Learning High-resolution Vector Representation from Multi-Camera Images for 3D Object Detection

Zhili Chen[1†], Shuangjie Xu[1], Maosheng Ye[1], Zian Qian[1], Xiaoyi Zou[2], Dit-Yan Yeung[1], and Qifeng Chen[1✉]

[1]HKUST [2]DeepRoute.AI

{zchenei, shuangjie.xu, myeag, zqianaa}@connect.ust.hk, xiaoyizou@deeproute.ai, {dyyeung, cqf}@cse.ust.hk

Abstract. The Bird's-Eye-View (BEV) representation is a critical factor that directly impacts the 3D object detection performance, but the traditional BEV grid representation induces quadratic computational cost as the spatial resolution grows. To address this limitation, we present a new camera-based 3D object detector with high-resolution vector representation: VectorFormer. The presented high-resolution vector representation is combined with the lower-resolution BEV representation to efficiently exploit 3D geometry from multi-camera images at a high resolution through our two novel modules: vector scattering and gathering. To this end, the learned vector representation with richer scene contexts can serve as the decoding query for final predictions. We conduct extensive experiments on the nuScenes dataset and demonstrate state-of-the-art performance in NDS and inference time. Furthermore, we investigate query-BEV-based methods incorporated with our proposed vector representation and observe a consistent performance improvement. Project page at `https://github.com/zlichen/VectorFormer`.

Keywords: Multi-view 3D Object Detection · Bird's-Eye-View

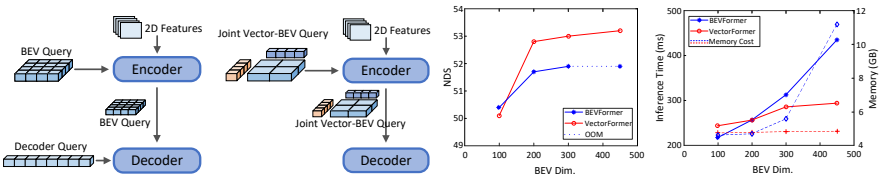
1 Introduction

Integrating multi-camera information into a powerful, unified representation has been a challenging task in the perception system for autonomous driving cars and robotics [20, 38]. To better fuse features from different cameras with various viewpoints, BEV (bird's-eye-view) is proposed, which constructs a unified discrete spatial space in which features are transformed through forward projection [12, 40] or backward projection [25, 26].

One mainstream approach to lifting multi-camera features into BEV space is through backward projection methods, with BEVFormer [25] pioneering this technique by utilizing transformers and deformable attention to sample camera

[†]Work done during an internship at DeepRoute.AI.

[✉]Corresponding author.



(a) The Query-BEV Architecture **(b)** Our proposed VectorFormer **(c)** Resolution & NDS **(d)** Resolution & Resource Cost

Fig. 1: Comparison between the typical query-BEV architecture [25] and our proposed VectorFormer with the novel vector query. Compared to the traditional design in Fig. 1a, our vector queries in Fig. 1b are encoded with finer-grained scene contexts, which transform into more accurate 3D predictions by decoder. Fig. 1c and Fig. 1d are the effectiveness and efficiency comparisons. Compared to the architecture in Fig. 1a, the performance (Fig. 1c) of VectorFormer keeps benefitting by scaling up the representation resolution without leading to a long inference time and high memory cost (Fig. 1d). Noted that BEVFormer will be out-of-memory (OOM) when training under the BEV resolution of 450×450 with NVIDIA A100 40GB GPUs.

features for each BEV location. Works of [12,35,46,52] follow a similar pipeline while replacing with polar representations or more efficient operations including sparse queries. To sum up, the standard BEV formulation process could be two steps: 1) construct BEV grids and associate them with image features according to the projection matrix. 2) Update the feature of each grid in the BEV space through learnable grid sampling based on the constructed association of the previous step [25]. However, we observe an apparent dilemma in the BEV representation. The BEV grid with higher spatial resolution can lead to a decent perception performance, as illustrated in Fig. 1c. It is primarily due to the fact that a more fine-grained BEV grid results in a higher sampling frequency in world coordinates, which ensures a more precise association with the image features [20]. As a trade-off, the computational cost (Fig. 1d and Tab. 4) of the BEV formulation grows quadratically with the BEV grid resolution because of the denser sampling. To solve the overhead of BEV representation formulation, a series of works [4, 11, 55] design efficient operations for view transformations through hashing the corresponding position calculation, which has predominantly focused on the input resolution and optimized the 2D-to-3D projection. However, they have not addressed the optimization of the BEV resolution itself.

Different from the prior approaches to solving this problem, we propose our VectorFormer, as shown in Fig. 1b. It is based on vector factorization on the crucial regions into two low-rank tensor components, which consists of a pair of learnable vector queries that represent the x-axis and y-axis to compress those scenario representations at a finer granularity. As the first time proposing vector factorization by vector queries in query-BEV-based methods, we can encode crucial regions at higher BEV resolution into the compact vector queries and further decode into the 3D predictions. Consequently, our model achieves superior performance even when operating under comparable resolutions in Fig. 1c

(e.g., BEVFormer with BEV size 200×200 v.s. ours with BEV size 150×150 and vector tensor size 200). Furthermore, we represent the innovative sparse high-resolution (HR) BEV features for the critical regions through our vector query scattering module and then interact with the multi-view image features, achieving a substantial reduction in both time and memory complexity. This stands in stark contrast to the $O(n^2)$ complexity observed in previous BEV methods, as our vector representation introduces a more efficient $O(n)$ complexity. This significant reduction contributes directly to the overall improvement in both inference time and memory efficiency within our proposed framework, as shown in Fig. 1d.

Our contributions are summarized as follows:

- We introduce the VectorFormer, which incorporates a novel Vector representation, a first for query-BEV-based methods. Our approach leverages the factorization philosophy which enables efficient modeling of spatial and temporal at a higher resolution. It consistently achieves performance gains as the resolution increases.
- We design the Vector Query Scattering and the Vector Query Gathering modules to learn the expressive high-resolution Vector queries from images. Thanks to the proposed vector representation, our approach benefits from fine BEV granularity with a more efficient $O(n)$ complexity, making additional overhead negligible.
- We conduct extensive experiments on the challenging nuScenes [1] and Waymo [44] datasets. VectorFormer outperforms state-of-the-art camera-based 3D object detectors and is comparable to leading depth-aware methods.

2 Related Work

2.1 Sparse Query 3D detector

As transformer-based methods like DETR [2] generate great performance gain in 2D object detection [2,16,19,42,49,58], researchers begin to explore its potential in 3D object detection [28–33,37,43,46,50,53]. As the pioneer of extending DETR to 3D object detection, DETR3D [50] projects the 3D object center back to 2D multi-view images and iteratively updates the object query from the corresponding sampled image features. Although DETR3D is very efficient, the limitation still exists. The inaccurate object location predicted previously will cause error accumulation in the image feature sampling process. Besides, the model has less information on the global features. To solve the above limitations, PETR [32] and PETRv2 [32] transfer 2D multi-view image features into 3D perception features through 3D position embedding. Sparse4D [28] and StreamPETR [46] aggregating temporal information to help the detection of the current object query. Although sparse query 3D detectors have a simple structure and are efficient, they can still hardly reach the state-of-the-art performance, while our method can reach the state-of-the-art performance while only inducing negligible additional overhead in computational cost.

2.2 Dense BEV Feature 3D Detector

The dense bird's-eye view (BEV) feature 3D detector has drawn great attention in recent years due to its strong performance and wide application in autonomous driving [10, 12, 21–25, 40, 41, 51, 52]. Different from sparse query detectors, dense BEV feature detectors tend to build an explicit BEV feature representation from multi-view 2D images and further utilize it for other perception tasks. BEVFormer [25] and BEVFormerv2 [52] build a BEV query and utilize spatial cross-attention to update the BEV query from the 2D multi-view images with the sampled points in 3D space. Besides, they also propose a temporal self-attention to fuse the temporal features. Based on BEVformer, OCBEV [41] has an object-aligned temporal fusion module to align the fast-moving objects, and a heatmap to provide prior knowledge of the position of the objects in the decoding stage. Unlike BEVFormer, BEVDet [12] and BEVDet4D [10] attempt to build BEV features in a forward way. They first extract the features from multi-view 2D images and then transform them into BEV with the help of the predicted depth and further encode them into BEV features.

Although BEV feature representation has rich information, the computation cost of building the query from multi-view 2D images is also expensive. The computational cost will grow quadratically as the resolution of the BEV query increases, therefore limiting the information contained within the BEV features. Our method can handle the BEV query with a high resolution while inducing negligible additional computational cost.

3 Method

We demonstrate the overall architecture of VectorFormer in Fig. 2. Inheriting the genre of query-BEV 3D detectors [25, 52], it consists of the image backbone, encoder, and decoder head. We mainly focus on innovating the encoder and the decoder head for high-resolution (HR) Vector representation learning. We introduce the HR BEV features factorization by two low-rank Vector queries in Sec. 3.1. In Sec. 3.2, we demonstrate how to locate the sparse informative regions and further guide the HR BEV features composition for those regions by the Vector queries. Later, we present how we interact the LR and sparse HR BEV features with the multi-view image features in a unified manner at Sec. 3.3. In Sec. 3.4, we update the Vector queries by gathering information from the learned HR BEV features with the multi-head cross-attention mechanism [45] (`MHCA`). By further enhancing through the temporal modeling (Sec. 3.5), the Vector queries are considered superior decoding queries and further transform into accurate predictions in the decoder.

3.1 Vector Query

In the real world, the objects have a much smaller scale compared to the whole perception range [1, 6]. We argue that not all regions are equally important,

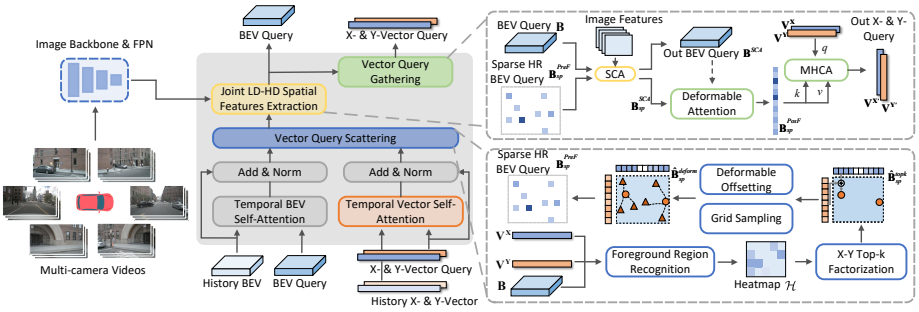


Fig. 2: The overall framework of our proposed VectorFormer. In the middle, the encoder takes the BEV query and Vector query at the current timestamp and the ones from the previous timestamp as inputs for interacting with multi-view image features at each layer. The learned representations are further transformed into 3D predictions by the decoder. We highlight our innovations for learning high-resolution vector representation within the encoder and zoom in on the designs of Vector Query Scattering (Fig. 3 for the details) and the Vector Query Gathering at the right. In the Vector Query Scattering module, we represent the sparse high-resolution (HR) BEV features by first recognizing the foreground regions and then compositing them with the factorized Vector queries of $\mathbf{V^X}$ and $\mathbf{V^Y}$. After jointly interacting with the image features, we finally factorize the learned HR BEV features back into $\mathbf{V^X}$ and $\mathbf{V^Y}$ through the Vector Query Gathering module.

and we instead overcome the limited computational resources by learning the BEV features at high resolution for a sparse set of crucial regions. Inspired by the factorization philosophy in the field of volume rendering [3, 8, 13], we propose to factorize the sparse HR BEV features by two vector-shaped queries of $\mathbf{V^X} \in \mathbb{R}^{W_{HR} \times C}$ and $\mathbf{V^Y} \in \mathbb{R}^{H_{HR} \times C}$. Additionally, we initialize their accompanying learnable positional embeddings, $\mathbf{PE^X}$ and $\mathbf{PE^Y}$, having the same dimensionality as $\mathbf{V^X}$ and $\mathbf{V^Y}$.

With two low-rank tensors $\mathbf{V^X}$ and $\mathbf{V^Y}$, the creation of a sparse HR BEV feature is achieved to enhance granularity. Our work introduces three distinct BEV representations: 1) Traditional BEV: Denoted as $\mathbf{B} \in \mathbb{R}^{H_{LR} \times W_{LR} \times C}$, this corresponds to the low-resolution full-grid BEV query. 2) Sparse BEV: Comprising sparse sets of the LR and HR BEV features, denoted as $\mathbf{B}sp \in \mathbb{R}^{N_{LR} \times C}$ and $\hat{\mathbf{B}}sp \in \mathbb{R}^{N_{HR} \times C}$, respectively. The latter enriches contextual features with minimal additional computational overhead.

Data Representation The factorization operates on:

- Vector query representation $\mathbf{V^X} = (\mathbf{c^x}, \mathbf{v^x})$, where $\mathbf{c^x} \in \mathbb{R}^{W_{HR} \times 2}$ denotes the 2D center coordinate set of all vector cells, which is represented as $(x, 0)$ uniformly located along the x-axis vector representation. $\mathbf{v^x} \in \mathbb{R}^{W_{HR} \times C}$ denotes the feature set of all vector cells. Similar definition could be derived for $\mathbf{V^Y}$, $\mathbf{PE^X}$ and $\mathbf{PE^Y}$.
- Sparse HR BEV query representation is defined as $\hat{\mathbf{B}} = (\mathbf{c}, \mathbf{b})$, where $\mathbf{c}$ is the sparse 2D coordinate set and $\mathbf{b}$ are their corresponding composited HR

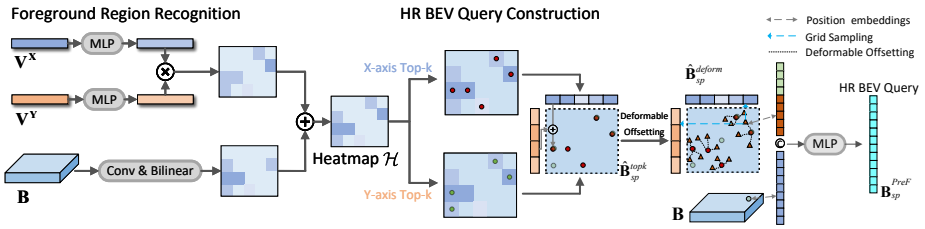


Fig. 3: The design details for the Vector Query Scatter Module of our proposed VectorFormer. We recognize the foreground regions by first predicting an objectness heatmap using high-resolution (HR) Vector queries $\mathbf{V^X}$ and $\mathbf{V^Y}$ and low-resolution (LR) BEV queries as input and then locate the foreground regions by taking the directional Top-k ($k = 1$ here as demonstration) on the heatmap. Next, we apply deformable offsetting for learning to cover more informative regions. We finally construct the sparse HR BEV by fusing the factorized Vector queries and the grid sampled LR BEV features for these informative regions.

BEV features. We utilize an operation $f(\cdot)$ to obtain the sparse HR BEV queries which takes the $\mathbf{V^X}$, $\mathbf{V^Y}$, and $\mathbf{c} = [\mathbf{c^x}.\mathbf{x}, \mathbf{c^y}.\mathbf{y}]$ as input:

$$\begin{aligned} \hat{\mathbf{B}}_{sp} &= f(\mathbf{V^X}, \mathbf{V^Y}, \mathbf{c}), \\ \mathbf{b} &= \texttt{GridSample}(\mathbf{V^X}, \mathbf{c^x}) + \texttt{GridSample}(\mathbf{V^Y}, \mathbf{c^y}), \end{aligned} \tag{1}$$

where we utilize an operation $\texttt{GridSample}(\mathbf{F}, \mathbf{C})$ [15] to bilinearly sample features by taking features $\mathbf{F}$ and the 2D coordinates $\mathbf{C}$ as input.

3.2 Vector Query Scattering

Throughout the BEV encoding module, the Vector queries $\mathbf{V^X}$ and $\mathbf{V^Y}$ are learned to encode with the finer-grained foreground contexts. The LR BEV queries are uniformly sampled, which are learned with more background contexts. As shown in Fig. 3, the Vector Query Scattering module needs first to utilize a heatmap to recognize the critical foreground areas and eventually composite a sparse set of finer-grained BEV queries for the important regions deformed from the proposed areas indicated by the heatmap. These finer-grained BEV queries will exploit the spatial geometry of these important regions from the image features.

Foreground Region Recognition To predict the grid-shape heatmap from the Vector queries, which indicates the probability of the objectness, we first apply an MLP with two hidden layers to the Vector queries. Then, we conduct matrix multiplication between $\mathbf{V^X}$ and $\mathbf{V^Y}$ resulting in the heatmap $\mathbf{h} \in \mathbb{R}^{H_{HR} \times W_{HR}}$. We apply a lightweight convolution module with two layers to the LR BEV feature map, resulting in another heatmap $\mathbf{h}'$. The final heatmap $\mathcal{H}$ is formed by adding the $\mathbf{h}$ with the bilinear upsampled heatmap of $\mathbf{h}'$:

$$\begin{aligned} \mathbf{h} = \texttt{MLP}_1(\mathbf{V^X})\texttt{MLP}_2(\mathbf{V^Y})^T, \mathbf{h}' = \texttt{Conv}(\mathbf{B}), \\ \mathcal{H} = \mathbf{h} + \texttt{Upsample}(\mathbf{h}'), \end{aligned} \tag{2}$$

where the `Conv` module transforms the BEV query $\mathbf{B}$ by $256 \rightarrow 64 \rightarrow 1$, followed by a bilinear upsampling layer denoted as `Upsample`. We use a gaussian focal loss [12, 17, 27, 56] to supervise the the heatmap predictions, resulting in $\mathcal{L}_{hm} = \mathcal{L}_{focal}(\mathcal{H}) + \mathcal{L}_{focal}(\mathbf{h}')$.

HR BEV Query Construction We select the foreground regions along x and y directions separately instead of locating them globally. This practice allows us to derive a uniform number of sparse HR BEV queries for each vector cell of $\mathbf{V^X}$ and $\mathbf{V^Y}$, which can further enable us to parallelly aggregate the learned sparse HR BEV queries into the vector queries via Vector Query Gathering module, as discussed in Sec. 3.4. Specifically, we take the predicted heatmap as input and then apply the Top-k operation along the H and W dimensions, as illustrated in Fig. 3. We obtain k proposal positions for each entry of both vector queries, resulting in a sparse set of 2D coordinates $\mathbf{c}_{topk}$ with a total number of $(W_{HR} + H_{HR}) * k$. The sparse HR BEV queries $\hat{\mathbf{B}}_{sp}^{topk} \in \mathcal{R}^{(W_{HR}+H_{HR})*k*C}$ for these proposal positions are constructed as defined in Eqn. 1:

$$\hat{\mathbf{B}}_{sp}^{topk} = f(\mathbf{V^X}, \mathbf{V^Y}, \mathbf{c}_{topk}). \tag{3}$$

However, the directional selection strategy might result in constructing sparse HR BEV queries located in the less important areas, especially for the ones at the margin of the perception range. To further exert the expressiveness for their representing local areas, we deform the proposal positions to more informative regions, namely deformable offsetting. Inspired by [5, 58], we predict δ offsets with an `MLP` for each proposal position by taking the previously constructed sparse HR BEV queries as input.

As a result, we ground each vector cell on Vector queries with $k*\delta$ of 2D positions on the BEV grid defined at a higher resolution. We obtain the coordinate set $\mathbf{c}_{deform}$ has a total number of $(W_{HR}+H_{HR})*k*\delta$ after deformable offsetting. The sparse HR BEV queries $\hat{\mathbf{B}}_{sp}^{deform}$ with a dimensionality of $\mathbb{R}^{(W_{HR}+H_{HR})*k*\delta*C}$ is obtained as Eqn. 1:

$$\hat{\mathbf{B}}_{sp}^{deform} = f(\mathbf{V^X}, \mathbf{V^Y}, \mathbf{c}_{deform}). \tag{4}$$

However, approximating the BEV with the vector representation only would definitely bring information loss. Therefore, we jointly interact the LR BEV query $\mathbf{B}$ and the sparse HR BEV query $\hat{\mathbf{B}}_{sp}^{deform}$ with the image features. To extract the complement information from the image features, we additionally let the sparse HR BEV queries aware of their corresponding LR BEV queries. Specifically, we first obtain their spatially aligned sparse LR BEV queries through the operation of `GridSample`. Finally, we fuse the features through an MLP, which takes the concatenation of the sparse LR and HR BEV queries as input:

$$\mathbf{B}_{sp} = \texttt{GridSample}(\mathbf{B}, \mathbf{c}_{deform}), \tag{5}$$

$$\hat{\mathbf{B}}_{sp}^{PreF} = \texttt{MLP}([\mathbf{B}_{sp}, \hat{\mathbf{B}}_{sp}^{deform}]), \tag{6}$$

where $\hat{\mathbf{B}}_{sp}^{PreF}$ are the resulting pre-fused sparse HR BEV queries before interacting with the image features.

3.3 Joint LR-HR Spatial Features Extraction

The LR and the sparse HR BEV queries exploit the 3D world geometric from image features through deformable-attention [58] according to the sampled 3D-2D mapping reference points. As discussed in Sec. 3.2, the sparse HR BEV queries $\hat{\mathbf{B}}_{sp}$ are derived from a finer-grained BEV grid, which has a more precise 3D-2D coordinates mapping. Compared to uniformly defined LR BEV queries $\mathbf{B}$, they focus on learning finer-grained BEV features for the foreground objects. For efficiency, we use a shared spatial cross-attention (SCA) module from BEVFormer [25] to jointly interact with the image features to exploit the spatial geometric. The SCA module takes the BEV queries at different resolution granularity as attention queries, and the multi-view image features are considered keys and values:

$$[\mathbf{B}^{SCA}, \hat{\mathbf{B}}_{sp}^{SCA}] = \texttt{SCA}\big([\mathbf{B}, \hat{\mathbf{B}}_{sp}^{PreF}], \mathbf{F}_t, \mathcal{P}[p_{LR}, p_{HR}])\big), \tag{7}$$

where $[\cdot]$ denotes the operation of concatenation, $\mathbf{B}$ and $\hat{\mathbf{B}}_{sp}^{PreF}$ are LR BEV queries and sparse HR BEV queries, $\mathbf{F}_t$ is the multi-view image features are used as keys and values, $\mathcal{P}$ is the 3D-2D projection function, p_{LR} and p_{HR} are the 3D coordinates of the reference points correspond to the LR and the sparse HR BEV queries, respectively.

The LR and sparse HR BEV queries independently interact with the image features to extract complement geometric features. Compared to the sparse HR BEV queries, LR BEV queries are evenly defined in the 3D world, which contains global environmental semantics. On the other hand, the sparse HR BEV queries are more foreground-focused and represent these regions at a finer-grained resolution. Before factorizing these sparse HR BEV queries into the Vector queries of $\mathbf{V}^{\mathbf{X}}$ and $\mathbf{V}^{\mathbf{Y}}$, we enhance the sparse HR BEV queries with more scene contexts by applying a deformable attention [58] to the LR BEV queries:

$$\hat{\mathbf{B}}_{sp}^{PosF} = \texttt{Deform}(\hat{\mathbf{B}}_{sp}^{SCA}, \mathbf{B}^{SCA}, \mathbf{c}_{deform}) + \hat{\mathbf{B}}_{sp}^{SCA}, \tag{8}$$

where the sparse HR BEV features $\hat{\mathbf{B}}_{sp}^{SCA}$ regarded as the attention queries, the LR BEV features $\mathbf{B}^{SCA}$ as keys and values. The extracted contexts feature from LR BEV features have a skip connection to $\hat{\mathbf{B}}_{sp}^{SCA}$, eventually output with the post-fused $\hat{\mathbf{B}}_{sp}^{PosF}$.

3.4 Vector Query Gathering

With the learned sparse HR BEV queries from the image features, we further aggregate them into vector queries through Vector Query Gathering. Benefiting from the evenly derived $k*\delta$ sparse HR queries for each vector cell, we can utilize the multi-head cross-attention mechanism `MHCA` [45] to query information from the learned HR queries back to both of the vector queries $\mathbf{V}^{\mathbf{X}}$ and $\mathbf{V}^{\mathbf{Y}}$.

For the aggregation of $\mathbf{V}^{\mathbf{X}}$, it absorbs the first $N_{HR}^{x} = W_{HR} * k * \delta$ of the sparse HR BEV queries which are derived from each cell of $\mathbf{V}^{\mathbf{X}}$ along the side

of the y-axis. And the remaining $N^y_{HR} = H_{HR} * k * \delta$ of those will squeeze into $\mathbf{V^Y}$. To remedy the information loss of the dimension about to collapse while conducting vector query gathering, we append the vector positional embeddings of each other to the sparse HR BEV queries, which are obtained by:

$$\begin{aligned} \mathbf{pe^x} &= \texttt{GridSample}(\mathbf{PE^X}, \mathbf{c}_{deform}), \\ \mathbf{pe^y} &= \texttt{GridSample}(\mathbf{PE^Y}, \mathbf{c}_{deform}), \\ \mathbf{pe}_{sp} &= \left[\mathbf{pe^y}_{[0:N^x_{HR}]}, \mathbf{pe^x}_{[N^x_{HR}:(N^x_{HR}+N^y_{HR})]}\right], \end{aligned} \tag{9}$$

where we slice the first N^x_{HR} positional embeddings from $\mathbf{pe^y}$ for the sparse HR BEV query about to aggregate into $\mathbf{V^X}$. The following N^y_{HR} HR BEV queries that will go to $\mathbf{V^Y}$ uses the corresponding positional embeddings from $\mathbf{pe^x}$.

The Vector Query Gathering process can be formulated as follows:

$$\begin{aligned} &[\mathbf{V^{X'}}, \mathbf{V^{Y'}}] = \texttt{MHCA}(q, k, v, q_{pos}, k_{pos}), \\ &q = [\mathbf{V^X}, \mathbf{V^Y}], q_{pos} = [\mathbf{PE^X}, \mathbf{PE^Y}], \\ &k = v = \hat{\mathbf{B}}^{PosF}_{sp}, k_{pos} = \mathbf{pe}_{sp}, \end{aligned} \tag{10}$$

where the attention query q is the concatenation of layer input $\mathbf{V^X}$ and $\mathbf{V^Y}$, and their corresponding positional embeddings $\mathbf{PE^X}$ and $\mathbf{PE^Y}$ are regarded as the query position embeddings q_{pos}. The sparse HR BEV features $\hat{\mathbf{B}}^{PosF}_{sp}$ are considered attention keys and values. The key position embeddings are $\mathbf{pe}_{sp}$. We then obtain the resulting $\mathbf{V^{X'}}$ and $\mathbf{V^{Y'}}$, which will be passed to the following encoding layer for further refinements.

3.5 Model Architecture Details

Temporal Features Extraction helps to reason about the existence of highly occluded objects and infer the motion of the objects. Similar to the BEV representation [25], the encoded Vector query from the previous frame can also be considered as strong priors, which could further improve the scene understanding ability. We apply multi-head self-attention [45] (`MHSA`) among vector queries of the previous frame and those of the current frame and use the average operation to fuse the attention output.

Vector Queries As Decoding Queries The Vector queries are encoded with strong spatial and temporal contexts. Different from the traditional practices [25, 52,58] that use randomly initialized learnable embeddings as the decoder queries, we use our learned Vector queries as the decoding query to enhance the decoding head. The total number of the decoding queries equals the summation length of the vector queries representing the x-axis and y-axis with the number of $H_{HR} + W_{HR}$. Inspired by [16], the vector queries at intermediate encoder layers are fed into the decoder and supervised to make 3D predictions for network training acceleration.

Method	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDet† [12]	V2-99 [18]	48.8	42.4	52.4	24.2	37.3	95.0	14.8
BEVDet4D† [10]	Swin-B [34]	56.9	45.1	**51.1**	**24.1**	38.6	**30.1**	12.1
PETRv1† [32]	V2-99 [18]	50.4	44.1	59.3	24.9	38.3	80.8	13.2
PETRv2† [33]	V2-99 [18]	58.2	49.0	56.1	24.3	36.1	34.3	**12.0**
BEVFormer [25]	V2-99 [18]	56.9	48.1	58.2	25.6	37.5	37.8	12.6
VectorFormer	V2-99 [18]	**58.3** (+1.4)	**49.2** (+1.1)	56.3	25.2	**35.2**	33.8	12.7

Table 1: Comparison of recent works on nuScenes detection [1] test set. Methods trained with CBGS [57] are indicated with †. The best results among methods are in bold.

Method	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDet† [12]	Swin-T [34]	41.7	34.9	63.7	26.9	49.0	91.4	26.8
DETR3D [50]	ResNet-101	42.5	34.6	77.3	26.8	38.3	84.2	21.6
DETR4D [37]	ResNet-101	50.9	42.2	68.8	26.9	38.8	49.6	**18.4**
PETRv1† [32]	ResNet-101	44.2	37.0	71.1	26.7	38.3	86.5	20.1
PETRv2† [33]	ResNet-101	52.4	42.1	68.1	26.7	35.7	37.7	18.6
3DPPE [43]†	ResNet-101	45.8	39.1	67.4	28.2	39.5	83.0	19.1
OCBEV [41]	ResNet-101	53.2	41.7	62.9	27.3	**33.9**	34.2	18.7
AeDet§ [7]	ResNet-101	50.6	39.4	**60.9**	**26.6**	41.2	42.0	20.1
BEVFormer-S [25]	ResNet-101	47.9	37.0	72.1	28.0	40.7	43.6	22.0
VectorFormer-S	ResNet-101	51.0 (+3.1)	40.5 (+3.5)	67.6	27.3	38.9	39.9	19.2
DFA3D-S [21]	ResNet-101	50.1	40.1	72.1	27.9	41.1	39.1	19.6
VectorFormer-DFA3D-S	ResNet-101	51.4 (+1.3)	40.4 (+0.3)	68.1	27.4	35.4	36.4	20.5
BEVFormer-B [25]	ResNet-101	51.7	41.6	67.3	27.4	37.2	39.4	19.8
VectorFormer-B	ResNet-101	53.2 (+1.5)	42.5 (+0.9)	64.3	27.5	35.2	34.4	18.8
DFA3D-B [21]	ResNet-101	53.1	43.0	65.4	27.1	37.4	34.1	20.5
VectorFormer-DFA3D-B	ResNet-101	**54.0** (+0.9)	**43.7** (+0.7)	64.3	27.0	36.3	**32.4**	18.6

Table 2: Comparison of recent works on nuScenes detection [1] validation set. Methods trained with CBGS [57] are indicated with †. Results reproduced for fair comparison are indicated as §. The best results among methods are in bold.

4 Experiments

4.1 Experimental Setup

nuScenes Dataset We conduct our experiments on the nuScenes [1] dataset with 1000 scenes. Each scene sample contains six-view RGB images. We evaluate the 3D detectors with the metrics of NDS, mAP, mATE, mASE, mAOE, mAVE, and mAAE as the existing works [21, 25].

Waymo Open Dataset We evaluate our method on the Waymo dataset [44], which provides five-view images covering 252° horizontal FOV. We follow the settings as [20, 25] to experiment on the same subset of the training split for a fair comparison and evaluate with the metrics of LET-3D-mAP, LET-3D-mAH, and LET-3D-mAL [14].

Implementation Details For the nuScenes dataset [1], we followed the typical practice as [25, 50] and adopted the ResNet101-DCN [5, 9] with pretrained checkpoint from FCOS3D [48] as image backbone to process the images with the resolution of 1600×900. Similarly, BEV queries are initialized with the size of

Method	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
BEVDet4D [10]	Swin-B [34]	51.5	39.6	61.9	26.0	36.1	39.9	18.9
PETRv2 [33]	V2-99 [18]	50.3	41.0	72.3	26.9	45.3	38.9	19.3
SA-BEV† [54]	ConvNeXt-B [36]	57.9	47.9	-	-	-	-	-
StreamPETR [46]	V2-99 [18]	57.1	48.2	61.0	**25.6**	37.5	**26.3**	19.4
SparseBEV* [31]	V2-99 [18]	57.9	49.4	-	-	-	-	-
VectorFormer	V2-99 [18]	**60.5**	**51.7**	**57.0**	26.6	**22.8**	28.7	**18.7**

Table 3: Comparison of recent works trained with no more than 24 epochs on nuScenes detection [1] validation set using large backbone. † indicates methods trained with CBGS [57]. * indicates a method with dual branches design. The best results among methods are in bold.

Method	BEV Dim.	x- and y-axis Vector Dim.	NDS ↑	mAP ↑	FPS	Mem. (GB)
BEVFormer [25]	200×200	-	51.7	41.6	3.9	4.70
Ours	150×150	1×200 & 200×1	52.8	41.8	3.9	4.78
BEVFormer [25]	300×300	-	51.9	41.4	3.2	5.58
Ours	200×200	1×300 & 300×1	53.0	42.1	3.5	4.83
BEVFormer [25]	450×450	-	-	-	2.3	11.21
Ours	200×200	1×450 & 450×1	53.2	42.5	3.4	4.85

Table 4: Effectiveness and efficiency comparisons between BEVFormer [25] and our proposed VectorFormer. BEVFormer with a BEV dimension of 450×450 could not be trained with NVIDIA A100 40G GPUs because of the expensive memory consumption. Frames-per-second (FPS) are tested with NVIDIA RTX4090 GPU.

200×200 for the base setting and 150×150 for the small setting. The vector queries are initialized with a length of 450. The numbers of encoding/decoding layers and used history frames are maintained the same as BEVFormer [25] under similar model settings. To scale up our method, we experiment with the large image backbone of V2-99 [18] with the pretrained DD3D [39] checkpoint. On the Waymo dataset [44], we followed the practice as [20, 25] to make a fair comparison. All models are trained with 24 epochs with a learning rate of 2×10^{-4} as the baseline [25].

4.2 Main Results

We demonstrate the 3D detection results of our proposed VectorFormer on the nuScenes [1] test set in Tab. 1 and validation set in Tab. 2 and Tab. 3. We also evaluate our method on the Waymo dataset [44] in Tab. 5.

nuScenes Dataset Without using any bells and whistles, we achieve the best performance and present with remarkable improvements against BEVFormer [25], with 1.4 points higher on NDS (58.3% vs. 56.9%) and 1.1 points higher on mAP (49.2% vs. 48.1%) on nuScenes [1] test set, as shown in Tab. 1.

We also compare our small and base settings with previous state-of-the-art methods on the nuScenes [1] validation set in Tab. 2. Besides, we extend our

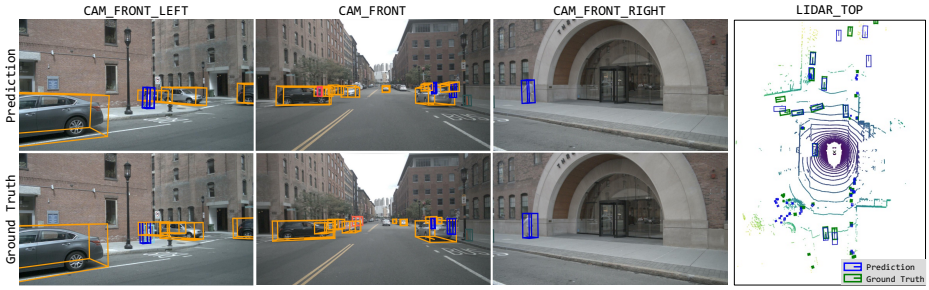


Fig. 4: Visualization results of VectorFormer on nuScenes [1] validation set. Detection predictions with ground truth in multi-view camera images are shown on the left and in bird's-eye-view is shown on the right.

Methods	LET-mAPL↑	LET-mAPH↑	LET-mAP↑
MV-FCOS3D++ [47]	**37.9**	48.8	52.2
BEVFormer [25]	35.0	47.1	51.0
VectorFormer	36.8	**49.1**	**53.2**

Table 5: Comparison on Waymo validation set.

proposed vector representation to a recent work of DFA3D [21]. Concretely, VectorFormer-S outperforms the baseline BEVFormer-S [25] with 3.1 points in NDS and 3.5 points in mAP. When extending our method to DFA3D [21], our VectorFormer-DFA3D-S consistently achieves a leap upon DFA3D-S [21] with 1.3 points in NDS and 0.3 points in mAP. Regarding the base setting, our VectorFormer-B presents improvements of 1.5 points in NDS and 0.9 points in mAP compared to the BEVFormer-B [25]. When extended to the DFA3D-B, we fine-tuned from the DFA3D's [21] pretrained checkpoint by freezing the image backbone to save the GPU memory cost. It is observed that we can further boost the performance by achieving a higher NDS of 54.0%. It is worth mentioning that our proposed VectorFormer-S and VectorFormer-B, which do not utilize depth information, still present a superior NDS against DFA3D-S [21] (51.0% vs. 50.1%) and a comparable NDS to DFA3D-B [21] (53.2% vs. 53.1%). When scaling up the method with a larger image backbone, we also demonstrate superior results upon the recent SOTAs, as shown in Tab. 3.

Overall, we achieve state-of-the-art performance, and our proposed vector query representation can significantly improve query-BEV-based methods.

Representation Dimension and Computation Overhead In Tab. 4, we illustrate the performance and computation overhead details of Fig. 1. When growing the BEV resolution, the traditional framework shows performance vanishing and leads to computational cost explosions, failing to take advantage of the increases in the representation dimension. In comparison, our method with the vector representation enjoys finer BEV granularity construction with $O(N)$ complexity. Comparing under a similar level of BEV resolution, we can speed

Heatmap	Spatial	Temporal	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
-	-	-	48.4	38.1	72.8	27.8	39.9	46.2	**19.7**
✓	-	-	48.9	38.8	72.1	27.8	41.3	43.5	20.0
✓	✓	-	49.5	39.3	**69.6**	27.6	41.4	**42.6**	20.6
✓	✓	✓	**49.7**	**39.8**	70.9	**27.4**	**38.8**	44.5	20.0

Table 6: The ablation studies of different components in our proposed VectorFormer. Heatmap indicates using heatmap supervision for the BEV features map, Spatial indicates incorporating our proposed vector representations for spatial modeling, and Temporal indicates conducting temporal modeling for the vector representation.

Vector Combin.	NDS	mAP	mATE	mASE	mAOE
Mult.	49.4	39.4	**70.2**	27.6	42.3
Add.	**49.7**	**39.8**	70.9	**27.4**	**38.8**

Table 7: The ablation study on the combination of vector query.

Top-k Proposals	NDS	mAP	mATE	mASE	mAOE
2	49.5	39.6	69.6	27.9	41.7
3	49.7	39.8	70.9	27.4	38.8
4	50.0	40.0	69.0	27.3	40.2

Table 8: The ablation study on the number of proposal positions.

up the FPS by up to 47.8% and save the GPU memory consumption by up to 56.7% while achieving better performance.

Qualitative Results As shown in Fig. 4, our proposed VectorFormer presents a conspicuous detection performance. Even for objects with severe occlusion or located further apart, the model can still produce satisfactory bounding boxes.

Waymo Dataset In Tab 5, we further evaluate our method on the dataset of Waymo [44]. Our proposed VectorFormer consistently achieves better results towards the baselines.

4.3 Ablation Study

All the experiments in this section are conducted with VectorFormer-S by default with 12 training epochs.

Effect of Model Components As illustrated in Tab. 6, we present the ablation studies on the effect of different component designs in VectorFormer. It is demonstrated that our proposed framework design for learning high-resolution vector representation is effective, and its components work in synergy to uplift the detection performance with 1.3 points in NDS and 1.7 points in mAP.

Effect of Vector Query Combination We factorize the high-resolution (HR) BEV features into x and y vector features. We conduct ablation studies on the operators of multiply and addition for combining these factorized vectors to build HR BEV features, as illustrated in Tab. 7. It is shown that combining the vectors to build HR BEV features by addition leads to better performance.

Effect of Foreground Proposal Number As shown in Tab. 8, we conduct ablation studies on the number of foreground proposal positions along the x and

Offset Pred.	LR-HR Fusion	Pos. Emb.	NDS	mAP	mATE	mASE	mAOE
-	✓	✓	**49.7**	39.2	70.0	27.7	39.7
✓	-	✓	49.6	39.3	**69.0**	**27.4**	41.3
✓	✓	-	49.3	39.1	70.6	27.7	41.6
✓	✓	✓	**49.7**	**39.8**	70.9	**27.4**	**38.8**

Table 9: The ablation study on offsetting the proposal positions, conducting fusion on sparse HR BEV queries with the LR BEV queries, and using positional embeddings in the Vector query gathering module.

y directions. Overall, the performance will improve as we increase the number of proposals, and we choose to use three proposals in each direction in practice.
Effect of Offset Prediction We represent the foreground proposal position with HR BEV queries and further adaptively deform the positions with predicted offsets. This practice improves the mAP with 0.4 points, as illustrated in the first and the fourth rows of Tab. 9.
Effect of LR-HR Fusion The evenly sampled LR BEV features and sparse HR BEV features independently interact with the image features. Therefore, we apply fusion between LR and HR features before (Eqn. 6) and after (Eqn. 8) passing the SCA module, and it results in improvements in overall performance, as shown in the second and the fourth rows of Tab. 9.
Effect of Positional Embeddings The sparse HR query located at (x, y) is composited according to the Eqn. 1. When aggregating the sparse HR query at (x, y) to $\mathbf{V^X}$ or $\mathbf{V^Y}$, we use their positional embeddings $\mathbf{PE^Y}$ or $\mathbf{PE^X}$ that correspond to the dimension about to collapse to remedy the information loss (Eqn. 9). Comparing the third and the fourth rows In Tab. 9, it is shown that the usage of positional embeddings as the attention keys contributes to the overall performance improvements.

5 Conclusion

In this paper, we have presented a camera-based 3D object detector, VectorFormer, accompanied by a novel representation of Vector query. Addressing the limitations of traditional BEV queries, which incur substantial computational costs and memory usage as spatial resolution increases, we propose to utilize a more lightweight representation of vector query, focusing on learning finer-grained representations for the crucial regions through our designed vector query scattering and gathering modules. The vector queries compact richer spatial and temporal priors with low complexity of time and memory, which are further used to enhance the decoder to produce more accurate predictions. The extensive experiments demonstrate that our proposed VectorFormer achieves state-of-the-art performance, and the designed framework is generalizable, which can consistently leap the performance of the query-BEV methods.

Acknowledgements

The authors are thankful for the financial support from the Hetao Shenzhen-HongKong Science and Technology Innovation Cooperation Zone (HZQB-KCZYZ-2021055), this work was also supported by Shenzhen Deeproute.ai Co., Ltd (HZQB-KCZYZ-2021055).

References

1. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
4. Chen, S., Cheng, T., Wang, X., Meng, W., Zhang, Q., Liu, W.: Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. arXiv preprint arXiv:2206.04584 (2022)
5. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
6. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8458–8468 (2022)
7. Feng, C., Jie, Z., Zhong, Y., Chu, X., Ma, L.: Aedet: Azimuth-invariant multi-view 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21580–21588 (2023)
8. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
11. Huang, J., Huang, G.: Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. arXiv preprint arXiv:2211.17111 (2022)
12. 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)
13. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9223–9232 (2023)

14. Hung, W.C., Kretzschmar, H., Casser, V., Hwang, J.J., Anguelov, D.: Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection. arXiv preprint arXiv:2206.07705 (2022)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)
16. Jia, D., Yuan, Y., He, H., Wu, X., Yu, H., Lin, W., Sun, L., Zhang, C., Hu, H.: Detrs with hybrid matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19702–19712 (2023)
17. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV). pp. 734–750 (2018)
18. 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: CVPR workshop (2019)
19. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
20. Li, H., Sima, C., Dai, J., Wang, W., Lu, L., Wang, H., Xie, E., Li, Z., Deng, H., Tian, H., et al.: Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. arXiv preprint arXiv:2209.05324 (2022)
21. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6684–6693 (2023)
22. 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: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1486–1494 (2023)
23. 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: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1477–1485 (2023)
24. Li, Z., Lan, S., Alvarez, J.M., Wu, Z.: Bevnext: Reviving dense bev frameworks for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20113–20123 (2024)
25. 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: European conference on computer vision. pp. 1–18. Springer (2022)
26. Li, Z., Yu, Z., Wang, W., Anandkumar, A., Lu, T., Alvarez, J.M.: Fb-bev: Bev representation from forward-backward view transformations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6919–6928 (2023)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
28. 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)
29. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
30. Lin, X., Pei, Z., Lin, T., Huang, L., Su, Z.: Sparse4d v3: Advancing end-to-end 3d detection and tracking. arXiv preprint arXiv:2311.11722 (2023)
31. Liu, H., Teng, Y., Lu, T., Wang, H., Wang, L.: Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18580–18590 (2023)

32. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
33. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X., Sun, J.: Petrv2: A unified framework for 3d perception from multi-camera images. arXiv preprint arXiv:2206.01256 (2022)
34. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
35. Liu, Z., Chen, S., Guo, X., Wang, X., Cheng, T., Zhu, H., Zhang, Q., Liu, W., Zhang, Y.: Vision-based uneven bev representation learning with polar rasterization and surface estimation. In: Conference on Robot Learning. pp. 437–446. PMLR (2023)
36. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
37. Luo, Z., Zhou, C., Zhang, G., Lu, S.: Detr4d: Direct multi-view 3d object detection with sparse attention. arXiv preprint arXiv:2212.07849 (2022)
38. Ma, Y., Wang, T., Bai, X., Yang, H., Hou, Y., Wang, Y., Qiao, Y., Yang, R., Manocha, D., Zhu, X.: Vision-centric bev perception: A survey. arXiv preprint arXiv:2208.02797 (2022)
39. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)
40. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020)
41. Qi, Z., Wang, J., Wu, X., Zhao, H.: Ocbev: Object-centric bev transformer for multi-view 3d object detection. arXiv preprint arXiv:2306.01738 (2023)
42. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse detr: Efficient end-to-end object detection with learnable sparsity. arXiv preprint arXiv:2111.14330 (2021)
43. Shu, C., Deng, J., Yu, F., Liu, Y.: 3dppe: 3d point positional encoding for transformer-based multi-camera 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3580–3589 (2023)
44. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
46. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3621–3631 (October 2023)
47. Wang, T., Lian, Q., Zhu, C., Zhu, X., Zhang, W.: Mv-fcos3d++: Multi-view camera-only 4d object detection with pretrained monocular backbones. arXiv preprint arXiv:2207.12716 (2022)

48. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913–922 (2021)
49. Wang, T., Yuan, L., Chen, Y., Feng, J., Yan, S.: Pnp-detr: Towards efficient visual analysis with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4661–4670 (2021)
50. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
51. Wu, Y., Li, R., Qin, Z., Zhao, X., Li, X.: Heightformer: Explicit height modeling without extra data for camera-only 3d object detection in bird's eye view. arXiv preprint arXiv:2307.13510 (2023)
52. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
53. Yao, J., Lai, Y.: Dynamicbev: Leveraging dynamic queries and temporal context for 3d object detection. arXiv preprint arXiv:2310.05989 (2023)
54. 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)
55. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13760–13769 (2022)
56. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
57. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. arXiv:1908.09492 [cs] (2019)
58. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)