



# Pose-Enhanced 3D Rotary Embedding for Multi-View 3D Object Detection

Ke Sheng<sup>1,2</sup>, Huiying Xu<sup>1,2(✉)</sup>, and Xinzhong Zhu<sup>1,2</sup>

<sup>1</sup> Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua 321004, Zhejiang, China  
cnbatmovean@zjnu.edu.cn

<sup>2</sup> School of Computer Science and Technology of Zhejiang, Normal University, Jinhua 321004, Zhejiang, China

**Abstract.** 3D object detection basing multi-view cameras has gained widespread attention in recent years. Sparse query-based approaches are particularly favored for their low computational cost. However, these methods rely on the transformer architecture, which has limitations in capturing the spatial positional information crucial for accurate 3D object localization. Most current methods still use absolute positional embedding similar to 2D detection tasks, lacking relative positional representations tailored for 3D tasks. To address this, we propose a relative positional embedding method called 3D Rotary Position Embedding (3DRoPE). This method embeds 3D positional information into queries and keys using rotation, then leveraging the properties of rotational calculations to incorporate relative positional information into the subsequent attention weights. Additionally, we introduce a learnable parameter for 3D positional information, allowing the model to adjust to different scales of relative relationships through 3DRoPE. To mitigate the impact of multi-view geometric information on relative position calculations, we incorporate the geometric information into the image's positional embedding, indirectly enhancing the model's understanding of different viewpoints as well. 3DRoPE demonstrates superior localization and detection performance compared to previous absolute positional embedding methods on the nuScenes dataset, achieving improvements of 2.4% in mAP and 2.0% in NDS.

**Keywords:** Autonomous Driving · 3D Object Detection · 3D Position Embedding · Multi-view Images

## 1 Introduction

As a rapidly advancing technology, autonomous driving is being actively developed by automotive companies globally, either through in-house innovation or partnerships with technology firms. Autonomous driving technology can be divided into three main areas: perception, decision-making, and control, with perception serving as the system's "eyes" and "ears". In the domain of autonomous driving perception, 3D object detection constitutes a pivotal task. Image-based 3D object detection has garnered extensive attention from both the academic and industrial communities due to its cost-effectiveness and ease

of deployment compared to alternative sensor technologies. Now autonomous vehicles are equipped with surround-view cameras, making 3D object detection from the BEV (Bird's Eye View) perspective [1–7] a prominent research focus and a highly beneficial advancement for perception tasks in autonomous driving.

The mainstream 3D object detection methods based on BEV can be divided into two paradigms. A type of approach [1–3, 6–9] is to transform multi-view images into a unified BEV representation, and then perform subsequent detection tasks in the BEV space. Lift-Splat-Shoot (LSS) [1] is the pioneering method that utilizes multi-view geometric and predicted depth information to transform image features into the BEV space, then BevFormer [3] introduced the transform structure to implicitly establish a continuous connection between image features and the BEV space. Another category of approaches, termed sparse query-based methods [4, 5, 10–13], draws inspiration from DETR [14]. Unlike methods relying on BEV feature conversion, these approaches utilize global 3D queries to represent objects. Through a decoder, the queries interact with image features and iteratively update themselves to achieve object detection. DETR3D [10] is the first to draw inspiration from DETR-style architecture. While these sparse query-based methods offer significant computational cost and efficiency advantages compared to other methods, they are still constrained by limitations in accuracy. Therefore, we aim to build on the advantages of this type of approach, making our own improvements to address issues related to accuracy.

Accurate localization of the object is essential for both 2D object detection and 3D object detection tasks. PETR [5] considers the importance of object localization in the 3D task; it introduces 3D position embedding allowing queries to directly learn the spatial information of objects in the images, without extra projection efforts like DETR3D. While PETR encodes spatial relationships through ray-based embeddings that represent directional vectors from the camera's optical center to image plane pixels, this approach lacks explicit depth priors, resulting in coarse geometric localization that limits precise alignment between 2D features and their 3D counterparts. 3DPPE [12] introduces 3D point position embedding instead of the camera-ray method; it designs a depth estimation network to estimate the depth information of image pixels and generate the 3D point positions of image features using depth information. OPEN [15] builds upon 3DPPE by estimating object-level depth information instead of pixel-level depth information, making the depth information more accurate. VeDet [16] designs a unified geometric representation that aligns images and 3D queries, then replaces the previous coordinate system-based position embedding with geometric position embedding. Although the aforementioned methods have achieved varying degrees of improvement, they all employ absolute position embedding. Absolute position embedding treats each position independently and lacks representation of the relative position relationships between locations. Through research on how these 3D sparse query methods perform object position prediction, we found that they use the positional information learned from the attention mechanism as an offset to update the 3D query's position. Therefore, it is clear that the model requires not only the absolute position information of the query and image but also their relative position relationships; more importantly, this is crucial for the model to achieve more accurate object localization. To address this issue, we propose a 3D relative position embedding method suitable for sparse query-based methods.

Inspired by Rotary Position Embedding (RoPE) [17] from Large Language Model (LLM), which cleverly embeds position information into queries and keys using rotations based on complex numbers, and the relative positional relationships are explicitly reflected in the attention weights during subsequent attention calculations. We design a relative position embedding method called 3D Rotary Position Embedding (3DRoPE) for 3D position embedding based on RoPE. We divide the position embedding into three parts, each used to embed positional information for the three dimensions, then combine the three parts into a complete position embedding. Just like RoPE, our method allows relative positional relationships to be reflected in the attention weights, but in our case, these relationships are in 3D space. Additionally, considering RoPE's ability to capture relative positional relationships at different scales based on varying rotation angles. To address this, we introduce a learnable parameter for 3D positional information, allowing the model to adjust relative positional relationships across different scales. Moreover, normalization of 3D positional information for model generalization can blur relative positions across different views in multi-view scenarios and potentially affect the 3D query's accuracy in capturing object positions. Therefore, we incorporate the geometric information from multi-view cameras into 3D positional embeddings of each view's images to mitigate the impact of multi-view information distortion on relative position calculations.

To demonstrate the feasibility of our method, we conduct extensive experiments on the nuScenes dataset [18]. The results demonstrate that our method performs exceptionally well, particularly in terms of object localization. In summary, our contributions are:

- 1) We propose a novel relative position embedding method for multi-view 3D object detection, enhancing object localization through explicit relative position calculations and through the introduction of a learnable parameter for 3D positional information, allowing the model to capture relative positions at different scales.
- 2) To mitigate the impact of multi-view information on relative position calculations, we incorporate the geometric information of multi-view cameras into the position embeddings of images at each viewpoint.
- 3) We demonstrate the effectiveness of our method on the nuScenes dataset, showing significant improvements in object localization and accuracy compared to previous absolute position embedding methods, achieving a 2.0% increase in mAP and a 2.8% increase in NDS.

## 2 Related Work

### 2.1 Multi-view 3D Object Detection

Initially, researchers conducted monocular 3D detection on each view separately and then merged the results [19]. Today, state-of-the-art methods utilize surround view images in Bird's Eye View (BEV) space, which is beneficial for various autonomous driving tasks, including lane detection, mapping, and sensor fusion. Recent advancements have led to two main approaches for Multi-view 3D object detection.

The first approach [1–3, 6–9, 20] employs explicit BEV feature-based methods. For instance, LSS [1] projects image features into cone-shaped volumes and converts

them into BEV grids using depth distributions. Bevddepth [8] enhances depth accuracy by integrating camera intrinsics for depth estimation. Bevstereo [6] improves depth estimates by comparing images over time and using disparity information. Bevformer [3] uses learnable queries to aggregate temporal and spatial features in BEV space, while Polarformer [21] applies polar coordinates to effectively handle radial and angular changes. DFA3D [22] introduces a 3D deformable attention mechanism to refine focus and improve depth representation quality.

The second approach [4, 5, 10–13, 23–25] features sparse query-based methods. DETR3D [10] extends the DETR [14] framework from 2D to 3D with sparse object queries, reducing computational costs but facing limitations in effectively linking 2D and 3D representations. PETR [5] addresses this issue by generating 3D spatial features from 2D images and using queries to interact directly with these features. PETRV2 [26] builds on PETR by enabling dynamic adjustments of 3D positional embeddings and incorporating temporal information for greater accuracy. StreamPETR [4] further refines predictions by leveraging historical frame queries. SparseBEV [11] uses adaptive scale self-attention and spatio-temporal sampling to better manage varying object sizes and temporal changes. MV2D [27] initializes queries with 2D detection results, enhancing 3D spatial information. CAPE [28] employs local camera view coordinates to generate stable and consistent 3D positional embedding, which addresses issues associated with global coordinate systems.

## 2.2 Position Embedding for 3D Object Detection

Transformers, introduced by Google in 2017, have become a cornerstone in natural language processing (NLP) and have demonstrated impressive performance in computer vision as well [25, 29–35]. Given the impressive performance of the Transform architecture in previous tasks, it has naturally been applied to the field of 3D vision, particularly in multi-view 3D object detection.

Detr3D [10] pioneers the application of the transform architecture to multi-view 3D tasks. However, it accomplishes this by projecting queries into image space, lacking a genuine 3D representation. PETR [5] builds on this by introducing 3D positional embedding, allowing image features to implicitly encode 3D information for better interaction with 3D queries. Subsequent research has increasingly focused on the importance of 3D positional embedding. 3DPPE [12] introduces a 3D point positional embedding by using a depth estimation network to estimate the image pixel depth, thereby embedding 3D positional information into the image and allowing it to be encoded consistently with 3D queries. OPEN [15] converts the pixel-level depth in 3DPPE into object-level depth information. VeDet [16] introduces a unified geometric representation that links images with 3D queries, using geometric relationships to represent positional relationships. CAPE [28] employs local camera view coordinates to generate stable and consistent 3D positional embedding. However, these methods primarily rely on absolute positional embedding; the embedded positional information is independent and lacks the explicit representation of relative positional relationships. Such relative position relationships are crucial for accurately capturing objects in 3D space, so we propose a 3D relative positional embedding method tailored for multi-view image 3D object detection.

### 3 Method

#### 3.1 Preliminary: Rotary Position Embedding

RoFormer [17] proposed a novel relative positional embedding method called Rotary Position Embedding (RoPE). Upon release, RoPE demonstrated strong performance in language modeling and showed great potential for further development. Previous research on relative positional embedding methods generally incorporates relative positional relationships as a bias added to the attention weights. Nevertheless, naively incorporating positional as a static bias may restrict its dynamic interplay with attention weights, potentially limiting the model's ability to fully exploit relative spatial relationships. A key limitation of prior approaches to relative positional encoding lies in their application as post hoc biases to the attention matrix following query-key interactions, thereby failing to directly influence the fundamental similarity computation between queries and keys. To resolve this limitation, RoPE innovatively introduces the multiplication of Euler's formula ( $e^{i\theta}$ ) to key and query vectors as relative positional embedding [36]. i.e., suppose (n, m-th) query and key is  $\mathbf{q}_n, \mathbf{k}_m \in \mathbb{R}^{1 \times d_{\text{head}}}$ , RoPE is applied as

$$\mathbf{q}_n' = \mathbf{q}_n e^{in\theta}, \mathbf{k}_m' = \mathbf{k}_m e^{im\theta} \quad (1)$$

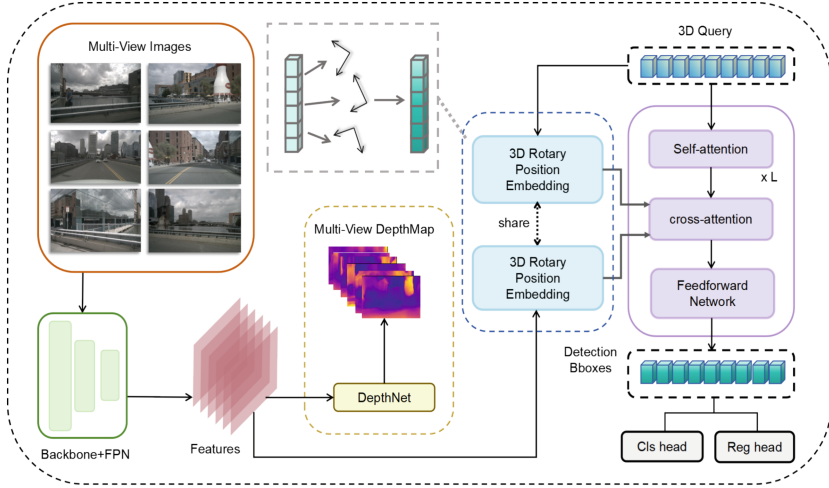
Then, (n-m)-th component of attention matrix is calculated as

$$\mathbf{A}_{(n,m)'} = \text{Re}[\mathbf{q}_n' \mathbf{k}_m'^*] = \text{Re}[\mathbf{q}_n \mathbf{k}_m^* e^{i(n-m)\theta}] \quad (2)$$

where  $\text{Re}[\cdot]$  denotes real part of complex number and  $*$  means complex conjugates. By multiplying complex rotation  $e^{in\theta}, e^{im\theta}$  depending on token position (n,m), RoPE injects the relative position (n-m) to the attention matrix in rotation form. Since the rotation of complex numbers corresponds to a two-dimensional plane rotation, in practical applications, the query and key should be paired by dimension with each dimension applying rotations according to its specific frequency. This method ingeniously integrates positional embedding and relative positional calculations into attention computations through rotation, eliminating the need for an additional bias term after the attention weights. It can be regarded as a relative positional embedding method implemented in the form of an absolute positional embedding method.

#### 3.2 Overall Architecture

As illustrated in the Fig. 1, the process begins with obtaining surround-view images  $\mathbf{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_n\}$ . These images are fed into a backbone network (ResNet [37], Swin Transformer [33]) and a FPN [38] neck, resulting in 2D features extracted from the images  $\mathbf{F} = \{\mathbf{F}_1, \dots, \mathbf{F}_n\}$ . Next, there is a depth estimation network that takes the 2D image features as input to generate a depth map of the 2D features  $\mathbf{D} = \{\mathbf{D}_1, \dots, \mathbf{D}_n\}$ . After that, we use the depth map  $\mathbf{D}$  and camera parameters to convert 2D pixel coordinates to 3D camera coordinates, then use the homogeneous transformation from the camera to the LiDAR coordinate system to obtain the 3D point of the image in the LiDAR coordinate system  $P^{3D} = \{P_i^{3D} \in \mathbb{R}^{3 \times H_{F_i} \times W_{F_i}}, i = 1, 2, \dots, N\}$  where  $H_{F_i}$  and



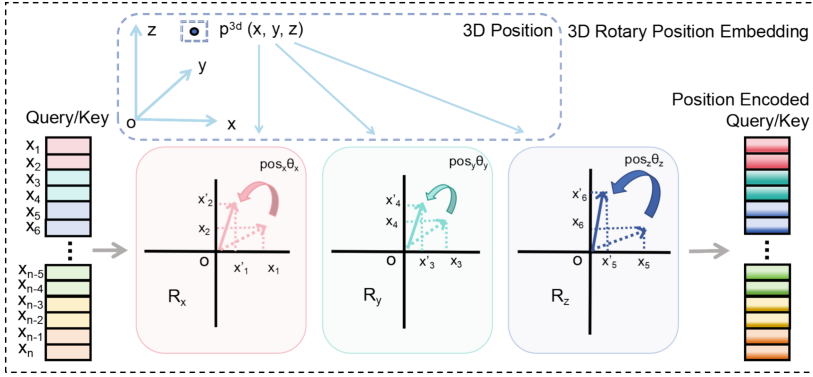
**Fig.1.** Our pipeline first extracts multi-view image features using a backbone and FPN. A depth estimation network predicts depth maps, which are combined with camera parameters to project image pixels into 3D space. The resulting 3D points and 3D queries are encoded with 3D Rotary Position Embedding. Finally, the 3D queries and image features interact via a decoder to generate 3D detection boxes.

$W_{F_i}$  are the  $i$ -th image shape. Then We use the obtained 3D points  $P^{3D}$  along with image features  $\mathbf{F}$  and 3D query along with its 3D points to generate their respective 3D rotary embedding via 3D Rotary Position Embedding (3DRoPE). Among 3DRoPE We introduce a learnable parameter for 3D positional information to automatically adapt to relative positions at different scales. Additionally we integrate the geometric information from multi-view cameras into the 3D positional embedding of each view's images to mitigate the impact of multi-view information on relative position calculations and enhance the model's sensitivity to multi-view information.

### 3.3 3D Rotary Position Embedding

Through our research on current sparse query-based methods we found that the positional information of the 3D query used for object localization is updated by the learned offsets from the decoder, these learned offsets represent the relative position between the 3D query and the object to some extent. But current methods typically use absolute positional embedding, where positional representations are independent of each other, requiring the model to infer spatial relationships from absolute positions to learn the offsets. So to address this issue, we design a 3D relative positional embedding method tailored to sparse query-based methods.

Drawing inspiration from RoPE used in large language models for relative position embedding, we propose 3DRoPE, which leverages the properties of RoPE (introduced in Sec. 3.1) to achieve 3D relative position embedding. As shown in the Fig. 2 we obtain the 3D point  $p^{3d} = (p^x, p^y, p^z)$  of the image feature and 3D query first. Then, we generate the respective complex rotation ( $e^{i\theta}$ ) based on the positional information along the x, y,



**Fig.2.** Implementation Details of 3D Rotary Position Embedding. 3D Rotary Position Embeddings are computed by deriving rotation angles from 3D positions along the x, y, and z axes. Query and key vectors are grouped in pairs and rotated accordingly to obtain position encoded representations.

and z axes.

$$\mathbf{R}_x = e^{i\theta_{tx}p^x}, \mathbf{R}_y = e^{i\theta_{ty}p^y}, \mathbf{R}_z = e^{i\theta_{tz}p^z} \quad (3)$$

where  $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z \in \mathbb{C}^{N \times (d_{\text{head}}/6)}$  and  $d_{\text{head}}$  denote channel dimensions of feature and query.  $\theta_t$  represents the frequencies used in the Sine and Cosine position embedding method, which can extend the 2D rotation represented by complex numbers to higher-dimensional rotations that correspond to the dimensions of the feature and query vectors.  $\theta_{tx}, \theta_{ty}, \theta_{tz} = 10000^{-t/(d_{\text{head}}/6)}$ ,  $t \in \{0, 1, \dots, d_{\text{head}}/6\}$ . Then we divide the image features and 3D query vectors along the channel dimension into three parts, respectively apply rotation that contains different position information. After rotating, we concatenate the three parts into a complete vector, thus completing the 3D rotation position embedding.

$$\mathbf{Q}_x = \{Q_1, Q_2, \dots, Q_{n-5}, Q_{n-4}\} \quad (4)$$

$$\mathbf{Q}_y = \{Q_3, Q_4, \dots, Q_{n-3}, Q_{n-2}\} \quad (5)$$

$$\mathbf{Q}_z = \{Q_5, Q_6, \dots, Q_{n-1}, Q_n\} \quad (6)$$

$$\mathbf{Q}_x' = \mathbf{Q}_x e^{i\theta_{tx}p^x}, \mathbf{Q}_y' = \mathbf{Q}_y e^{i\theta_{ty}p^y}, \mathbf{Q}_z' = \mathbf{Q}_z e^{i\theta_{tz}p^z} \quad (7)$$

$$\mathbf{Q}_{\text{ros}} = \text{cat}(\mathbf{Q}_x', \mathbf{Q}_y', \mathbf{Q}_z') \quad (8)$$

where  $\mathbf{Q}_x, \mathbf{Q}_y, \mathbf{Q}_z$  are the vectors that have been divided into three parts,  $\mathbf{Q}_x', \mathbf{Q}_y', \mathbf{Q}_z'$  are the vectors that have been rotated and contain their respective 3D positional information,  $\mathbf{Q}_{\text{ros}}$  is the vector formed by concatenating the three rotational embeddings together to form a complete 3D rotary position embedding of query and key. Consequently, using the rotary embedding for attention computation, the attention weight will encode 3D relative positional information similar to Eq. 2.

### 3.4 Adaptive Position Information

Through prior research [17, 36], which has demonstrated that rotary position embedding is capable of learning relative positions at various scales based on the rotation angle. Larger rotation angles are employed to capture global relationships, while smaller rotation angles are utilized for discerning subtle positional relationships. Consequently, we employ a learnable parameter for the 3d position information used as rotary embedding, enabling the model to autonomously adjust relative position relationships of different scales.

$$p_{\alpha}^{3d} = \alpha(p^x, p^y, p^z) \quad (9)$$

where  $p_{\alpha}^{3d}$  denotes the adaptive position and  $\alpha$  denotes the learnable parameter.

### 3.5 Multi-view Pose Enhanced Position Embedding

Since we use global attention similar to PETR [5] and normalize the 3D position during 3D rotary embedding, the relative position relationships will be influenced by multi-view information, making it difficult for the query to accurately distinguish objects from different views. To address this, we incorporate the geometric information of multi-view cameras into the positional embedding of each image feature to help mitigate this effect and enhance the model’s ability to understand multi-view information. We use the intrinsic  $\mathbf{I}_n$  and extrinsic  $\mathbf{K}_n$  for each multi-view camera to compute the position and orientation of each camera as  $[\bar{\mathbf{q}}_n, \mathbf{t}_n]$  where  $\bar{\mathbf{q}}_n$  denotes the quaternion vector and  $\mathbf{t}_n$  denotes the translation of the camera pose. Inspired by Nerf [39], we first use a Fourier transform to capture the fine-grained changes in the geometric attributes.

$$\gamma(x|[f_1, \dots, f_k]) = [\sin(f_1 \pi x), \cos(f_1 \pi x), \dots] \quad (10)$$

The  $k$  frequencies  $[f_1, \dots, f_k]$  are sampled evenly between  $[0, f_{\max}]$  and  $x$  to denote the characteristics of the multi-view geometric attributes. Then, we employ an MLP to project the output to the dimensions of image features as the geometric information embedding for multi-view images as  $\mathbf{G}_n^e \in \mathbb{R}^{C \times H \times W}$ .

$$\mathbf{G}_n^e = \text{MLP}_{\text{enc}}(\gamma[\bar{\mathbf{q}}_n, \mathbf{t}_n]) \quad (11)$$

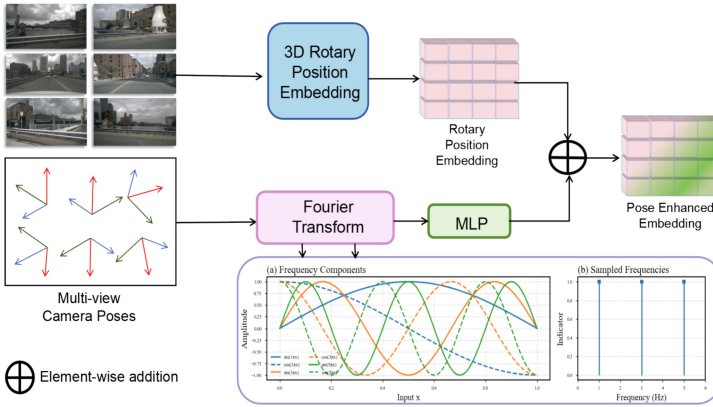
As shown in Fig. 3, we add the geometric embedding to the position embedding to form the complete multi-view pose-enhanced rotary position embedding.

## 4 Experiment

### 4.1 Dataset and Metrics

Experiments are conducted on the nuScenes dataset [18], a large-scale benchmark for autonomous driving with 1,000 driving scenes from diverse urban environments. It provides multi-sensor data (6 cameras, 1 LiDAR, 5 RADARs) and high-quality 3D annotations for 10 object categories at 2 Hz. The dataset is split into 700 training, 150 validation,





**Fig.3.** Multi-view images are processed to obtain 3D rotary position embeddings via feature extraction and depth estimation. Camera pose information is Fourier-transformed and projected by an MLP, then added to the rotary embeddings to produce pose-enhanced position embeddings.

and 150 test scenes. Evaluations include comparisons and ablation studies using standard metrics: nuScenes Detection Score (NDS), Mean Average Precision(mAP), Mean Average Translation Error(mATE), Mean Average Scale Error(mASE), Mean Average Orientation Error(mAOE), Mean Average Velocity Error(mAVE), and Mean Average Attribute Error(mAAE).

## 4.2 Main Results of the Comparative Experiments

We compare our method with other state-of-the-art methods on the nuScenes dataset. All compared methods, including ours, follow the single-frame paradigm and do not utilize temporal information. The P4 feature is leveraged by default and all methods do not use the test-time augmentation.

Table 1 shows the results on the nuScenes dataset validation set, using ResNet-50 and VoVNet-99 [40] as backbones, respectively. Most comparisons are conducted using ResNet-50, including methods based on sparse query as well as dense BEV feature; our method outperforms these methods in both mAP and NDS, the two primary metrics. Notably, for methods about 3D position embedding, such as CAPE, PETR, and 3DPPE, our method consistently outperforms them to varying degrees. With the larger VoVNet-99 backbone, we compare our method only against two methods using 3D absolute position embedding: PETR and 3DPPE. The comparison across these two different backbones robustly demonstrates that our proposed relative position embedding method surpasses the previous absolute embedding methods. As shown in Table 2, we further validate the effectiveness of our method on the test set, using VoVNet-99 as the backbone. Our method, 3DRoPE, continues to outperform various previous methods, including PETR and 3DPPE; this further underscores the reliability and accuracy of our approach.

**Table 1.** Comparison of 3D object detection on the nuScenes val split. "S" denotes model with a single time stamp input. \* is trained with CBGS. CBGS is a data augmentation method which will elongate 1 epoch into 4.5 epochs. † indicates using the pre-trainedFOCS3D backbone for model initialization.

Methods	Backbone	Resolution	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
BEVDet	Res-50	$784 \times 256$	0.298	0.379	0.725	0.279	0.559	0.860	0.245
BEVDepth-S	Res-50	$784 \times 256$	0.315	0.367	0.702	<b>0.271</b>	0.621	1.042	0.315
DETR3D	Res-50	$1408 \times 512$	0.303	0.374	0.860	0.278	0.637	0.967	0.235
CAPE	Res-50	$1408 \times 512$	0.337	0.380	0.778	0.280	0.568	0.963	0.224
PETR *	Res-50	$1408 \times 512$	0.339	0.403	0.748	0.273	0.539	0.907	0.203
3DPPE *	Res-50	$1408 \times 512$	0.370	0.433	0.689	0.279	<b>0.524</b>	0.828	0.202
<b>3DRoPE *</b>	Res-50	$1408 \times 512$	<b>0.380</b>	<b>0.440</b>	<b>0.682</b>	0.272	0.534	<b>0.815</b>	<b>0.197</b>
PETR†	VoV-99	$800 \times 320$	0.378	0.426	0.746	0.272	0.488	0.906	0.212
3DPPE†	VoV-99	$800 \times 320$	0.398	0.446	0.704	0.270	0.495	<b>0.843</b>	0.218
<b>3DRoPE†</b>	VoV-99	$800 \times 320$	<b>0.399</b>	<b>0.450</b>	<b>0.670</b>	<b>0.263</b>	<b>0.483</b>	0.849	<b>0.211</b>

**Table 2.** Comparison of 3D object detection on the nuScenes test split. "S" denotes model with a single time stamp input. All methods are trained with CBGS and use the DD3D pre-trained model weights for model initialization. CBGS is a data augmentation method which will elongate 1 epoch into 4.5 epochs.

Methods	Backbone	Resolution	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
DETR3D	VoV-99	$1600 \times 640$	0.412	0.479	0.641	0.255	0.394	0.845	0.133
DD3D	VoV-99	$1600 \times 640$	0.418	0.477	0.572	0.249	<b>0.368</b>	1.014	<b>0.124</b>
BEVDet	VoV-99	$1600 \times 640$	0.424	0.488	0.524	<b>0.242</b>	0.373	0.950	0.148
BEVFormer-S	VoV-99	$1600 \times 640$	0.435	0.495	0.589	0.254	0.402	0.842	0.131
PETR	VoV-99	$1600 \times 640$	0.441	0.504	0.593	0.249	0.383	0.808	0.132
CAPE	VoV-99	$1600 \times 640$	0.458	0.520	0.561	0.252	0.389	<b>0.758</b>	0.132
3DPPE	VoV-99	$1600 \times 640$	0.460	0.514	0.569	0.255	0.394	0.796	0.138
<b>3DRoPE</b>	VoV-99	$1600 \times 640$	<b>0.473</b>	<b>0.529</b>	<b>0.544</b>	0.246	0.384	0.779	0.126

**Table 3.** Ablation experiments of our method on nuScenes val split. PETR is employed as the baseline, and we add the 3DRoPE, Adaptive Position and multi-view geometric in order. Here we exploit VoVNet-99 as the backbone and set the input resolution as  $800 \times 320$ .

#	3DRoPE	Adaptive	Pose	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
1				0.378	0.426	0.746	0.272	0.488	0.906	0.212
2	✓			0.403	0.458	0.648	0.256	0.461	0.881	0.174
3	✓	✓		0.411	0.469	0.599	0.251	0.397	0.816	0.147
4	✓		✓	0.409	0.465	0.604	0.246	0.386	0.799	0.151
5	✓	✓	✓	<b>0.414</b>	<b>0.471</b>	<b>0.572</b>	<b>0.242</b>	<b>0.374</b>	<b>0.788</b>	<b>0.136</b>

**Table 4.** Ablation experiments with different frequencies  $\theta$  and  $\alpha$  for 3DRoPE. L denotes whether the L is a learnable parameter or not. Here we exploit Res-50 as the backbone and set the input resolution as  $704 \times 256$ .

$\theta$	$\alpha$	L	mAP $\uparrow$	NDS $\uparrow$	mAOE $\downarrow$
100	1		0.282	0.330	0.712
100	10		0.282	0.342	0.697
1000	1		0.283	0.336	0.704
1000	10		0.285	0.337	0.686
10000	1		0.287	0.339	0.678
10000	10		0.289	0.346	0.630
<b>10000</b>	<b>10</b>	$\checkmark$	<b>0.293</b>	<b>0.350</b>	<b>0.621</b>

### 4.3 Ablation Study & Analysis

In this section, we present ablation experiments and a comprehensive analysis of the essential components of our model. We use PETR as the baseline and Table 3 shows that we start from PETR and add each module in order.

**3DRoPE.** The comparison between #1 and #2 shows that replacing PETR’s original ray embedding with 3DRoPE results in an improvement of 1.2% mAP and 1.4% NDS; this demonstrates that our basic 3DRoPE also provides a noticeable improvement. Next, as shown in #3, we introduce a learnable parameter for 3D position, which further brings an improvement of 0.7% mAP and 0.7% NDS. In Table 4, to identify the optimal combination of frequency  $\theta$  and learnable parameter  $\alpha$ , we conduct ablation studies across various combinations, testing three different frequencies  $\theta$  and two different parameters  $\alpha$ . The experimental results show that optimal performance is achieved when  $\theta$  is set to 10000 and  $\alpha$  is set to 10. Furthermore, when  $\alpha$  is a learnable parameter, the model attains the best results. This highlights the effectiveness and importance of introducing a learnable parameter  $\alpha$  for positional information, as it enables the model to capture richer and more accurate relative positional relationships, thereby enhancing object localization.

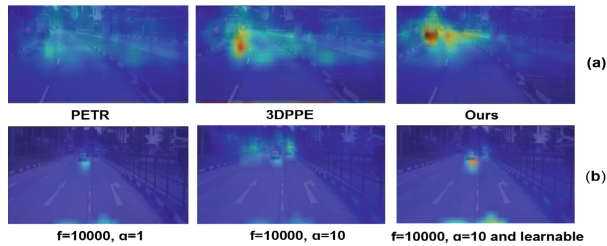
**Table 5.** Different encoding methods of multi-view geometric embedding. Here we exploit Res-50 as the backbone and set the input resolution as  $704 \times 256$ .

Encoding Method	mAP $\uparrow$	NDS $\uparrow$	mAOE $\downarrow$
SineCosie	0.285	0.332	0.704
<b>Nerf(Ours)</b>	0.282	<b>0.341</b>	<b>0.673</b>

**Pose-enhanced Embedding.** The comparison between #2 and #4 presents that Pose-enhanced Embedding, compared to the basic 3DRoPE, results in an improvement of

**Table 6.** Experiments of different temporal fusion methods. Here we exploit VoVNet-99 as the backbone and set the input resolution as  $800 \times 320$ .

Temporal Method	mAP $\uparrow$	NDS $\uparrow$	mATE $\downarrow$
PETRv2	0.410	0.503	0.722
<b>PETRv2 + 3DRoPE</b>	<b>0.442</b>	<b>0.529</b>	<b>0.695</b>
StreamPETR	0.482	0.571	0.598
<b>StreamPETR + 3DRoPE</b>	<b>0.501</b>	<b>0.588</b>	<b>0.488</b>

**Fig. 4.** The visualization of attention maps for different methods in single-view. (a) represents the comparison of attention maps between our method, PETR and 3DPPE. (b) represents the visual results of different combinations from our experiments in Table 4.

0.4% mAP and 0.4% NDS. It confirms the effectiveness of integrating multi-view camera geometric information into 3DRoPE, as it enhances model performance by improving the model’s understanding of multi-view information and reducing the impact on relative positional calculations. In Table 5, we compare the commonly used Sine-Cosine encoding method with the Nerf-based encoding method for embedding geometric information. The results indicate that the Nerf-based method outperforms the sine-cosine method in geometry-related metrics: NDS and mAOE, demonstrating the compatibility of the Nerf-based approach with our method and its superior ability to capture geometric information.

Figure 4 presents the visualization results of attention maps of different methods. From part (a), it can be observed that our method provides more accurate object localization compared to PETR and 3DPPE, including for some distant objects. Part (b) visualizes the experimental results of the last three combinations from Table 4, it can be seen that the last combination achieves the best object localization result.

#### 4.4 Experiments on Temporal Fusion

Currently, most sparse query-based methods improve the accuracy of 3D object detection by incorporating temporal fusion mechanisms. Taking the baseline model PETR as an example, its subsequent versions—PETRv2 [26] and StreamPETR [4]—both adopt different temporal fusion strategies. To verify the effectiveness of our proposed method when extended to temporal fusion tasks, we conducted relevant experiments by replacing

the original 3D absolute positional embedding in PETRv2 and StreamPETR with our designed 3DRoPE. The Table 6 demonstrate that after applying 3DRoPE, the detection performance improves to varying degrees.

This experiment also highlights the importance of temporal fusion for the future development of sparse query-based 3D object detection methods. Designing an effective temporal fusion approach can partially compensate for the accuracy limitations of pure vision-based methods.

**Table 7.** Results of using the ground-truth depth for generating 3D position. Here we exploit Res-50 as the backbone and set the input resolution as  $1408 \times 512$ .

Ground-truth Depth	mAP $\uparrow$	NDS $\uparrow$	mAOE $\downarrow$
3DPPE	0.400	0.420	0.690
<b>3DRoPE(Ours)</b>	<b>0.454</b>	<b>0.467</b>	<b>0.598</b>

#### 4.5 Limitations and Further Improvements

While our experiments and theoretical analysis demonstrate the advantages and improvements of our proposed method compared to previous approaches, no method is without its limitations. As shown in Table 7, when using ground-truth depth, our method outperforms the 3DPPE by 5.4% in mAP and 4.7% in NDS. However, in previous experiments using a depth estimation network, the improvement was less pronounced. This highlights the inherent superiority of our 3DRoPE but also reveals that performance is significantly affected when depth information is less accurate. Thus, future research should focus on reducing dependency on depth accuracy, as there is substantial room for improvement in this area. Moreover, many current methods incorporate temporal information, a direction not explored in this paper. So future research should focus on the fusion of information across different frames and the study of relative positional relationships between frames. These aspects are worth further investigation in the future.

## 5 Conclusion

In this paper, we present a 3D relative position embedding method based on RoPE for multi-view 3D object detection. By embedding positional information into queries and keys through rotation, our approach ensures that relative positional information is explicitly reflected in the attention weights during calculations. We also introduce a learnable parameter for 3D positional information, allowing the model to use 3DRoPE to capture relative positional relationships across different scales. To mitigate the impact of multi-view information on relative positional calculations, we incorporate the geometric information of multi-view cameras into 3D positional embedding of the images. Extensive experiments on the nuScenes dataset demonstrate the effectiveness of our method,

showing superior object localization performance compared to previous absolute position embedding methods. Finally, we discuss the limitations and further improvements of our approach, specifically its reliance on accurate depth information and the absence of temporal information fusion.

**Acknowledgments.** This study was supported by the National Natural Science Foundation of China (62376252); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003); Zhejiang Province Leading Geese Plan(2025C02025,2025C01056); Zhejiang Province Province-Land Synergy Program(2025SDXT004–3).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. 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, Cham (2020). [https://doi.org/10.1007/978-3-030-58568-6\\_12](https://doi.org/10.1007/978-3-030-58568-6_12)
2. 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)
3. Li, Z., et al.: BEVFormer: learning Bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In: European Conference on Computer Vision, pp. 1–18. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-20077-9\\_1](https://doi.org/10.1007/978-3-031-20077-9_1)
4. 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, pp. 3621–3631 (2023)
5. 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, Cham (2022). [https://doi.org/10.1007/978-3-031-19812-0\\_31](https://doi.org/10.1007/978-3-031-19812-0_31)
6. 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. vol. 37, pp. 1486–1494 (2023)
7. Liu, Z., et al.: BEVFusion: multi-task multi-sensor fusion with unified bird’s-eye view representation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 2774–2781. IEEE (2023)
8. Li, Y., et al.: BEVDepth: acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 1477–1485 (2023)
9. Yang, C., 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)
10. 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)

11. 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)
12. 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)
13. Jiang, X., et al.: Far3D: expanding the horizon for surround-view 3D object detection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 2561–2569 (2024)
14. 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, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13)
15. Hou, J., et al.: OPEN: object-wise position embedding for multi-view 3D object detection. *ArXiv preprint arXiv:2407.10753* (2024)
16. Chen, D., Li, J., Guizilini, V., Ambrus, R.A., Gaidon, A.: Viewpoint equivariance for multi-view 3D object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9213–9222 (2023)
17. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: RoFormer: enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024)
18. Caesar, H., et al: nuScenes: a multimodal dataset for autonomous driving. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631 (2020)
19. 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)
20. Qi, Z., Wang, J., Wu, X., Zhao, H.: OCBEV: object-centric BEV transformer for multi-view 3D object detection. In: *2024 International Conference on 3D Vision (3DV)*, pp. 1188–1197. IEEE (2024)
21. Jiang, Y., et al.: PolarFormer: multi-camera 3D object detection with polar transformer. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1042–1050 (2023)
22. Li, H., et al: 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)
23. Wang, S., Jiang, X., Li, Y.: Focal-PETR: embracing foreground for efficient multi camera 3D object detection. *IEEE Trans. Intell. Veh.* (2023)
24. Yang, Z., Yu, Z., Choy, C., Wang, R., Anandkumar, A., Alvarez, J.M.: Improving distant 3D object detection using 2D box supervision. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14853–14863 (2024)
25. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10033–10041 (2021)
26. Liu, Y., et al.: PETRv2: a unified framework for 3d perception from multi-camera images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3262–3272 (2023)
27. Wang, Z., Huang, Z., Fu, J., Wang, N., Liu, S.: Object as query: lifting any 2D object detector to 3D detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3791–3800 (2023)
28. Xiong, K., et al.: CAPE: camera view position embedding for multi-view 3d object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21570–21579 (2023)
29. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)

30. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. arXiv preprint [arXiv:1803.02155](https://arxiv.org/abs/1803.02155) (2018)
31. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
32. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
33. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
34. Chu, X., Tian, Z., Zhang, B., Wang, X., Shen, C.: Conditional positional encodings for vision transformers. arXiv preprint [arXiv:2102.10882](https://arxiv.org/abs/2102.10882) (2021)
35. Touvron, H., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint [arXiv:2307.09288](https://arxiv.org/abs/2307.09288) (2023)
36. Heo, B., Park, S., Han, D., Yun, S.: Rotary position embedding for vision transformer. arXiv preprint [arXiv:2403.13298](https://arxiv.org/abs/2403.13298) (2024)
37. 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)
38. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
39. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 99–106 (2021)
40. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 752–760 (2019)