URoPE: Universal Relative Position Embedding across Geometric Spaces

Yichen Xie[1,2], Depu Meng[1], Chensheng Peng[1,2], Yihan Hu[1], Quentin Herau[1], Masayoshi Tomizuka[2], and Wei Zhan[1,2]

[1] Applied Intuition
[2] University of California, Berkeley

Abstract. Relative position embedding has become a standard mechanism for encoding positional information in Transformers. However, existing formulations are typically limited to a fixed geometric space, namely 1D sequences or regular 2D/3D grids, which restricts their applicability to many computer vision tasks that require geometric reasoning across camera views or between 2D and 3D spaces. To address this limitation, we propose URoPE, a universal extension of Rotary Position Embedding (RoPE) to cross-view or cross-dimensional geometric spaces. For each key/value image patch, URoPE samples 3D points along the corresponding camera ray at predefined depth anchors and projects them into the query image plane. Standard 2D RoPE can then be applied using the projected pixel coordinates. URoPE is a parameter-free and intrinsics-aware relative position embedding that is invariant to the choice of global coordinate systems, while remaining fully compatible with existing RoPE-optimized attention kernels. We evaluate URoPE as a plug-in positional encoding for transformer architectures across a diverse set of tasks, including novel view synthesis, 3D object detection, object tracking, and depth estimation, covering 2D–2D, 2D–3D, and temporal scenarios. Experiments show that URoPE consistently improves the performance of transformer-based models across all tasks, demonstrating its effectiveness and generality for geometric reasoning.

Keywords: Relative Position Embedding · Projective Geometry · Multi-view Vision

1 Introduction

Transformers [31] have become the dominant architecture in the field of computer vision including multi-view perception and generation tasks, from novel view synthesis [13, 23] and stereo matching [14, 41], to 2D/3D object detection [2, 19, 20, 34]. A critical challenge in applying Transformers to geometric tasks is how to encode the spatial relationships between tokens that originate from different viewpoints, coordinate systems, or even different geometric modalities (2D images and 3D points).

As a standard solution, position embeddings inject positional information into the permutation invariant Transformer architecture. Compared with absolute position embeddings [31], relative formulations [3, 22, 24, 27] offer improved

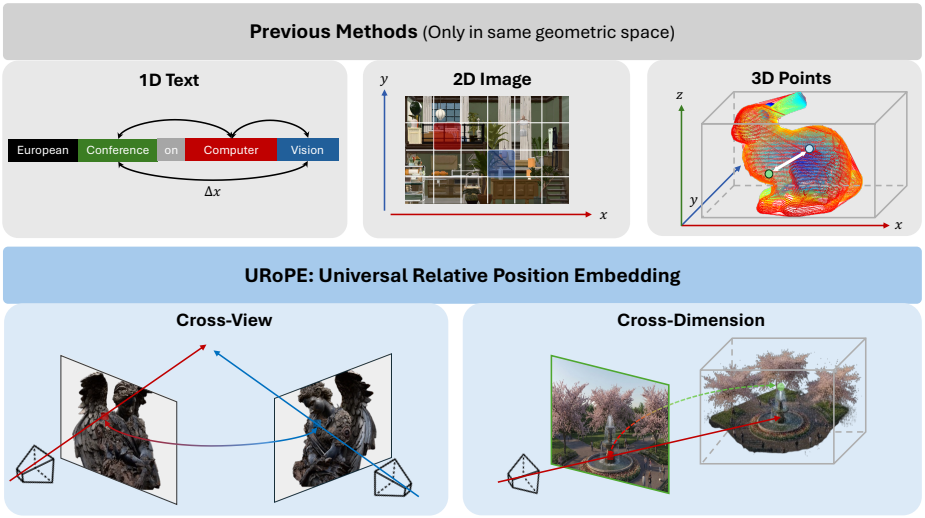


Fig. 1: URoPE for Relative Position Embedding across Geometric Spaces. Previous relative position embeddings can only handle a shared geometric space (top), while URoPE focuses on the relative position embedding across geometric spaces (bottom).

generalization and length extrapolation, making them a mainstream choice in modern Transformers, especially for geometric tasks. A particularly important development is Rotary Position Embedding (RoPE) [27], which encodes *relative* rather than absolute positions through rotation matrices applied to query/key pairs. Standard RoPE operates within a single flat coordinate space, assigning positions by sequence indices (1D) or image grid locations (2D), which is fundamentally inadequate for cross-view geometric reasoning: *pixels from two camera views may be close in 3D space yet far apart in their respective 2D grids.*

Recent efforts have begun to incorporate camera geometry into attention, but often rely on matrix multiplication rather than extending RoPE itself to cross-view settings. Plücker ray embeddings [25] concatenate ray origin and direction as input features. This is an absolute encoding that lacks the relative bias properties crucial for attention generalization, while its relative variant [38] still relies on the definition of a global coordinate system. Recent work [15, 21] combines camera parameters and standard 2D RoPE in a channel-wise split. However, the inter-camera geometry and intra-image spatial position are encoded disjointly with no interactions, and the inter-camera geometry is only considered for each camera rather than the local patch. Overall, it remains an open question to model the relative position across camera views and wider geometric spaces.

We observe that the fundamental question in cross-view relative position is: *where does a key token's 3D content appear in the query token's image?* Instead of encoding abstract ray coordinates, URoPE uses explicit projective geometry to express cross-view correspondences *directly in the query image plane*, so the relative position can be modeled in a single shared coordinate system.

Table 1: Comparison of Cross-View Position Embedding. **Mechanism:** how geometry modulates attention: concatenation with input tokens (Concat.), matrix multiplication on Q/K/V features (MatMul), sinusoidal rotation on Q/K (RoPE), or a combination. **Per-patch Geo.:** whether the camera geometry is encoded at the individual patch level. **SE(3) Inv.:** invariance to rigid transformations of the global coordinate system. **Param-Free:** no learnable parameters introduced.

Method	Mechanism	Geometric Info	Per-patch Geo.	SE(3) Inv.	Param-Free
Plücker [25]	Concat.	6D Ray	✓	✗	✓
Relative Ray [38]	RoPE	6D Ray	✓	✗	✓
GTA [21]	MatMul	Extrinsics	✗	✓	✓
PRoPE [15]	RoPE + MatMul	Proj. matrix + grid	✗	✓	✓
RayRoPE [36]	RoPE + Linear	Ray + learned depth	✓	✓	✗
URoPE	RoPE	Proj. coords	✓	✓	✓

Specifically, for each key token from a source view, URoPE starts from its camera ray and samples 3D points along the ray at a set of fixed depth anchors. Each sampled 3D point is then projected into the query camera using the relative camera transform and the query intrinsics, yielding a depth-conditioned pixel coordinate in the query image plane. Finally, we apply standard 2D RoPE between the query location and the projected key location, producing a geometry-aware *relative* positional bias that is consistent across camera views.

A central challenge is that cross-view projection is inherently depth ambiguous: a source pixel corresponds to an epipolar line in the query view. We address this with **depth-anchored multi-head attention**: different attention heads (or head groups) are assigned with different fixed depth anchors, so each head encodes one depth hypothesis and multi-head attention jointly covers near- to far-field correspondences. In each head, we split the per-head channels across the horizontal and vertical axes ($d_h/2$ for u and $d_h/2$ for v) and apply standard 2D RoPE in the query image plane, keeping URoPE natively compatible with FlashAttention [4] and other RoPE-optimized kernels.

A key contribution of URoPE is its universality. We demonstrate that the same projective RoPE formulation without task-specific modifications, serves as a plug-in position encoding across diverse geometric tasks:

- **Novel view synthesis**: 2D→2D cross-view attention with known camera poses on Objaverse [5] and RealEstate10k [43].
- **3D object detection and tracking**: 2D→3D attention between image features and 3D query positions on nuScenes [1].
- **Stereo depth estimation**: 2D→2D cross-view matching for depth prediction on RGBD [26], Scenes11 [30], and SUN3D [37].

We show consistent improvements across all benchmarks, establishing URoPE as a general-purpose geometric position encoding for Transformers.

2 Related Work

Position Encoding in Transformers. Since sinusoidal positional encodings are introduced for sequence modeling [31], position embeddings have become standard in transformers [2, 6] for multiple tasks. As an important milestone, Rotary Position Embedding (RoPE) [27] encodes relative positions through rotation matrices applied to query and key vectors, enabling relative position bias without explicit bias terms. RoPE has become the default position encoding in modern large language models [8, 29], which is also extended to 2D for vision and multi-modal tasks [11,33]. Our work extends RoPE to cross-view and cross-dimensional geometric spaces by utilizing the explicit projection to convert the key tokens to the same geometric space of query tokens.

Transformers across Geometric Spaces. Transformers are widely applied to computer vision tasks across geometric spaces including novel view synthesis [12,13], 3D scene understanding [19,35,40], and stereo depth estimation [41]. To bridge the geometric gap, some method adopts explicit projection. For example, epipolar attention [10,39] restricts attention to the projected epipolar lines but can not capture off-epipolar-line correspondences. DETR3D [35] projects 3D queries to multiview image for feature sampling. BEVFormer [16] applies deformable attention [44] with 3D reference points. However, these complex attention masks or kernels are unfriendly with efficient implementations like FlashAttention [4]. In contrast, URoPE compute the RoPE positions via projective geometry, natually encoding epipolar constraints through the soft attention bias. By operating entirely through RoPE rather than matrix multiplications, URoPE unified inter-camera and geometry and intra-image spatial position into a single mechanism while keeping the original Q/K/V multiplication format of standard attention operation.

Multiview Position Embedding. Incorporating camera geometry into multiview attention can be broadly categorized by mechanism. At the input level, Plücker ray embeddings [25] and camera ray maps [7] concatenate ray information with image features, while PETR [19] and StreamPETR [34] encodes 3D positions into image features via position-aware embeddings. These are absolute encodings that lack the relative position bias property. RAYNOVA [38] takes a step further to encode the relative position in the ray space, but it still relies on the global coordinate system. GTA [21] applies camera extrinsic matrices to Q/K/V features, achieving SE(3)-invariant. PRoPE [15] extends GTA by incorporating intrinsics via normalized projection matrices and combining with 2D RoPE for each patch positions. However, it decouples inter-camera and intra-camera geometry across separate head dimension blocks. RayRoPE [36], concurrent with our work, represents each patch as a ray segment with a layer-wise parametric module to predict the depth, but the unsupervised module offers no guarantee to estimate the actual scene depth. In contrast, URoPE assigns fixed depth anchors across head groups, providing explicit multi-depth coverage without learnable parameters.

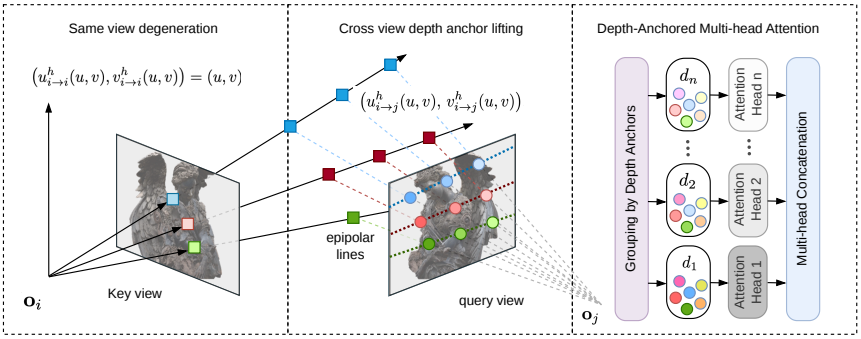


Fig. 2: Overview of URoPE. **Left**: in the same-view case, URoPE reduces to standard 2D RoPE. **Middle**: for cross-view interaction, a key-view pixel is lifted along its camera ray to multiple depth anchors and projected into the query view, producing head-specific positions along epipolar lines. **Right**: projected positions are grouped by depth anchors and assigned to different attention heads, enabling depth-anchored multi-head attention.

3 Method

We introduce URoPE, a relative position embedding for cross-view and cross-dimensional geometric reasoning. Our key insight is to use projective geometry to express cross-view correspondences directly in the query image plane. We first formalize camera rays as a bridge between 2D image coordinates and 3D space (Sec. 3.1), then derive depth-anchored lifting and cross-view projection (Sec. 3.2), and finally integrate the resulting projected coordinates into RoPE-based attention (Sec. 3.3). The overview of our method is visualized in Fig. 2.

3.1 Camera Rays as a Bridge Between 2D and 3D

Each image pixel corresponds to a light ray passing through the camera center. Camera rays therefore provide a natural bridge between the 2D image plane and 3D space. Given an image location (u, v) in camera view i with intrinsics $\mathbf{K}_i$ and extrinsics $[\mathbf{R}_i, \mathbf{t}_i]$, we represent its camera ray in the world coordinate system as

$$\mathbf{ray}_i(u,v) = (\mathbf{o}_i, \mathbf{r}_i(u,v)), \quad \mathbf{r}_i(u,v) = \mathbf{R}_i^T \mathbf{K}_i^{-1} [u, v, 1]^T, \tag{1}$$

where $\mathbf{o}_i \in \mathbb{R}^3$ is the camera center in world coordinates, defined as $\mathbf{o}_i = -\mathbf{R}_i^T \mathbf{t}_i$, and $\mathbf{r}_i(u,v) \in \mathbb{R}^3$ is the corresponding ray direction.

The camera rays work as a natural absolute position embedding [13] for attention modules across views. We also consider a direct ray-level relative position embedding baseline similar to [38] that applies independent 1D RoPE to the ray components $\mathbf{ray}_i(u,v) = (\mathbf{o}_i, \mathbf{r}_i(u,v)) \in \mathbb{R}^6$ by splitting the per-head channels into six blocks (6D RoPE in Tab. 2).

However, the $\mathbf{ray}_i(u, v)$ representation can hardly model the potential intersection or spatial relationship between camera rays in the 3D space. Cross-view reasoning ultimately requires understanding where content observed along a ray in one camera would appear in another camera. This motivates lifting points along the ray and explicitly projecting them into the query image plane.

3.2 Bridging Geometric Gaps with Lifting and Projection

To capture cross-view correspondences implied by camera rays, we consider 3D points along each ray. In the ideal case, a 3D point $\mathbf{p}_i(u, v) \in \mathbb{R}^3$ can be recovered from the ray given its depth. However, true depth is typically unavailable, and predicting depth inside every attention layer is undesirable.

Depth-anchored lifting. We therefore introduce a set of fixed depth anchors $\mathcal{D} = \{d^h\}_{h=1}^K$ and lift each pixel (u, v) into a set of 3D points

$$\mathbf{p}_i^h(u, v) = \mathbf{o}_i + d^h \cdot \mathbf{r}_i(u, v), \tag{2}$$

forming $\mathcal{P}_i(u, v) = \{\mathbf{p}_i^h(u, v)\}_{h=1}^K$.

Projection across camera views. Given a source view i and a query view j, we project each lifted point $\mathbf{p}_i^h(u, v)$ into the image plane of camera j:

$$\tilde{\mathbf{u}}_{i\to j}^h(u, v) = \mathbf{K}_j \left(\mathbf{R}_j \mathbf{p}_i^h(u, v) + \mathbf{t}_j\right) = \begin{bmatrix} \tilde{u}_{i\to j}^h(u, v) \\ \tilde{v}_{i\to j}^h(u, v) \\ \tilde{w}_{i\to j}^h(u, v) \end{bmatrix}, \tag{3}$$

$$u_{i\to j}^h(u, v) = \tilde{u}_{i\to j}^h(u, v) \,/\, \tilde{w}_{i\to j}^h(u, v), \qquad v_{i\to j}^h(u, v) = \tilde{v}_{i\to j}^h(u, v) \,/\, \tilde{w}_{i\to j}^h(u, v), \tag{4}$$

yielding a set of projected pixel coordinates $\{(u_{i\to j}^h(u, v), v_{i\to j}^h(u, v))\}_{h=1}^K$ in the query image plane. These projected points lie along the epipolar line induced by the source pixel, and provide an explicit, intrinsics-aware mapping from the source view to the query view.

Degeneration to the single-view case. When $i = j$, projection reduces to the identity mapping, so the projected coordinates coincide with original locations:

$$\left(u_{i\to i}^h(u, v), v_{i\to i}^h(u, v)\right) = (u, v), \quad \forall h = 1, \ldots, K. \tag{5}$$

Thus, URoPE naturally degenerates to standard 2D RoPE in a single image.

Extension to 2D–3D interaction. URoPE also extends to 2D–3D interactions by skipping the image-plane projection. Given a 3D token at (x, y, z), we measure relative positions between (x, y, z) and the lifted points $\mathcal{P}_i(u, v)$ in 3D, enabling cross-attention between image features and 3D queries.

3.3 Depth-Anchored Multi-head Attention

URoPE for relative position embedding. We integrate the projected geometry derived in Sec. 3.2 into Transformer attention. For self- or cross-attention

across camera views, we consider a key/value image patch (u, v) from camera view i and a query image patch (u', v') from camera view j. Let $h \in \{1, \ldots, K\}$ denote the attention head index, and associate each head with a fixed depth anchor $d^h \in \mathcal{D}$. For head h, we lift the key/value patch using Eq. 2 and project it into the query view using Eq. 4, yielding

$$\big(u^h_{i\to j}(u, v),\, v^h_{i\to j}(u, v)\big). \tag{6}$$

Image-plane RoPE. We apply RoPE [27] in the query image plane between the query location (u', v') and the projected key location $\big(u^h_{i\to j}(u, v), v^h_{i\to j}(u, v)\big)$. Concretely, we apply standard 1D RoPE independently along the x- and y-axes on the first half of the per-head channels:

$$\begin{aligned} \mathbf{R}^{\text{query}} &= \text{diag}\Big(\text{RoPE}_{\frac{C}{2}}(u'),\ \text{RoPE}_{\frac{C}{2}}(v')\Big), \\ \mathbf{R}^{\text{key}}(h) &= \text{diag}\Big(\text{RoPE}_{\frac{C}{2}}\big(u^h_{i\to j}(u, v)\big),\ \text{RoPE}_{\frac{C}{2}}\big(v^h_{i\to j}(u, v)\big)\Big), \end{aligned} \tag{7}$$

where $\text{RoPE}_{\frac{C}{2}}(\cdot) \in \mathbb{R}^{\frac{C}{2}\times\frac{C}{2}}$ is the 1D rotary position embedding [27] and C is the per-head dimension. Finally, we apply $\mathbf{R}^{\text{query}}$ to the query features and $\mathbf{R}^{\text{key}}(h)$ to each head of key features. This keeps URoPE fully compatible with RoPE-optimized attention kernels.

Applying URoPE to multiview features. Although Eq. 7 resembles standard RoPE [27], it is not directly applicable when query tokens come from multiple camera views within the same sequence, since $\mathbf{R}^{\text{query}}_{\text{loc}}$ depends on the query view. Without loss of generality, we describe the self-attention case. Let $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B\times L\times H\times C}$ be the query/key/value tensors spanning N views, where $L = N \cdot L_v$ and L_v is the number of patches per view.

To ensure that all queries within one attention call share the same view, we reshape the queries by moving the view dimension from the sequence length into the batch dimension:

$$\tilde{\mathbf{Q}} \in \mathbb{R}^{(B\cdot N)\times L_v\times H\times C}. \tag{8}$$

Correspondingly, we repeat keys and values N times along the batch dimension:

$$\tilde{\mathbf{K}},\ \tilde{\mathbf{V}} \in \mathbb{R}^{(B\cdot N)\times L\times H\times C}. \tag{9}$$

We then compute attention to obtain $\tilde{\mathbf{O}} \in \mathbb{R}^{(B\cdot N)\times L_v\times H\times C}$, and reshape it back to $\mathbf{O} \in \mathbb{R}^{B\times L\times H\times C}$. This rearrangement guarantees that each sample in $\tilde{\mathbf{Q}}$ corresponds to a single query view, enabling URoPE to be applied in the same manner as standard RoPE.

Analysis of computation complexity. The time complexity of attention operation over $\tilde{\mathbf{Q}}, \tilde{\mathbf{K}}, \tilde{\mathbf{V}}$ is

$$O(BN \times H \times L_v \times L \times C) = O(BHL^2C), \tag{10}$$

which matches the complexity of attention over $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. In terms of memory, the main overhead comes from repeating $\mathbf{K}, \mathbf{V}$ along the batch dimension during the current layer's computation; this is transient and typically modest relative to the total memory footprint across layers. Overall, URoPE introduces no additional asymptotic computational burden to Transformers.

4 Experiments

In this section, we conduct extensive experiments to evaluate URoPE on three tasks including novel view synthesis (Sec 4.1), 3D object detection (Sec 4.2), and stereo depth estimation (Sec 4.3). For all the tasks, URoPE is integrated as a plug-in relative position embedding for the attention layers without any task- or model-specific designs. Afterwards, we give in-depth analysis of the role of URoPE in Sec. 4.4. Finally, we justify several key design modules in Sec. 4.5.

4.1 Novel View Synthesis Task

Novel view synthesis requires a comprehensive understanding of the 3D environment by aggregating visual information across multiple camera views.

Setup. We combine URoPE with LVSM [13] framework, a decoder-only transformer that performs self-attention across reference and target view tokens. URoPE serves as the relative position embedding, replacing the plücker ray. We follow P-RoPE [15] to reduce the model size due to limited computational resources. We conduct experiments on two datasets: Objaverse [5] and RealEstate10k [43]. Objaverse [5] contains synthetic 3D objects, and the images are rendered from diverse viewpoints. To improve the difficulty, we randomly change the camera focal lengths for the images. RealEstate10k [43] contains real-world indoor and outdoor scenes gathered from YouTube videos. We evaluate the quality of novel view synthesis with three standard metrics: PSNR, SSIM, and LPIPS.

Baselines. We compare URoPE with **1) Plücker ray**: the absolute position embedding for camera rays adopted in LVSM [13]; **2) 6D RoPE**: rotary position embedding in 6D ray space mentioned in Sec. 3.1; **3) P-RoPE [15]**: relative position embedding that combines RoPE on image space and camera parameters across views; **4) RayRoPE** [36]: a concurrent work that attaches extra depth prediction module in attention layers. For fairness, we run all the baselines on the same datasets by ourselves.

Results. We report the quantitative comparison between different methods in Tab. 2. On both datasets, a notable performance gain of URoPE is witnessed compared to all the baselines. On the Objaverse dataset, URoPE can handle the challenging case of different camera intrinsic parameters, thanks to the design of explicit projection. On the RealEstate10k dataset, URoPE also performs well on large-scale scenes with complex geometry and appearance. The results show that URoPE can handle the self-attention including intra- and inter-view image features. By using fixed depth anchors, URoPE aligns cross-view features without additional parametric depth modules, simplifying optimization and avoiding errors from inaccurate depth prediction. Fig. 3 shows qualitative examples, where URoPE improves LVSM's ability to render fine-grained details.

4.2 3D Object Detection and Tracking Task

Vision-based 3D perception aims to recognize objects in 3D space using 2D images. Transformer-based algorithms often depend on the cross-attention module

Table 2: Results on Novel View Synthesis

Methods	*Objaverse*			*RealEstate10k*		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Plücker Ray [13]	22.28	0.856	0.279	23.95	0.764	0.118
6D RoPE	24.42	0.891	0.191	25.73	0.819	0.086
P-RoPE [15]	24.88	0.896	0.176	25.28	0.806	0.092
RayRoPE [36]	24.96	0.897	0.175	24.94	0.799	0.097
URoPE (ours)	**25.09**	**0.900**	**0.165**	**26.02**	**0.827**	**0.080**

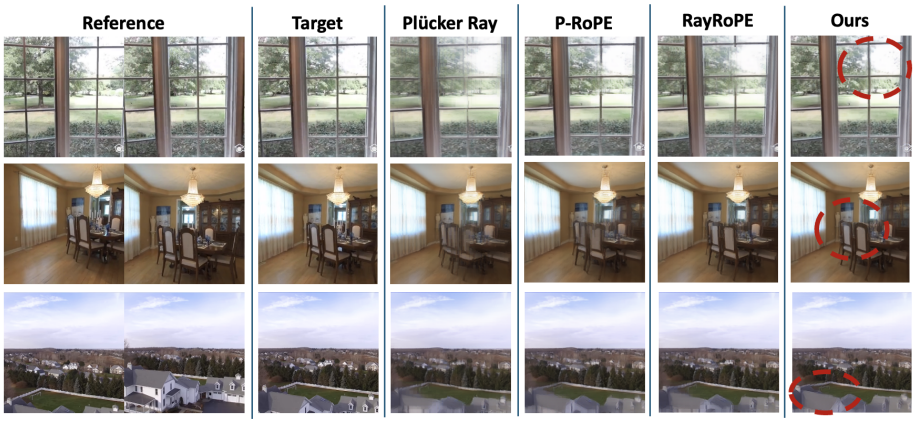


Fig. 3: Qualitative Results for Novel View Synthesis. URoPE exploits relevant local information across views to synthesize sharper details.

for the interaction between 3D queries and image features. As mentioned at the end of Sec. 3.2, URoPE can be generalized to this cross-dimensional interaction.

Setup. To show the compatibility of URoPE, we consider both single-frame and long-horizontal multi-frame algorithms for 3D object detection, namely PETR [19] and StreamPETR [34], with additional tracking module from CenterPoint [42]. The experiments are conducted on nuScenes dataset [1], which contains large-scale outdoor scenes with dynamic objects. We report the NDS and mAP metrics for object detection, and AMOTA metric for object tracking.

Results. As reported in Tab. 3, URoPE consistently improves 3D object detection and tracking performance, which verifies the effectiveness of URoPE in 2D-3D cross-dimensional interaction. URoPE helps to lift the visual features from 2D image space into 3D space by explicitly building correspondence between 3D queries and 2D image features. Especially, Tab. 3b shows that URoPE can handle not only multi-view camera images at the same time step but also long-horizontal multi-frame sequences with fast ego-camera motions. As demonstrated in Fig. 4, URoPE improves the 3D object detection algorithm in the ability to identify small objects. Besides, the performance gain in tracking fur-

Table 3: Results on 3D Object Detection and Tracking

(a) Single-Frame Multiview Performance

Methods	*Detection*		*Tracking*
	NDS↑	mAP↑	AMOTA↑
PETR [19]	34.9	30.9	0.222
PETR + URoPE	**37.3**	**32.2**	**0.255**

(b) Multi-Frame Multiview Performance

Methods	*Detection*		*Tracking*
	NDS↑	mAP↑	AMOTA↑
StreamPETR [34]	47.6	37.5	0.335
StreamPETR + URoPE	**50.6**	**41.1**	**0.380**

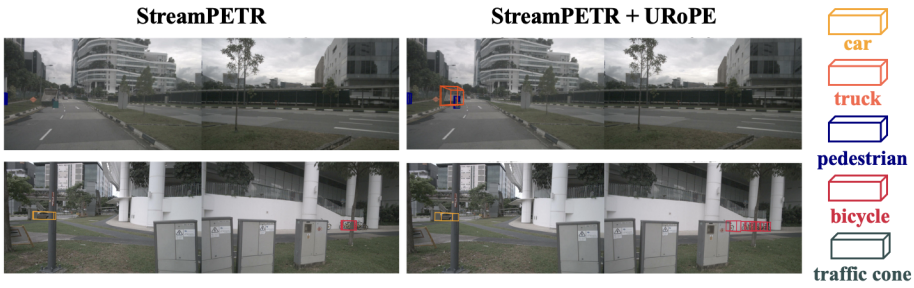


Fig. 4: Qualitative Comparison on 3D Object Detection. URoPE can help to identify small details from multiframe and multiview images.

ther demonstrates that URoPE enhances the robust and coherent recognition of objects along the entire trajectories at multiple frames.

4.3 Stereo Depth Estimation Task

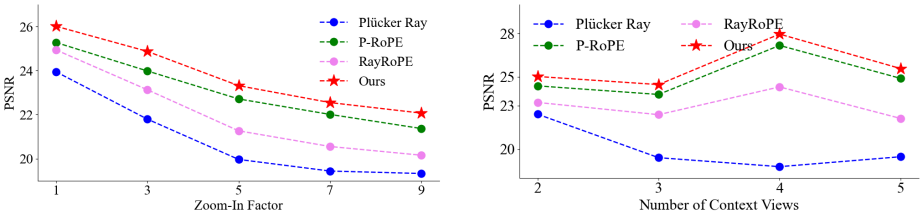
Stereo depth estimation exploits parallax to predict depth, which requires information transfer between two camera views. URoPE can also be integrated into stereo depth estimation algorithms, which can help correspond visual features across camera views.

Setup. We integrate URoPE into UniMatch [41], which is a transformer-based method for stereo depth estimation. We conduct experiments on three datasets: RGBD [26], SUN3D [37], and Scenes11 [30], where RGBD and SUN3D datasets contain real scenes, but Scenes11 dataset is composed with synthetic scenes. We evaluate the performance with standard metrics for depth estimation task including Abs Rel, Sq Rel, RMSE, and RMSE log.

Results. We report the metrics on all three datasets in Tab. 4. In the stereo depth estimation task, there are only two camera views with a small distance between each other, and the model has a decoupled design for intra-view self-attention and inter-view cross-attention. Both factors simplify the combination of multiview visual information, and the advantage of explicit projection cannot be fully exhibited. However, the model can still benefit from URoPE to consistently improve its performance in this simple case, which reflects the versatility of our proposed method.

Table 4: Results on Stereo Depth Estimation. † : The "Sq Rel" metric is less reliable on the RGBD dataset due to the imperfect depth and camera pose [30].

Datasets	Methods	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓
RGBD [26]	UniMatch [41]	0.123	0.175†	0.678	0.203
	UniMatch + P-RoPE [15]	0.105	0.203†	0.573	**0.181**
	UniMatch + URoPE	**0.103**	0.201†	**0.571**	**0.181**
Scenes11 [30]	UniMatch [41]	0.065	0.085	0.575	0.126
	UniMatch + P-RoPE [15]	**0.049**	0.063	0.474	**0.104**
	UniMatch + URoPE	**0.049**	**0.062**	**0.450**	**0.104**
SUN3D [37]	UniMatch [41]	0.131	0.098	0.397	0.169
	UniMatch + P-RoPE [15]	0.117	0.075	0.343	0.152
	UniMatch + URoPE	**0.112**	**0.063**	**0.329**	**0.148**



(a) Different Camera Focal Lengths. **(b)** Different Context View Numbers.

Fig. 5: Robustness Analysis. Although the model is trained with fixed camera focal length and 2 context views, URoPE can maintain a robust performance in out-of-distribution cases. Fig. 5b adopts different context view selection with Tab. 2, so the metrics are not aligned.

Table 5: Scaling-up (50×) Experiment Results for Novel View Synthesis

Methods	*RealEstate10k* PSNR↑	SSIM↑	LPIPS↓
Plücker Ray [13]	28.66	0.889	0.113
URoPE (ours)	**29.24**	**0.897**	**0.104**

4.4 Analysis

Scaling-up Experiments. As a relative position embedding, it is critical for URoPE to exhibit improved performance with scaled-up computational resources. To evaluate the scalability of URoPE, we scale up the novel view synthesis experiments on RealEstate10k dataset (Sec. 4.1) by enlarging the training batch size, network depth, and network width, which as a result yields a 50× larger computational cost. As shown in Tab. 5, URoPE still brings reasonable performance gain compared to the default setting with scaled-up computational resources, which reflects the great scalability of our proposed URoPE.

Fig. 6: Region-wise Anchor Depth Importance. In the middle layer, the importance of depth anchors is highly related to the actual depth, while this correlation is not significant in both the shallow and deep layers.

Table 6: Analysis of Multihead Collapse. We calculate the attention score entropy across heads for all the key/value patches from the context views per attention layer.

Methods	Per-layer head-wise normalized entropy					
	1	2	3	4	5	6
P-RoPE	0.93	0.71	0.82	0.80	0.83	0.79
URoPE	0.95	0.87	0.84	0.80	0.82	0.83

Robustness Analysis. We dig into the ability of URoPE in out-of-distribution scenarios. We consider two out-of-distribution scenarios in Fig. 5: 1) Different camera focal lengths and 2) More context views. Although URoPE is trained with fixed camera intrinsic parameters and two context views, it could generalize to other out-of-distribution scenarios during inference, where it maintains robust performance to all these changes. Since URoPE exploits explicit epipolar projection to obtain the spatial correlation across camera views, it is not very sensitive to the change of camera parameters and view number. As a result, it naturally has great robustness in out-of-distribution situations.

Region-wise Anchor Depth Importance. Since we assign multiple anchor depths to each image patch, it is natural to think whether the actual depth affects the importance of each depth anchor. Since each depth anchor is associated with different attention heads, we can explore the head-wise importance in attention module for different regions as an indicator for the anchor depth importance. We visualize the distribution of the most important attention head in the self-attention for all the key/value image patches from context views, where the depth of anchors increases along with the order of the attention head. The strong correlation between real depth and anchor importance are only witnessed in the middle layer, as visualized in Fig. 6. This fact reveals that the network can learn to adaptively exploit different depth anchors based on the local information. In contrast, the correlation is weak in both the shallowest and deepest layers. In the shallow layers, the model does not have enough information to utilize specific anchor depths, while in the deep layers, the model may prefer learning more high-level semantic information beyond the depth value.

Risk of Multi-Head Collapse. With head-wise depth-anchor assignment, a potential failure mode is *multi-head collapse*, i.e., only a small subset of heads

Table 7: Effect of Combining Absolute and Relative Position Embeddings. The experiments are conducted on RealEstate10k using LVSM model.

Abs. Pos.	**Rel. Pos.**	PSNR↑	SSIM↑	LPIPS↓
plücker ray	-	23.95	0.764	0.118
plücker ray	*2D RoPE*	24.96	0.795	0.098
plücker ray	*URoPE*	25.89	0.826	0.078
local ray	*URoPE*	26.00	0.828	0.077
-	*URoPE*	25.85	0.824	0.083

corresponding to specific depth anchors dominates attention for most key/value tokens. To examine this, we conduct a statistical analysis. For each key/value token from the context views, we compute its average attention score among all the query tokens from the target view for each head. Then, we calculate the entropy across all the attention heads, which is averaged over key/value tokens from multiple samples as a metric for multihead collapse and normalzed into $(0, 1)$. Higher entropy indicates better balance across heads. In Tab. 6, we find the entropy values are similar between URoPE and P-RoPE [15] across all the six layers. This shows that our head-wise depth anchor assignments would not cause the multihead collapse.

4.5 Ablation Study

In this part, we justify the design of key modules in URoPE through ablation studies. The experiments are conducted on the RealEstate10k dataset [43] using the LVSM model [13] with the same setup as Sec. 4.1. Due to some different hyperparameters, the results may differ from Tab. 2, but the fairness of each comparison is guaranteed.

Combination of Absolute and Relative Position Embedding. We explore whether URoPE can benefit from knowing the absolute camera ray position in the world coordinate by combining URoPE with the absolute position embedding. The results are reported in Tab. 7. URoPE can hardly achieve a performance gain when combined with global plücker rays since URoPE already contains enough information to reflect the spatial relationships across multi-view images. In contrast, URoPE can obtain some slight improvements if local ray directions in the camera coordinate are integrated since it reflects the intra-view positions. However, for simplicity, we do not include these local camera rays in other experiments by default.

Depth Anchor Splitting. We consider two ways to split anchor depth: head-wise *v.s.* channel-wise split. Head-wise splitting assigns multiple anchor depths to different attention heads, while channel-wise splitting divides the per-head feature into different depth anchors. Due to the limited feature dimension, channel-wise splitting cannot support many anchor depths. As shown in Tab. 8, head-wise splitting can greatly improve performance. With head-wise splitting, we

Table 8: Ablation of URoPE on RealEstate10k with LVSM [13]. **Number** is the count of fixed depth anchors. **Splitting** assigns anchors either to heads (head-wise) or to channels (channel-wise). **Sampling** controls how anchor depths are spaced (uniform, log-uniform, or LID [28]). **Depth** indicates whether we use fixed anchor depths or a learned per-layer depth prediction module.

Depth	Number	Splitting	Sampling	PSNR↑	SSIM↑	LPIPS↓
fixed anchors	16	*head-wise*	*uniform*	25.85	0.824	0.083
	8			25.84	0.823	0.084
	4			**26.01**	**0.827**	0.082
	2			25.94	0.826	0.082
	1			25.37	0.807	0.092
	2	*channel-wise*	*uniform*	25.47	0.815	0.087
	4	*head-wise*	*LID*	**26.01**	**0.827**	**0.081**
			log-uniform	25.94	0.825	0.082
learned	–	–	–	25.57	0.818	0.085

can accommodate more components of different frequencies to deal with both short-range and long-range relationships.

Number of Depth Anchors. With head-wise splitting, we consider different numbers of depth anchors. When the depth anchors are fewer than the attention heads, we divide the attention heads into groups, and each group shares the same depth anchor. In Tab. 8, we compare the performance with different numbers of depth anchors. Results show that the performance of URoPE is relatively robust for the anchor number. We think four anchor depths are already enough to reflect the position of the projected epipolar line, which helps to guide the attention module to focus on the most important regions across camera views.

Sampling of Depth Anchors. We consider three different ways to sample depth anchors. 1) Uniform: The anchor depths are sampled uniformly within a range. 2) Log Uniform: The logarithm of anchor depths are sampled uniformly. 3) Linear-Increasing Discretization (LID) [28]: The bin size of each range linearly increases along the depth dimension. In Tab. 8, we compare these three different methods. We find that the uniform sampling and LID sampling achieve the similar great performance since both of them can represent the projected epipolar line in a balanced manner.

Parametric Depth Estimation Module. URoPE utilizes fixed depth anchors to build the relationship across the geometric gap. We consider an alternative way to predict the depth in each attention layer. In this case, we inject a lightweight parametric depth estimation module for each layer instead of the fixed depth anchors. As shown in Tab. 8, this parametric depth estimation brings worse performance than fixed depth anchors since it is challenging to predict accurate depth from the image feature, as it may not include enough geometric cues, especially in the shallow layers. In contrast, fixed anchors can lead to stable performance regardless of the semantics in the image feature.

5 Conclusion

In this paper, we presented URoPE, a universal extension of relative position embedding for cross-view and cross-dimensional geometric reasoning. By lifting key image tokens at fixed depth anchors and projecting them into the query image plane, URoPE enables standard 2D RoPE to encode cross-view correspondences in a shared image coordinate system. The resulting encoding is parameter-free, intrinsics-aware, invariant to global coordinate, and natively compatible with RoPE-optimized attention kernels. Experiments on novel view synthesis, 3D object detection, and stereo depth estimation show that URoPE consistently outperforms prior position encodings across all the benchmarks, while demonstrating strong out-of-distribution generalization. A current limitation is the reliance on known camera parameters. It is a promising direction to extend URoPE to uncalibrated settings in future work.

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. Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp. 2978–2988 (2019)
4. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
5. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13142–13153 (2023)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Du, Y., Smith, C., Tewari, A., Sitzmann, V.: Learning to render novel views from wide-baseline stereo pairs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4970–4980 (2023)
8. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
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. He, Y., Yan, R., Fragkiadaki, K., Yu, S.I.: Epipolar transformers. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 7779–7788 (2020)
11. Heo, B., Park, S., Han, D., Yun, S.: Rotary position embedding for vision transformer. In: European Conference on Computer Vision. pp. 289–305. Springer (2024)
12. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
13. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: Lvsm: A large view synthesis model with minimal 3d inductive bias. arXiv preprint arXiv:2410.17242 (2024)
14. Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., Yang, L., Liu, J., Fan, H., Liu, S.: Practical stereo matching via cascaded recurrent network with adaptive correlation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16263–16272 (2022)
15. Li, R., Yi, B., Liu, J., Gao, H., Ma, Y., Kanazawa, A.: Cameras as relative positional encoding. arXiv preprint arXiv:2507.10496 (2025)
16. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(3), 2020–2036 (2024)
17. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
18. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9298–9309 (2023)
19. 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)
20. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3651–3660 (2021)
21. Miyato, T., Jaeger, B., Welling, M., Geiger, A.: Gta: A geometry-aware attention mechanism for multi-view transformers. arXiv preprint arXiv:2310.10375 (2023)
22. Press, O., Smith, N.A., Lewis, M.: Train short, test long: Attention with linear biases enables input length extrapolation. arXiv preprint arXiv:2108.12409 (2021)
23. Sajjadi, M.S., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lučić, M., Duckworth, D., Dosovitskiy, A., et al.: Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6229–6238 (2022)
24. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 464–468 (2018)
25. Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. Advances in Neural Information Processing Systems **34**, 19313–19325 (2021)

26. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 573–580. IEEE (2012)
27. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
28. Tang, Y., Dorn, S., Savani, C.: Center3d: Center-based monocular 3d object detection with joint depth understanding. In: DAGM German Conference on Pattern Recognition. pp. 289–302. Springer (2020)
29. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
30. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5038–5047 (2017)
31. 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)
32. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
33. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
34. 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)
35. 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)
36. Wu, Y., Jeon, M., Chang, J.H.R., Tuzel, O., Tulsiani, S.: Rayrope: Projective ray positional encoding for multi-view attention. arXiv preprint arXiv:2601.15275 (2026)
37. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE international conference on computer vision. pp. 1625–1632 (2013)
38. Xie, Y., Peng, C., Abdelfattah, M., Hu, Y., Yang, J., Higgins, E., Brigden, R., Tomizuka, M., Zhan, W.: Raynova: Scale-temporal autoregressive world modeling in ray space. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
39. Xie, Y., Xu, C., Peng, C., Zhao, S., Ho, N., Pham, A.T., Ding, M., Tomizuka, M., Zhan, W.: X-drive: Cross-modality consistent multi-sensor data synthesis for driving scenarios. arXiv preprint arXiv:2411.01123 (2024)
40. Xie, Y., Xu, C., Rakotosaona, M.J., Rim, P., Tombari, F., Keutzer, K., Tomizuka, M., Zhan, W.: Sparsefusion: Fusing multi-modal sparse representations for multi-sensor 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17591–17602 (2023)
41. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

42. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)
43. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
44. 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)

In this appendix, we exhibit additional experiment results in Sec. A. Afterwards, the implementation details of our experiments are explained in Sec. B. Finally, we discuss some limitations of our work in Sec. C.

Table A.9: Sensitivity to Depth Ranges.

Depth Range	PSNR↑	SSIM↑	LPIPS↓
[2m, 20m]	**26.01**	**0.827**	0.082
[10m, 20m]	25.67	0.816	0.088
[2m, 10m]	25.93	0.826	0.082
[2m, 30m]	25.98	0.826	**0.081**
[2m, 40m]	25.88	0.824	0.083

A Additional Experiment Results

A.1 Sensitivity to Depth Ranges

URoPE samples multiple depth anchors within a pre-defined depth range. We conduct an additional ablation study to evaluate the sensitivity of URoPE to the pre-defined depth ranges. All the experiment settings are same as Sec. 4.5 and we sample 4 depth anchors within the range uniformly. We report the results in Tab. A.9. It is important to cover the close regions, but URoPE is robust to different reasonable upper bounds. As a result, URoPE is not very sensitive to the selection of depth ranges.

A.2 Additional Qualitative Results

In Fig. A.7, we visualize some additional results on novel view synthesis using different methods. Compared with other position embeddings, URoPE helps produce sharper details and the synthesized novel views are more spatially reasonable. In Fig. A.8, we also provide some additional visualizations for 3D object detection task. URoPE helps to identify some small and confusing objects.

B Experiment Details

B.1 Novel View Synthesis Details

We follow the implementation in [15] to reduce the computational burden of the original LVSM model [13]. Some key implementation details are listed as follows.

- The image resolution are fixed at 256×256 for both training and inference with a transformer patch size 8×8.

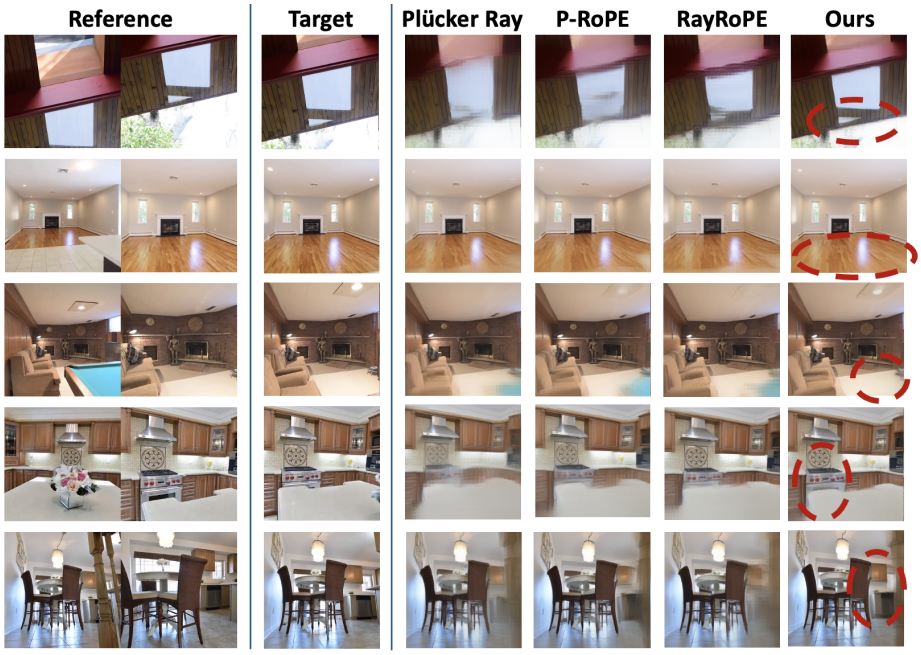


Fig. A.7: Additional Qualitative Results for Novel View Synthesis. URoPE helps to produce sharper details and synthetic novel views are more geometrically reasonable.

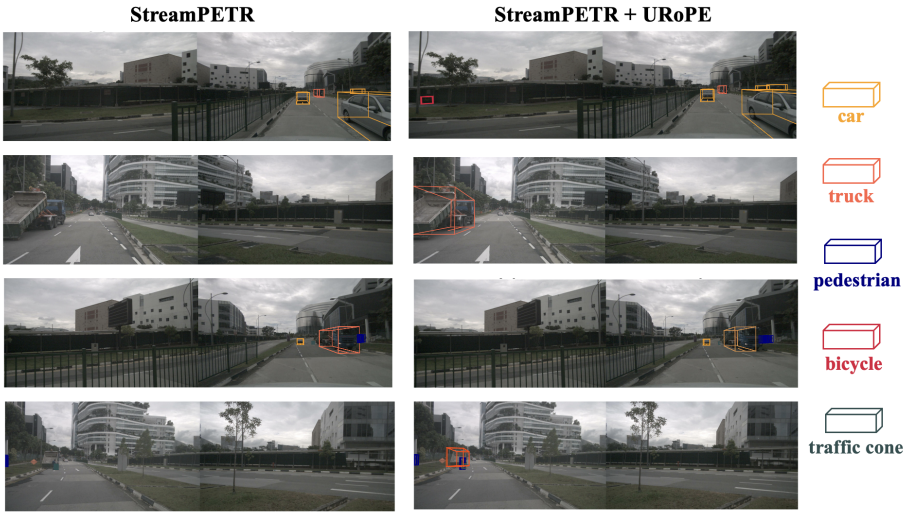


Fig. A.8: Additional Qualitative Results for 3D Object Detection. URoPE helps to identify some and difficult objects in the 3D space.

- Given the limited computational resources, we reduce the number of transformer blocks from 24 to 6. The MLP channel dimension is also reduced from 3072 to 1024. This modification allows us to train the model using 2 GPUs with a total batch size of 16 for 8k iterations on both RealEstate10k [43] and Objaverse [5] datasets.
- For the Objaverse dataset [5], we download the rendered images provided by Zero-1-to-3 [18]. To evaluate the robustness to camera parameters, we follow RayRoPE [36] to randomly change the camera focal length with a random scale between 0.4 and 1.6.

To ensure the fair comparison, we reproduce all the methods in Tab. 2 by ourselves on both datasets using their public codes.

B.2 3D Object Detection and Tracking

We integrate URoPE into PETR [19] and StreamPETR [34] with their official code. Both models adopt ResNet50 [9] backbone with a resolution of 256×704 for input images. They are trained for 24 epochs on nuScenes dataset [1] with a total batch size of 16 using 2 GPUs. Both models include 900 queries that attend to multiview image features through 2D-3D cross-attention modules.

B.3 Depth Estimation

For depth estimation, we adopt UniMatch model [41]. It consists of intra-image self-attention and inter-image cross-attention modules. We follow [15] to integrate URoPE into both self-attention and cross-attention modules on the Q/K/V/O vectors. As explained in Sec. 3.2, URoPE degrades to RoPE [11] in the intra-view self-attention. For all the three datasets, the model is trained for 100k steps using 3 GPUs with a total batch size of 78. The image resolution is set as 448×576.

C Limitations

URoPE relies on ground-truth camera intrinsic and extrinsic parameters for lifting and projection, which limits its applicability to tasks where camera calibration is available. Consequently, it cannot be directly applied to scenarios such as 3D reconstruction from uncalibrated camera observations. In addition, due to limited computational resources, we do not evaluate our approach on large-scale models. In future work, we plan to extend URoPE to operate with estimated camera parameters, enabling its integration with modern 3D reconstruction frameworks such as VGGT [32] and DP3 [17].