p. Specifically, the process of spread voxel pooling can be formulated as:

$$f^{\text{BEV}}_{\hat{p}} = \text{Add}(f^{\text{BEV}}_{\hat{p}}, \omega_{p,\hat{p}} \cdot f^{\mathcal{C}}_{p}), \forall \hat{p} \in \Omega_{p,k} \tag{6}$$

where $\Omega_{p,k}$ is the set of top-k neareast BEV grid centers of p, $f^{\text{BEV}}_{\hat{p}}$ denotes the BEV feature of $\hat{p}$, $\omega_{p,\hat{p}}$ represents the weight of $\hat{p}$ determined by the weight decay function, $f^{\mathcal{C}}$ is the context feature included in p, and $\text{Add}(a, b) = a + b$.

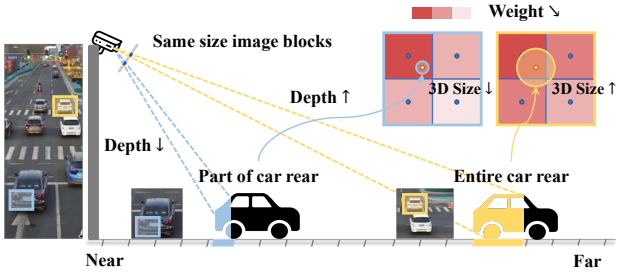


Figure 3. **Effect of depth in voxel pooling.** Same size image blocks with deeper depth represent objects of larger 3D scales, which results in distant objects containing few image features. Therefore, it is reasonable to assign larger weights to the surrounding BEV grids for the distant targets.

Weight Function. We discover that the weights should be related to the distance and depth in spread process. **(a)** Weight decay with distance can retain more location information, which is beneficial to recover the accurate position of $p \in P^{\text{BEV}}$ through subsequent network learning, so as to eliminate the position approximation error in the original voxel pooling process. Additionally, we have designed an intuitive experiment to demonstrate this point in Sec. 4.5. **(b)** As shown in Fig. 3, same size image blocks with deeper depth represent objects of larger 3D scales, resulting in distant objects containing few image features. Therefore, it is reasonable to assign larger weights to the surrounding BEV grids for distant targets, manifesting that the weights decay more slowly with distance, as shown in Fig. 2.

To this end, we design a specific weight function, which ingeniously utilizes a Gaussian function to integrate the distance and depth information. The function is defined as:

$$\omega_{p,\hat{p}} = \exp(\frac{-d^2_{p,\hat{p}}}{\sigma^2}) \tag{7}$$

$$\sigma^2 = \alpha \cdot D_p \tag{8}$$

where $\omega_{p,\hat{p}}$ represents the calculated weight of $\hat{p}$, $d_{p,\hat{p}}$ represents the the Euclidean distance between p and $\hat{p}$, D_p is the predicted depth of p, σ^2 is the variance of Gaussian function which is positively related to D_p and controls the decay

Algorithm 1 Spread Voxel Pooling

INPUT: 3D geometry points $P^{3D} \in \mathbb{R}^{X \times Y \times Z}$, context image feature of each 3D geometry point $f^{\mathcal{C}} \in \mathbb{R}^{Cc}$, depth vector of 3D geometry points D.
OUTPUT: BEV features f^{BEV}.
BEGIN:

$P^{\text{BEV}} \in \mathbb{R}^{X \times Y}$ extracted from P^{3D}
for p in P^{BEV} **do**
  Get $\Omega_{p,k}$ by Eq. (4) ▷ *Top-k Neareast BEV Grids*
  for $\hat{p}$ in $\Omega_{p,k}$ **do**
    $\omega_{p,\hat{p}} \leftarrow \exp(\frac{-d_{p,\hat{p}}}{\alpha \cdot D_p})$ ▷ *Weight Calculation*
    $f^{\text{BEV}}_{\hat{p}} \leftarrow \text{Add}(f^{\text{BEV}}_{\hat{p}}, \omega_{p,\hat{p}} \cdot f^{\mathcal{C}}_{p})$ ▷ *Feature Accumulation*
  end for
end for
return f^{BEV}

END

speed of $\omega_{p,\hat{p}}$, and α is a learnable parameter to maintain σ^2 within the interval [0,2]. Through this function, the weights change adaptively depending on the distance and depth. In summary, the pseudocode of the spread voxel pooling strategy is shown in Algorithm 1.

4. Experiments

In this section, we first introduce two roadside benchmark datasets and the implementation details. Then, we compare our proposed BEVSpread with state-of-the-art methods. Finally, comprehensive ablation studies are conducted to validate the effects of each component.

4.1. Datasets

DAIR-V2X-I. DAIR-V2X [46] is a large-scale dataset for vehicle-infrastructure cooperative autonomous driving, which offers a multi-modal 3D object detection resource. Here, we focus on DAIR-V2X-I subset, containing $10k$ images from mounted cameras to study roadside perception. DAIR-V2X-I involves $493k$ 3D bounding box annotations, spanning distances from 0 to 200 meters. Following the previous work [44], 50%, 20% and 30% images are split into train, validation, and testing, respectively. Noting that the testing set is not yet published and we evaluate the results on the validation set.

Rope3D. Rope3D [45] is another benchmark for roadside 3D object detection, consisting of $50k$ images and over $1.5M$ 3D objects collected across a variety of lighting conditions (daytime / night / dusk), different weather conditions (rainy / sunny / cloudy) and 26 distinct intersections, spanning distances from 0 to 200 meters. Following the split strategy detailed in Rope3D, we use 70% of the images as training, and the remaining 30% as testing.

Metrics. For both DAIR-V2X-I and Rope3D datasets, we report the 40-point average precision ($AP_{3D|R40}$) [31] of 3D bounding boxes, which is further categorized into three modes: Easy, Middle and Hard, based on the box characteristics, including size, occlusion and truncation, following the metrics of KITTI [4].

4.2. Implementation Details

For fair comparison with state-of-the-art methods, we use ResNet-101 [6] as image encoder, BEV grid size is set to 0.4 meters, the range of X axis is set to 0-100 meters, and the neighbors number is set to 6. ResNet-50 and 0.8m grid size are used for ablation studies. Following BEVHeight [44], we adopt image data augmentations including random intrinsic and extrinsic changes. We use AdamW [21] as an optimzer with a learning rate set to $2e-4$. All experiments are conducted on 8 RTX-3090 GPUs.

4.3. Comparison with state-of-the-art

For a comprehensive evaluation, we compare the proposed BEVSpread with state-of-the-art BEV detectors on DAIR-V2X-I and Rope3D. Since the proposed spread voxel pooling strategy is a plug-in, we deploy it to BEVHeight, dubbed BEVSpread. The results are described as follows.

Results on DAIR-V2X-I. Tab. 1 illustrates the performance comparison on DAIR-V2X-I. We compare our BEVSpread with the state-of-the-art vision-based methods, including ImVoxelNet [28], BEVFormer [17], BEVDepth [16] and BEVHeight [44], and the traditional LiDAR-based methods, including PointPillars [12], SECOND [42] and MVXNet [32]. The results demonstrate that BEVSpread outperforms state-of-the-art methods by a significant margin of (1.12, 5.26 and 3.01) AP in vehicle, pedestrian, and cyclist categories, respectively. We notice that previous methods are trained only in 0-100m, while DAIR-V2X-I contains the labels of 0-200m. To this end, we cover a longer range of 3D object detection, locating targets in 0-200m, which is denoted as DAIR-V2X-I* in Tab. 1.

Results on Rope3D. We compare our BEVSpread with the state-of-the-art vision-centric methods, including BEVDepth [16] and BEVHeight [44], on the Rope3D validation set in homologous settings. As shown in Tab. 1, BEVSpread outperforms all other methods across the board, with significant improvements of (2.59, 3.44 and 2.14) AP in vehicle, pedestrian, and cyclist, respectively.

Visualization Results. As shown in Fig. 4, we present the visualization results of BEVHeight [44] and our BEVSpread in the image and BEV view. It can be observed in the upper half that BEVSpread detects the targets which BEVHeight misses in multiple scenes. The main reason is shown in the lower half. Image features show that BEVSpread focuses more attention on the foreground area. And the BEV features generated by BEVSpread are

Table 1. **Comparison $AP_{3D|R40}$ results of 3D object detection on the validation set of DAIR-V2X-I [46] and Rope3D [45]** . ResNet-101 is used as image encoder, the BEV grid size is set to 0.4 meters, and top-k (k=6) nearest BEV grid centers are selected as neighbors. "$*$" denotes covering the longer range between $0\sim200m$, while others cover $0\sim100m$.

Dataset	Method	Modality	Venue	Vehicle (*IoU=0.5*)			Pedestrian (*IoU=0.25*)			Cyclist (*IoU=0.25*)		
				Easy	Middle	Hard	Easy	Middle	Hard	Easy	Middle	Hard
DAIR-V2X-I [46]												
	PointPillars [12]	LiDAR	CVPR' 19	63.07	54.00	54.01	38.53	37.20	37.28	38.46	22.60	22.49
	SECOND [42]	LiDAR	Sensors	71.47	53.99	54.00	55.16	52.49	52.52	54.68	31.05	31.19
	MVXNet [32]	LiDAR & Camera	ICRA' 19	71.04	53.71	53.76	55.83	54.45	54.40	54.05	30.79	31.06
	ImVoxelNet [28]	Camera	WACV' 22	44.78	37.58	37.55	6.81	6.75	6.74	21.06	13.57	13.17
	BEVFormer [17]	Camera	ECCV' 22	61.37	50.73	50.73	16.89	15.82	15.95	22.16	22.13	22.06
	BEVDepth [16]	Camera	AAAI' 23	75.50	63.58	63.67	34.95	33.42	33.27	55.67	55.47	55.34
	BEVHeight [44]	Camera	CVPR' 23	77.78	65.77	65.85	41.22	39.29	39.46	60.23	60.08	60.54
	BEVSpread (**Ours**)	Camera	-	**79.07**	**66.82**	**66.88**	**46.54**	**44.51**	**44.71**	**62.64**	**63.50**	**63.75**
	w.r.t. BEVHeight			+1.29	+1.05	+1.03	+5.32	+5.22	+5.25	+2.41	+3.42	+3.21
DAIR-V2X-I* [46]												
	BEVHeight [44]	Camera	CVPR 23	81.62	75.90	75.94	40.89	38.98	39.18	60.29	60.60	61.13
	BEVSpread (**Ours**)	Camera	-	**82.84**	**77.10**	**77.19**	**43.96**	**42.03**	**42.13**	**62.31**	**64.44**	**64.89**
	w.r.t. BEVHeight			+1.22	+1.21	+1.25	+3.07	+3.05	+2.95	+2.02	+3.84	+3.76
Rope3D [45]												
	BEVDepth [16]	Camera	AAAI 23	76.90	66.91	66.89	30.42	28.08	28.11	55.34	53.53	53.51
	BEVHeight [44]	Camera	CVPR 23	77.93	67.50	67.49	36.26	30.35	30.30	61.49	56.98	56.90
	BEVSpread (**Ours**)	Camera	-	**80.61**	**70.04**	**70.03**	**38.65**	**34.32**	**34.25**	**63.66**	**59.11**	**59.03**
	w.r.t. BEVHeight			+2.69	+2.55	+2.54	+2.39	+3.97	+3.95	+2.17	+2.13	+2.13

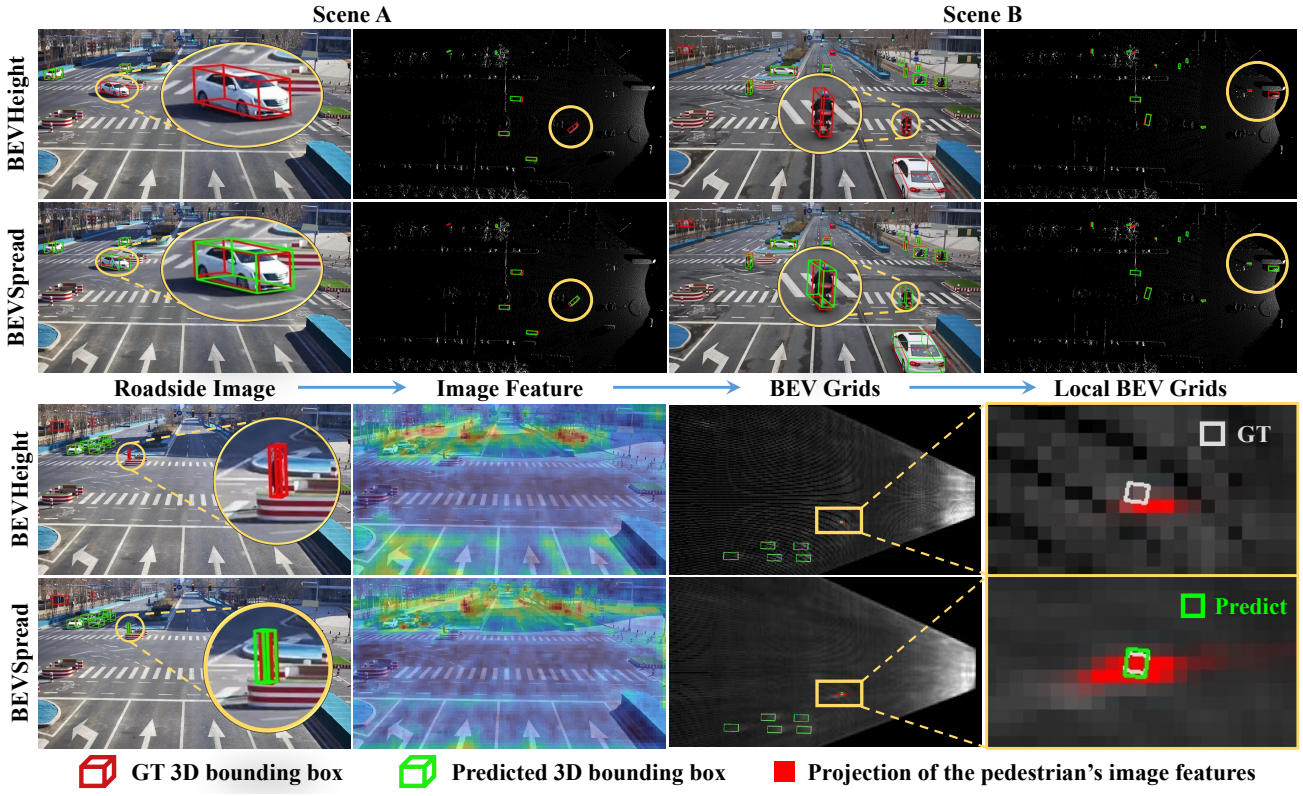


Figure 4. **Visualization results of BEVHeight and our proposed BEVSpread in image and BEV view.** It can be observed in the upper half that BEVSpread detects the targets which BEVHeight have not detected in multiple scenes. The lower half demonstrates the reasons. We notice that BEVHeight misses the pedestrian because no corresponding image features are projected onto the correct BEV grids. However, BEVSpread spreads the image features to the surrounding BEV grids and thus successfully detects the target.

Method	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDepth	0.436	0.330	0.702	0.280	0.535	0.553	0.227
BEVDepth*	0.432	0.325	0.701	0.283	0.572	0.531	0.224
BEVDepth* + ours	**0.450**	**0.327**	**0.688**	**0.275**	**0.489**	**0.470**	**0.217**

Table 2. Comparison on the nuScenes *val* set. The experiment is reproduced based on the official BEVDepth repository with config named bev_depth_lss_r50_256x704_128x128_24e_2key. Both CBGS and EMA are not used. BEVDepth denotes the official result of this config. * denotes the results we reproduce.

smoother than those generated by BEVHeight. BEVHeight misses the pedestrian because no corresponding image features are projected onto the correct BEV grids. While BEVSpread spreads the image features to the surrounding BEV grids and exactly covers the correct BEV grids, so as to successfully detect the target.

4.4. Results on nuScenes.

Our approach specifically targets the roadside scenario. To further assess its robustness, we conduct additional experiment on nuScenes following BEVDepth [16]. Tab. 2 shows that BEVSpread still works in ego-vehicle settings, and the improvement (4.2% NDS) is comparable to that in roadside scenario (5.5% Avg-AP).

4.5. Proof Experiment for Position Recovery

We have designed an intuitive experiment to demonstrate that the proposed spread voxel pooling strategy can achieve accurate position recovery in BEV space. Initially, 10 random vectors of C dimensions representing image features are randomly generated. Then, we randomly generate 3D points and assign for these 10 features. Based on the original voxel pooling and spread voxel pooling, the 3D points are projected onto the 16×16 BEV grids to obtain the BEV features. The U-Net encoder network is utilized to regress the accurate position of the first image feature in the BEV space, and MSE loss is used. Note that the training process contains 5,000 iterations, and the batch size is set to 128 per iteration. The inputs are random for each iteration. The experimental process is shown in Fig. 1. As shown in Fig. 5, our spread voxel pooling recovers the random point position with 0.003 MSE loss when the neighbors number ≥ 3, while the original voxel pooling obtains 0.095 MSE loss.

4.6. Ablation Study

Performance as a plugin. The proposed spread voxel pooling strategy, as a plug-in, can significantly improve the performance of existing frustum-based BEV methods. As shown in Tab. 3, after being deployed to BEVDepth [16], the performance has been significantly improved by a margin of (4.17, 8.93 and 8.2) AP in three categories. After being deployed to BEVHeight [44], the performance has been improved by a margin of (1.55, 5.58 and 7.56) AP in three

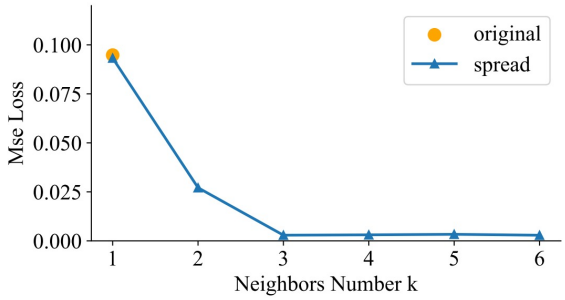


Figure 5. **Proof Experiment for Position Recovery.** Spread voxel pooling recovers the random point position with 0.003 MSE loss when the neighbors number $k \geq 3$, while the original voxel pooling ($k = 1$) obtains 0.095 MSE loss.

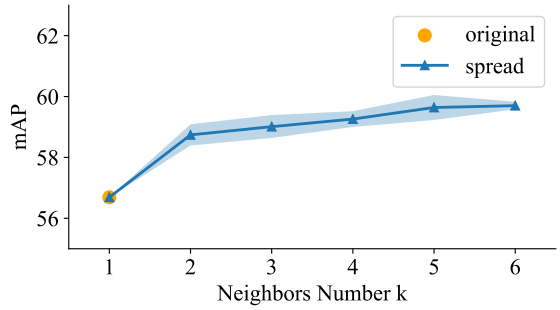


Figure 6. **Hyperparameter sensitivity experiment on neighbors number k.** It can be observed that the performance of $k \geq 2$ is significantly better than $k = 1$ (baseline). As k increases, the performance gradually improves and becomes stable.

categories. It is worth noting that the recognition ability for pedestrian and cyclist has been greatly improved.

Analysis on Neighbor Selection. Fig. 6 shows how the mAP of three categories changes with neighbors number k. For each hyperparameter selection, we repeat 3 times, and the light-blue area indicates the error range. It can be observed that the performance of $k \geq 2$ is significantly better than $k = 1$ (baseline). As k increases, performance gradually improves and becomes stable.

Analysis on Weight Function. We validate the effectiveness of the depth and the learnable parameter α for weight function in Tab. 4. The improvement in three major categories proves that the application of depth and the learnable parameter α allows for better spread performance. When applying both, considerable performance (65.80%, 31.00%, 56.34%) is gained for three categories in middle difficulty.

Analysis on Different Backbones. We further compared BEVSpread with BEVHeight using different backbones. Results of ResNet-50/101 are listed in Tab. 1 and Tab. 3, and experiments for ConvNeXt-B can be found in Tab. 5. Results show that stronger backbones lead to greater performance and our method can further improve it.

Table 3. **Ablation study of spread voxel pooling on the DAIR-V2X-I [46].** ResNet-50 is used as image encoder, the BEV grid size is set to 0.8 meters, and the detection range is set to $0\sim100m$, and top-k (k=2) nearest BEV grid centers are selected as neighbors.

Method	Vehicle (*IoU=0.5*)			Pedestrian (*IoU=0.25*)			Cyclist (*IoU=0.25*)		
	Easy	Middle	Hard	Easy	Middle	Hard	Easy	Middle	Hard
BEVDepth [16]	71.09	60.37	60.46	21.23	20.84	20.85	40.54	40.34	40.32
+ spread voxel pooling	**76.15**	**64.09**	**64.19**	**30.87**	**29.27**	**29.57**	**48.06**	**48.53**	**49.21**
w.r.t. BEVDepth	+5.06	+3.72	+3.73	+9.64	+8.43	+8.72	+7.52	+8.19	+8.89
BEVHeight [44]	76.24	64.54	64.13	26.47	25.79	25.72	48.55	48.21	47.96
+ spread voxel pooling	**77.91**	**65.80**	**65.86**	**32.48**	**31.00**	**31.25**	**54.19**	**56.34**	**56.88**
w.r.t. BEVHeight	+1.67	+1.26	+1.73	+6.01	+5.21	+5.53	+5.64	+8.13	+8.92

Table 4. **Ablation study of weight function on DAIR-V2X-I [46].** ResNet-50 is used as image encoder, the BEV grid size is set to 0.8 meters, and the detection range is set to $0\sim100m$, and top-k (k=2) nearest BEV grid centers are selected as neighbors.

Spacing			Vehicle (*IoU=0.5*)			Pedestrian (*IoU=0.25*)			Cyclist (*IoU=0.25*)		
spread	Depth	α	Easy	Middle	Hard	Easy	Middle	Hard	Easy	Middle	Hard
-	-	-	76.24	64.54	64.13	26.47	25.79	25.72	48.55	48.21	47.96
✓	-	-	77.67 (+1.43)	65.61 (+1.07)	65.69 (+1.56)	31.34 (+4.87)	29.94 (+4.15)	30.08 (+4.36)	53.53 (+4.98)	54.65 (+6.44)	55.17 (+7.21)
✓	✓	-	77.88 (+1.64)	65.79 (+1.25)	65.76 (+1.63)	32.40 (+5.93)	30.97 (+5.18)	31.18 (+5.46)	53.69 (+5.14)	55.54 (+7.33)	55.46 (+7.50)
✓	-	✓	77.71 (+1.47)	65.66 (+1.12)	65.74 (+1.61)	31.72 (+5.25)	30.31 (+4.52)	30.52 (+4.80)	53.97 (+5.42)	55.64 (+7.43)	55.58 (+7.62)
✓	✓	✓	**77.91** (+1.67)	**65.80** (+1.26)	**65.86** (+1.73)	**32.48** (+6.01)	**31.00** (+5.21)	**31.25** (+5.53)	**54.19** (+5.64)	**56.34** (+8.13)	**56.88** (+8.92)

Table 5. **Ablation study of different backbones on the DAIR-V2X-I [46].** ConvNeXt-B is used as image encoder, the BEV grid size is set to 0.4 meters, and the detection range is set to $0\sim100m$, and top-k (k=4) nearest BEV grid centers are selected as neighbors.

Method	Vehicle (*IoU=0.5*)			Pedestrian (*IoU=0.25*)			Cyclist (*IoU=0.25*)		
	Easy	Middle	Hard	Easy	Middle	Hard	Easy	Middle	Hard
BEVHeight (ConvNeXt-B)	78.08	65.99	66.07	41.76	40.84	40.03	58.76	60.69	60.76
BEVSpread (ConvNeXt-B)	**79.29**	**67.03**	**67.09**	**47.06**	**44.97**	**45.14**	**62.34**	**64.14**	**64.60**
w.r.t. BEVHeight	+1.21	+1.04	+1.02	+5.30	+4.13	+5.11	+3.58	+3.45	+3.84

Method	Neighbors	Avg-AP ↑	Latency-Total(ms) ↓	Latency-Pooling(ms) ↓
BEVHeight (ResNet-101)	k = 1	56.69	74.3	5.5
BEVSpread (ResNet-101)	k = 1	56.69(+0.00)	69.8(-6.1%)	0.8
BEVSpread (ResNet-101)	k = 2	58.68(+1.99)	73.9(-0.5%)	4.9
BEVSpread (ResNet-101)	k = 3	59.01(+2.32)	76.6(+3.1%)	7.7
BEVSpread (ResNet-101)	k = 6	59.83(+3.14)	85.6(+15.2%)	15.3
BEVHeight (ResNet-50)	k = 1	55.90	61.4	5.5
BEVSpread (ResNet-50)	k = 1	55.90(+0.00)	57.1(-7.0%)	0.8
BEVSpread (ResNet-50)	k = 2	58.12(+2.22)	61.6(+0.3%)	4.9
BEVSpread (ResNet-50)	k = 3	58.55(+2.65)	64.2(+4.6%)	7.7

Table 6. Speed under different neighbor size k on DAIR-V2X-I.

5. Limitations and Analysis

The proposed spread-voxel pooling brings a certain amount of calculation, resulting in an increase in latency. While our approach is flexible to balance accuracy and speed by adjusting the spread scope, which is denoted as neighbor size k. As shown in Table 6, when k=2, BEVSpread still achieves significant improvement in Avg-AP without latency increase, benefiting from our CUDA optimization. Besides, the coordinates of these spread points are calculated online in this version. During the practical deployment phase, BEVSpread can use a preprocessing look-up table, akin to BEVPoolv2, for enhanced acceleration.

6. Conclusion

In this paper, we point out a approximation error in the current voxel pooling method. We proposed a novel voxel pooling strategy named BEVSpread to reduce this error. BEVSpread considers each frustum point as a source and spreads the image features to the surrounding BEV grids with adaptive weights. Additionally, a specific weight function is designed to dynamically control the decay speed based on distance and depth. Experiments in DAIR-V2X-I and Rope3D show that BEVSpread significantly improves the performance of existing frustum-based BEV methods.

Acknowledgements. This work is supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020016, National Natural Science Foundation of China under Grant U20A20222, National Science Foundation for Distinguished Young Scholars under Grant 62225605, Zhejiang Key Research and Development Program under Grant 2023C03196, Baidu, SupreMind and The Ng Teng Fong Charitable Foundation in the form of ZJU-SUTD IDEA Grant, 188170-11102.

References

[1] Christian Creß, Walter Zimmer, Leah Strand, Maximilian Fortkord, Siyi Dai, Venkatnarayanan Lakshminarasimhan, and Alois Knoll. A9-dataset: Multi-sensor infrastructure-based dataset for mobility research. In IV, pages 965–970, 2022. 2

[2] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In CVPR, pages 8458–8468, 2022. 3

[3] Siqi Fan, Zhe Wang, Xiaoliang Huo, Yan Wang, and Jingjing Liu. Calibration-free bev representation for infrastructure perception, 2023. arXiv:2303.03583. 2, 3

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In ICCV, pages 3354–3361, 2012. 5

[5] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In CVPR, 2020. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 4, 5

[7] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In AAAI, pages 969–979, 2022. 3

[8] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection, 2022. arXiv:2203.17054. 2

[9] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment, 2022. arXiv:2211.17111.

[10] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view, 2021. arXiv:2112.11790. 2

[11] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In AAAI, pages 1042–1050, 2023. 3

[12] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019. 3, 5, 6

[13] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, Hao Tian, Enze Xie, Jiangwei Xie, Li Chen, Tianyu Li, Yang Li, Yulu Gao, Xiaosong Jia, Si Liu, Jianping Shi, Dahua Lin, and Yu Qiao. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe, 2022. arXiv:2209.05324. 2

[14] 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, 2022. arXiv:2209.10248. 3

[15] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. NeurIPS, 35:18442–18455, 2022. 3

[16] 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. In AAAI, pages 1477–1485, 2023. 2, 3, 5, 6, 7, 8, 4

[17] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In ECCV, pages 1–18, 2022. 2, 3, 5, 6

[18] 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. NeurIPS, 35:10421–10434, 2022. 3

[19] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In ECCV, pages 531–548, 2022. 3

[20] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In ICCV, pages 3262–3272, 2023. 3

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. arXiv:1711.05101. 5

[22] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey, 2022. arXiv:2208.02797. 2

[23] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In ECCV, pages 194–210, 2020. 2, 3

[24] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. In ICCV, pages 8690–8699, 2023. 3

[25] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. In CVPR, pages 8555–8564, 2021. 3

[26] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection, 2018. arXiv:1811.08188. 2, 3

[27] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Sabev: Generating semantic-aware bird's-eye-view feature for multi-view 3d object detection. In ICCV, 2023. 2, 3

[28] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In WACV, pages 2397–2406, 2022. 5, 6

[29] Hao Shi, Chengshan Pang, Jiaming Zhang, Kailun Yang, Yuhao Wu, Huajian Ni, Yining Lin, Rainer Stiefelhagen, and Kaiwei Wang. Cobev: Elevating roadside 3d object detection with depth and height complementarity, 2023. arXiv:2310.02815. 3

[30] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-

voxel feature set abstraction for 3d object detection. In CVPR, 2020. 3

[31] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In ICCV, 2019. 5

[32] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In ICRA, pages 7276–7282, 2019. 5, 6

[33] Jiayao Tan, Fan Lyu, Linyan Li, Fuyuan Hu, Tingliang Feng, Fenglei Xu, and Rui Yao. Dynamic v2x autonomous perception from road-to-vehicle vision, 2023. arXiv:2310.19113. 2

[34] Jiayao Tan, Fan Lyu, Linyan Li, Fuyuan Hu, Tingliang Feng, Fenglei Xu, and Rui Yao. Monogae: Roadside monocular 3d object detection with ground-aware embeddings, 2023. arXiv:2310.00400. 2, 3

[35] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In CVPR, pages 13520–13529, 2023. 3

[36] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In ICCV, 2023. 2, 3

[37] 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 CoRL, pages 180–191, 2022. 3

[38] Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Frustumformer: Adaptive instance-aware resampling for multi-view 3d detection. In CVPR, pages 5096–5105, 2023. 2

[39] Yiming Wu, Ruixiang Li, Zequn Qin, Xinhai Zhao, and Xi Li. Heightformer: Explicit height modeling without extra data for camera-only 3d object detection in bird's eye view, 2023. arXiv:2307.13510. 2

[40] Yichen Xie, Chenfeng Xu, Marie-Julie Rakotosaona, Patrick Rim, Federico Tombari, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. Sparsefusion: Fusing multi-modal sparse representations for multi-sensor 3d object detection. In ICCV, 2023. 3

[41] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3d object detection. In ICCV, pages 18268–18278, 2023. 3

[42] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. Sensors, 18(10):3337, 2018. 4, 5, 6

[43] Lei Yang, Tao Tang, Jun Li, Peng Chen, Kun Yuan, Li Wang, Yi Huang, Xinyu Zhang, and Kaicheng Yu. Bevheight++: Toward robust visual centric 3d object detection, 2023. arXiv:2309.16179. 2, 3

[44] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen. Bevheight: A robust framework for vision-based roadside 3d object detection. In CVPR, pages 21611–21620, 2023. 2, 3, 5, 6, 7, 8, 1, 4

[45] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task. In CVPR, pages 21341–21350, 2022. 2, 5, 6

[46] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In CVPR, pages 21361–21370, 2022. 2, 5, 6, 8, 1, 3

[47] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, Juan Song, Jirui Yuan, Ping Luo, and Zaiqing Nie. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In CVPR, pages 5486–5495, 2023. 2