HROPE: HOMOGRAPHY RELATIVE POSITIONAL ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric Relative Positional Encoding has become the primary mechanism for injecting spatial inductive biases into multi-view transformers. Existing designs condition attention on camera-level frustums or ray-marched depth, associating cross-view tokens through the sensor geometry. Such sensor-centric formulations, however, describe only where a token is observed from, leaving the rich surface geometry of the scene region depicted by the token. In this paper, we advocate a shift toward surface-aware positional encoding that explicitly grounds patch tokens in the geometry of physical surfaces. We introduce **HRoPE**, which represents the local geometry of each patch as a slanted 3D plane, following the slanted support-window assumption of classical multi-view stereo. We inject token-level, plane-induced homographies into attention as relative transformations. This explicitly enables the transformer to rectify perspective distortions and align patch features across viewpoints rather than having to learn this alignment implicitly from data. Our encoding requires no ground-truth geometry: the plane parameters can be predicted end-to-end under a view-synthesis objective alone, or derived from foundation models. Extensive experiments on RealEstate10K, CO3Dv2, and DL3DV show that our best variant of HRoPE consistently outperforms the evaluated geometric positional encodings for novel view synthesis. Extensive ablations and analysis show the effectiveness of the proposed encoding.

1 INTRODUCTION

The Transformer has become the de facto architecture for recent feed-forward 3D vision (Wang et al., 2024; 2025; Jin et al., 2025). By tokenizing images from multiple viewpoints and exchanging information via attention, these models exhibit strong capabilities in jointly resolving appearance, visibility, and scene geometry. Attention, however, is permutation-equivariant: it operates on unordered sets and is inherently agnostic to the camera poses, metric scales, and epipolar constraints that couple multi-view observations. Consequently, geometric priors must be explicitly injected, either by conditioning token representations directly (e.g., via camera tokens (Wang et al., 2025) or ray maps (Jin et al., 2025; Zhang et al., 2024a)) or by imposing epipolar constraints on attention (He et al., 2020; Charatan et al., 2024; Ye et al., 2024). Positional encoding (PE) constitutes a distinct approach within this design space, which modulates attention directly, shaping how tokens are compared rather than what each token carries.

While existing PE designs exhibit increasing geometric fidelity by using priors from images, cameras, and rays, they invariably anchor positional information to the sensor. RoPE extensions (Su et al., 2024) encode 2D intra-view offsets, leaving inter-view transformations implicit (Weinzaepfel et al., 2023; Wang et al., 2024). GTA (Miyato et al., 2024) and CaPE (Kong et al., 2024) inject relative camera extrinsics to achieve invariance to global rigid motions, while PRoPE (Li et al., 2025) generalizes this to the full projection matrix. Ray-based formulations like RayRoPE (Wu et al., 2026) and URoPE (Xie et al., 2026) integrate depth by localizing patches along viewing rays and re-projecting into the query frame. Across this progression, geometry is parameterized via sensor-centric abstractions: an image coordinate, a camera frustum, or a point of ray termination, leaving the rich local surface structure of the underlying scene unmodeled (see Fig. 1, left and middle).

This limitation is fundamental because multi-view matching does not compare isolated points; it compares local neighborhoods whose appearances are related through surface-induced deforma-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

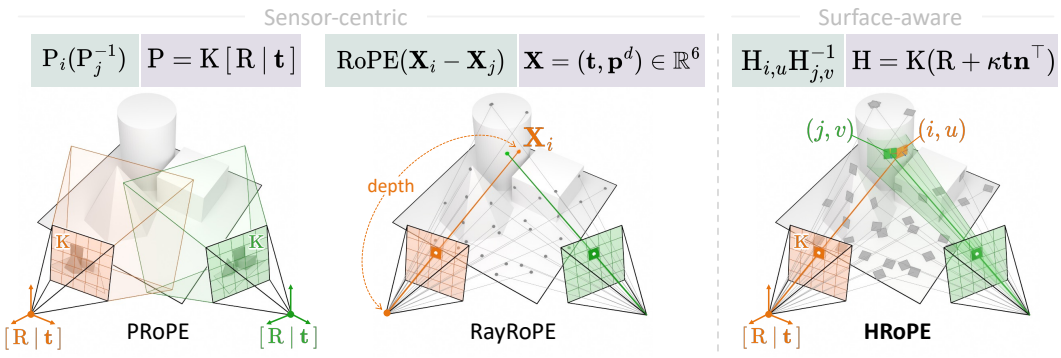


Figure 1: **From sensor-centric to surface-aware geometric positional encoding. Left:** PRoPE (Li et al., 2025) assigns every token in a view the same camera projection $P = K[R \mid t]$. **Middle:** RayRoPE (Wu et al., 2026) predicts a depth d for each token and places it at a 3D point X along the camera ray, but does not model local surface orientation. **Right:** HRoPE associates each patch with a slanted tangent plane and relates token pairs through the plane-induced homography $H_{i,u}H_{j,v}^{-1}$ (Eq. 2), thereby encoding both depth and local surface orientation in the attention comparison.

tions. A patch in one view and its corresponding patch in another are connected by a warp determined by the underlying surface geometry, and reliable correspondence requires compensating for this warp before comparing visual features. Given the pixel location and camera geometry, depth locates the patch center, while surface orientation additionally determines how its surrounding neighborhood deforms across views. By providing only view or positional information, existing PEs implicitly impose a fronto-parallel assumption on the attention computation mechanism. Although large neural models can learn to compensate for this bias given sufficient training data, doing so compromises efficiency and generalization. We argue that surface orientation is the missing degree of freedom: current PEs capture the zeroth-order position of a patch but neglect the first-order local orientation that dictates its cross-view appearance transformation.

To this end, we present **HRoPE**, a Homography Relative Positional Encoding that embeds orientation-aware rectification directly into the attention mechanism (Fig. 1, right). Inspired by the slanted support windows of classical multi-view stereo (Bleyer et al., 2011; Wang et al., 2021a), HRoPE anchors a local tangent plane to each patch token. Rather than treating this tangent plane as an auxiliary feature, we use its normal and plane distance, together with the camera poses to uniquely define the homography that governs cross-view neighborhood deformations. Concretely, we apply a per-patch 3×3 transformation block-diagonally to the queries, keys, and values, and incorporate a closed-form inverse into the key projection so that the resulting attention logits depend exclusively on the relative homography between tokens. As we show in Sec. 3.2 and Appendix A.2, this mechanism acts as an in-attention analogue of computing a matching score only after both neighborhoods have been rectified onto a shared tangent plane. Consequently, whereas prior geometric PEs merely encode where a token is located, HRoPE explicitly models how its underlying surface transforms across views. This formulation offers two compelling advantages. First, HRoPE strictly generalizes point- and ray-based PEs: when a predicted normal aligns with the viewing direction, the induced transformation naturally reduces to the fronto-parallel case. Second, the per-patch transformation parameters can be regressed from patch features by a lightweight head trained jointly with the view-synthesis model, using the view-synthesis objective alone. The head can optionally use pretrained features from DINOv2 (Oquab et al., 2024) or VGGT (Wang et al., 2025), without changing the encoding formulation or training objective.

We validate HRoPE on feed-forward novel view synthesis (NVS) across the RealEstate10K (Zhou et al., 2018), CO3Dv2 (Reizenstein et al., 2021), and DL3DV (Ling et al., 2024) datasets, where it consistently outperforms view-, projection-, and ray-based encodings. Notably, HRoPE surpasses PRoPE on RealEstate10K without ground-truth camera poses, demonstrating that per-patch homography is a stronger prior than view-level projection. Furthermore, our ablations show that this geometry must condition token interactions multiplicatively, injecting the same information as additive features sharply degrades performance. Finally, HRoPE acts as a genuine inductive bias: it markedly improves out-of-distribution generalization (e.g., unseen zooms and wider baselines) by

enabling geometric extrapolation rather than overfitting to the training view configurations. Beyond NVS, HRoPE transfers unchanged to a multi-view diffusion model and to the visual tokens of a spatial reasoning MLLM, improving performance on both tasks (Appendix A.7).

We summarize our main contributions as follows:

1. We formulate *the principle of surface-aware positional encoding*, incorporating local surface orientation into the comparison of patch neighborhoods across views.
2. We propose **HRoPE**, a per-patch relative encoding that uses plane-induced transforms with closed-form inverses to condition attention logits on relative cross-view homographies.
3. We demonstrate consistent improvements over the evaluated geometric positional encodings on multiple NVS benchmarks and under out-of-distribution view settings. Controlled ablations analyze the roles of transformation structure, local geometry, and feature source, supporting the benefit of the homography inductive bias.

2 Related Work

Geometric Positional Encoding. Geometric PEs encode camera or scene geometry in tokens and attention. Absolute variants encode Plücker rays (Sitzmann et al., 2021; Sajjadi et al., 2022; Jin et al., 2025) or 3D positions (Liu et al., 2022), but depend on the chosen world frame. Relative variants extend rotary encodings (Su et al., 2024; Shaw et al., 2018) to camera geometry: GTA (Miyato et al., 2024) and CaPE (Kong et al., 2024) use relative extrinsics, while PRoPE (Li et al., 2025) also includes intrinsics through projection matrices. Recent methods move from view- to token-level geometry: RayRoPE (Wu et al., 2026) predicts per-token depth, RoRE (Griffiths & Dansereau, 2026) rotates features along rays, and URoPE (Xie et al., 2026) projects per-head depth anchors into the query frame, following epipolar attention (He et al., 2020; Suhail et al., 2022). These encodings describe camera relationships and point- or ray-level geometry. HRoPE adds an explicit local-plane parameterization whose orientation, together with distance and camera geometry, determines a planar neighborhood's cross-view deformation. This distinguishes surface-conditioned transformations from point localization without requiring the latter to assume fronto-parallel surfaces.

Multi-view Transformers. Multi-view transformers use camera geometry to associate image tokens across views. Their applications include stereo depth estimation (He et al., 2020; Yao et al., 2018; Gu et al., 2020), camera and structure recovery (Wang et al., 2024; 2025), cross-view completion and matching (Weinzaepfel et al., 2023), spatial reasoning (Yang et al., 2025; Wu et al., 2025), and camera-controlled video and world modeling (Li et al., 2026; Zhang et al., 2026), as well as novel view synthesis (NVS). For feed-forward NVS, methods use feature-conditioned radiance fields (Yu et al., 2021; Wang et al., 2021b; Chen et al., 2021; Trevithick & Yang, 2021; Hong et al., 2024), predict 3D Gaussians (Charatan et al., 2024; Chen et al., 2024a; Zhang et al., 2024a), or directly regress pixels (Jin et al., 2025; Jiang et al., 2025). We evaluate HRoPE primarily on novel view synthesis using the LVSM backbone with plane-induced positional encoding.

Slanted Support Windows. Traditional local stereo methods assume constant disparity within a support window, introducing a fronto-parallel bias on slanted surfaces (Bleyer et al., 2011). Building on randomized correspondence search (Barnes et al., 2009), PatchMatch Stereo (Bleyer et al., 2011) addresses this limitation by estimating a local disparity plane at each pixel and evaluating matching costs over slanted support windows. More generally, local planar geometry induces a homography that relates corresponding image neighborhoods across views (Hartley & Zisserman, 2003). Learned multi-view stereo methods also use differentiable geometric warping to compare features under depth hypotheses (Yao et al., 2018; Gu et al., 2020; Wang et al., 2021a). HRoPE draws on the principle of using local surface geometry to guide cross-view matching. It incorporates per-patch, plane-induced transformations into multi-view attention, bringing the geometric intuition of slanted-window matching into positional encoding.

3 Method

HRoPE is a geometric positional encoding that incorporates a per-token surface plane transform. In Sec.3.1, we first review how prior work incorporates camera geometry into multi-view attention. Then, in Sec. 3.2, we introduce the plane-induced transformation and show how it modulates multi-view attention. With consistent camera geometry, the relative transformation exactly maps points

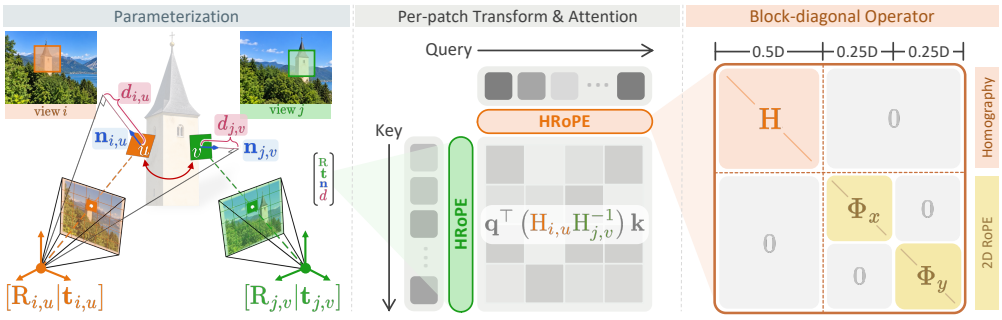


Figure 2: **Overview of HRoPE. Left:** per-patch parameters $(\mathrm{R}, \mathbf{t}, \mathbf{n}, \kappa)$ are predicted from patch-level image features (Sec. 3.3), where $\mathbf{n}$ and $\kappa = 1/d$ represent the local plane's unit normal and inverse plane distance, respectively. Together with intrinsics K, these parameters define $\mathrm{H} = \mathrm{K}\left(\mathrm{R} + \kappa\ \mathbf{t}\ \mathbf{n}^\top\right)$ (Eq. 5). **Middle:** token-specific transforms modulate the cross-view attention, yielding the relative factor $\mathrm{H}_{i,u}\mathrm{H}_{j,v}^{-1}$ in each attention logit (Eq. 4). **Right:** the transform is repeated block-diagonally over the geometric half of each head, while the remaining dimensions retain the 2D rotary blocks Φ_x and Φ_y (Eq. 17).

on a shared physical plane across views. For independently predicted per-patch parameters, the same algebraic form modulates attention without requiring an exact physical correspondence for every token pair. Finally, Sec. 3.3 details how these per-patch parameters are predicted from image features and presents a calibrated variant that leverages known camera parameters. The overall pipeline of HRoPE is illustrated in Fig. 2.

3.1 Preliminaries: Geometric Positional Encodings

Self-attention alone does not encode token positions or camera geometries. RoPE (Su et al., 2024) applies position-dependent rotations to queries and keys so that their dot product depends on relative positions. Multi-view relative PEs extend this principle to cameras: GTA (Miyato et al., 2024) and CaPE (Kong et al., 2024) encode relative extrinsics, while PRoPE (Li et al., 2025) adds intrinsics.

Projective Relative Encodings. Let P_i be PRoPE's 4×4 projection transform for view i, built from its intrinsics and extrinsics. For a query $\mathbf{q}$ in view i and a key $\mathbf{k}$ in view j, the attention score is $(\mathrm{P}_i^\top \mathbf{q})^\top (\mathrm{P}_j^{-1}\mathbf{k}) = \mathbf{q}^\top \left(\mathrm{P}_i \mathrm{P}_j^{-1}\right) \mathbf{k}$. Thus, the score depends on the projection transforms only through $\mathrm{P}_i\mathrm{P}_j^{-1}$, which is invariant to a shared change of world coordinates. Yet all tokens in a view share the same transform, so this view-level modulation cannot encode local surface variation.

From Camera to Surface Geometry. RayRoPE (Wu et al., 2026) and URoPE (Xie et al., 2026) move to token-level geometry by locating points along viewing rays, but still represent a patch through sensor- or point-centered coordinates rather than the deformation of its local surface. We instead associate each patch with a tangent plane to capture both its position and local surface orientation. Composing the camera projection with a surface-induced transform then allows the positional encoding to account for how the patch's surrounding neighborhood deforms across views.

3.2 Homography Relative Positional Encoding

Suppose the local surface observed by an image patch lies on a plane $\mathbf{n}^\top \mathbf{X} = d$ in the world coordinate frame, with unit normal $\mathbf{n}$ and signed distance d. The projection of points $\mathbf{X}$ on this plane into view i can be expressed as a linear mapping $\tilde{\mathbf{x}}_i \sim \mathrm{H}_i \mathbf{X}$, where

$$\mathrm{H}_i = \mathrm{K}_i \left(\mathrm{R}_i + \frac{\mathbf{t}_i \mathbf{n}^\top}{d} \right). \tag{1}$$

Applying this similarly for view j and eliminating $\mathbf{X}$ yields the plane-induced homography (Malis & Vargas, 2007) between the two views:

$$\tilde{\mathbf{x}}_i \sim \mathrm{H}_i \mathrm{H}_j^{-1} \tilde{\mathbf{x}}_j. \tag{2}$$

For visible points on the assumed plane, Eq. 2 determines a correspondence between the two views, whereas epipolar geometry alone restricts the match to a line. When a local neighborhood is approximately planar, the same homography describes its cross-view deformation. This use of a local plane is related to the slanted support windows in PatchMatch Stereo (Bleyer et al., 2011).

From Homography to Positional Encoding. Eq. 2 *suggests a factorized form of a relative positional encoding*: a per-view transformation to a common reference frame, composed pairwise such that only the relative transformation survives. PRoPE realizes this structure using the camera projection matrix P_i, where the relative factor $\mathrm{P}_i\mathrm{P}_j^{-1}$ is shared globally by every token pair between views i and j. Replacing P_i with the plane-induced H_i from Eq. 1 allows the relative factor to depend on both the local surface geometry and the camera poses. Furthermore, assigning a distinct plane to each patch makes this transformation specific to each individual token pair.

We adopt this exact formulation: each patch (i, u) is assigned a transformation $\mathrm{H}_{i,u}$ in the form of Eq. 1, parameterized by its local plane, and a pair of patches is related by the composition $\mathrm{H}_{i,u}\mathrm{H}_{j,v}^{-1}$. When both patches correspond to the same physical plane, this product exactly recovers the homography in Eq. 2. In general, it acts as a projective transformation determined by the surface parameters of both patches. We use this as a geometry-conditioned modulation for the query-key comparison, rather than as a strict physical image warp.

Per-patch Transform and Attention. For patch token u in view i, we define:

$$\mathrm{H}_{i,u} = \mathrm{K}_i \left(\mathrm{R}_{i,u} + \kappa_{i,u}\, \mathbf{t}_{i,u}\mathbf{n}_{i,u}^\top\right), \tag{3}$$

where $(R_{i,u}, t_{i,u})$ parameterizes a patch-specific rotation and translation, $\mathbf{n}_{i,u}$ and $\kappa_{i,u}$ act as a unit plane normal and inverse plane distance. Although camera extrinsics are shared within a view, we relax this constraint when using the homography parameterization to modulate learned features. Predicting $(R_{i,u}, t_{i,u})$ per patch allows both the rotational component and the rank-one correction to adapt to local features, while retaining the structured form of a plane-induced transformation. These quantities therefore serve as latent transformation parameters optimized for view synthesis.

Following PRoPE, we multiply the query of patch (i, u) by $\mathrm{H}_{i,u}^\top$ and the key of patch (j, v) by $\mathrm{H}_{j,v}^{-1}$. These operations are applied block-diagonally over 3-dimensional chunks of the attention head dimension. Consequently, the attention logit (omitting standard scaling) is given by:

$$\left(\mathrm{H}_{i,u}^\top \mathbf{q}\right)^\top \left(\mathrm{H}_{j,v}^{-1}\mathbf{k}\right) = \mathbf{q}^\top \left(\mathrm{H}_{i,u}\mathrm{H}_{j,v}^{-1}\right)\mathbf{k}. \tag{4}$$

The logits are normalized using the standard softmax. The value and output transformations are detailed in Appendix A.4. Unlike PRoPE, where all token pairs between views i and j share the same relative factor, HRoPE uses token-specific transforms, allowing the relative factor to vary across token pairs. For two patches u and v in the same view, the geometric part of their attention score depends on $\mathrm{H}_{i,u}\mathrm{H}_{i,v}^{-1}$. This factor is the identity when their predicted transforms are equal; otherwise, it reflects the difference between those transforms. We apply standard 2D RoPE to the remaining feature channels to encode the patches' relative image positions.

Parameterization and Closed-form Inverse. Evaluating the key-side transformation in Eq. 4 requires computing the inverse of every predicted $\mathrm{H}_{j,v}$. Dropping the patch and view indices for brevity, we can decompose the transformation as:

$$\mathrm{A} = \mathrm{R} + \kappa\, \mathbf{t}\mathbf{n}^\top, \qquad \mathrm{H} = \mathrm{K}\mathrm{A}, \qquad \mathrm{H}^{-1} = \mathrm{A}^{-1}\mathrm{K}^{-1}. \tag{5}$$

The intrinsic-free 3×3 matrix A is parameterized by 13 predicted scalars per patch: a translation vector $\mathbf{t} \in \mathbb{R}^3$, a rotation matrix $\mathrm{R} \in \mathrm{SO}(3)$ derived from a 6D continuous representation (Zhou et al., 2019) via Gram–Schmidt orthogonalization, a unit normal vector $\mathbf{n}$ normalized to unit length, and the inverse plane distance κ.

Because A is constructed as a rank-one update to a rotation matrix, its exact inverse can be computed analytically using the Sherman–Morrison identity (Sherman & Morrison, 1950):

$$\mathrm{A}^{-1} = \mathrm{R}^\top - \frac{\kappa\,(\mathrm{R}^\top\mathbf{t})(\mathbf{n}^\top\mathrm{R}^\top)}{1 + \kappa\,\mathbf{n}^\top\mathrm{R}^\top\mathbf{t}}. \tag{6}$$

This identity allows both A and A^{-1} to be computed efficiently in a single forward pass from the same parameters, completely bypassing the need for explicit matrix inversion routines. The

denominator in Eq. 6 is equal to $\det \mathrm{A}$ and vanishes exactly when the camera center lies on the predicted plane. To ensure the transformation remains numerically well-conditioned, we clamp $\mathbf{t}$, κ, and this denominator (see Appendix A.5 for details).

3.3 Instantiation

Predicted Instance. The patch-specific transformation described in Sec. 3.2 is constructed from $(\mathrm{R}, \mathbf{t}, \mathbf{n}, \kappa)$. These parameters are predicted by a lightweight regression head that maps patch features to the components of A (13 scalar values in the general case). The input features can be extracted from a predictor trained from scratch (acting as a learnable positional encoding), a pretrained 2D backbone (e.g., DINO), or a 3D-aware backbone (e.g., VGGT). Our primary implementation regresses these parameters from VGGT (Wang et al., 2025) features; VGGT's pretraining on geometric prediction tasks provides a strong inductive bias for local surface estimation. In Sec. 4, we compare this choice against a 2D self-supervised backbone and a randomly initialized predictor, keeping the encoding formulation otherwise unchanged. Note that these geometric parameters are predicted exclusively for observed input views.

Calibrated Instance. When camera calibration is available, we retain PRoPE's projection P_i and predict only a local plane per patch, represented by a frame in world coordinates:

$$\mathrm{F}_{i,u} = \begin{bmatrix} \mathrm{R}_{\mathbf{n}_{i,u}} & z_{i,u}\mathbf{n}_{i,u} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \mathrm{H}^{\mathrm{cal}}_{i,u} = \mathrm{P}_i \mathrm{F}_{i,u}, \tag{7}$$

where $\mathrm{R}_{\mathbf{n}}$ is the minimal rotation aligning $\mathbf{e}_3$ with $\mathbf{n}$, and $z_{i,u}$ is the signed orthogonal plane offset. Unlike the predicted 3×3 homography-structured transform, this variant uses a 4×4 composition of camera projection and local frame. Applying the same relative-attention rule (Eq. 4) to 4-dimensional feature blocks yields the relative factor $\mathrm{P}_i(\mathrm{F}_{i,u}\mathrm{F}_{j,v}^{-1})\mathrm{P}_j^{-1}$, combining relative local-frame geometry with calibrated camera projections. When all patches share a common frame, this reduces to $\mathrm{P}_i\mathrm{P}_j^{-1}$, which is the same as PRoPE.

4 Experiments

We first evaluate HRoPE on novel view synthesis, comparing it with existing geometric positional encodings within the LVSM (Jin et al., 2025) framework (Sec. 4.1). We then ablate the transformation parameterization and parameter source (Sec. 4.2.1), and analyze the role and quality of the predicted local geometry (Sec. 4.2.2). Additional experiments evaluate generalization under shifts in camera intrinsics and view sampling (Appendix A.6), as well as transfer to a multi-view diffusion model and to the visual tokens of a spatial reasoning MLLM (Appendix A.7).

4.1 Novel View Synthesis

Experimental Setup. We evaluate HRoPE on novel view synthesis by applying it to the decoder-only LVSM (Jin et al., 2025) architecture. Each model takes two reference views as input and synthesizes a target view. Following PRoPE (Li et al., 2025), we train downsized LVSM backbone variants (∼25M parameters) with different positional encoding methods. We train and evaluate the models separately on CO3Dv2 (Reizenstein et al., 2021), RealEstate10K (Zhou et al., 2018), and Objaverse (Deitke et al., 2023), and report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). To evaluate scaling with the number of reference views, we additionally follow the RayZer (Jiang et al., 2025) protocol and train a larger LVSM backbone variant (∼53M parameters) on DL3DV (Ling et al., 2024). We evaluate the model with 2, 4, and 8 reference views on the held-out DL3DV benchmark.

Results. Table 1 shows that the VGGT-based HRoPE outperforms baseline PEs on CO3Dv2, RealEstate10K, and Objaverse. Relative to PRoPE, it gains 0.53, 1.27, and 1.03 dB in PSNR, respectively, with higher SSIM and lower LPIPS. On DL3DV, HRoPE also outperforms PRoPE with 2, 4, and 8 reference views (Table 2). PSNR and SSIM gains are largest with two views, suggesting greater benefits under sparse observations. Qualitative comparisons on RealEstate10K show that HRoPE better preserves object boundaries and fine details (Fig. 3).

Table 1: Novel view synthesis on CO3Dv2, RealEstate10K, and Objaverse. HRoPE predicts per-patch transformation parameters from image features rather than using calibrated input-camera projections. Best in green, second in blue.

Method	CO3Dv2			RealEstate10K			Objaverse		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LVSM (Jin et al., 2025)	18.617	0.601	0.618	25.282	0.802	0.130	29.676	0.923	0.098
GTA (Miyato et al., 2024)	19.482	0.612	0.573	26.076	0.824	0.110	30.500	0.929	0.079
PRoPE (Li et al., 2025)	19.495	0.626	0.583	26.142	0.826	0.111	30.588	0.929	0.078
RayRoPE (Wu et al., 2026)	18.948	0.617	0.601	26.348	0.830	0.107	30.871	0.932	0.079
URoPE (Xie et al., 2026)	-	-	-	26.836	0.838	0.099	30.890	0.930	0.073
HRoPE (w. Learnable)	19.612	0.630	0.549	26.419	0.831	0.105	30.879	0.932	0.071
HRoPE (w. DINOv2)	19.732	0.635	0.519	26.701	0.833	0.099	31.365	0.934	0.060
HRoPE (w. VGGT)	20.025	0.647	0.462	27.410	0.848	0.091	31.622	0.937	0.057

Table 2: Scaling on the DL3DV benchmark with a 53M LVSM under 2-, 4-, and 8-view inputs. HRoPE consistently improves over PRoPE across all three input-view settings. Best in green.

Method	2 views			4 views			8 views		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LVSM (Jin et al., 2025)	17.718	0.459	0.474	19.331	0.571	0.317	20.441	0.615	0.273
PRoPE (Li et al., 2025)	18.574	0.503	0.459	19.957	0.602	0.289	20.653	0.621	0.271
HRoPE	19.284	0.544	0.447	20.373	0.624	0.270	20.940	0.635	0.257

4.2 Ablation Studies and Analysis

We investigate the design and behavior of HRoPE from two perspectives. First, we compare transformation parameterizations and feature sources for parameter prediction (Sec. 4.2.1). Second, we examine the contribution and quality of local geometry through a calibrated-camera comparison, qualitative visualizations, and an oracle geometry experiment (Sec. 4.2.2).

4.2.1 Parameterization and Feature Sources

Transformation Parameterization. Keeping the overall LVSM architecture unchanged, we compare transformations with different geometric structures and degrees of freedom: a depth term ("Depth"), a rotation ($\mathrm{SO}(3)$), the calibrated $P \cdot F$ transformation, a rigid transformation ($\mathrm{SE}(3)$), the homography parameterization used by HRoPE, and an unconstrained 4×4 matrix ("Full"). Fig. 4(a) compares these variants against the LVSM baseline ("None"). Whether using a predictor trained from scratch or frozen VGGT features, HRoPE achieves the highest PSNR among the evaluated transformations. In both settings, it outperforms the unconstrained 4×4 transformation despite using fewer degrees of freedom. This suggests that the homography structure provides an effective geometric inductive bias without requiring a fully unconstrained transformation. We additionally compare the homography encoding with a 768-dimensional additive encoding constructed from the same VGGT features. The homography formulation performs better despite using the same feature source, indicating that the improvement is not solely due to access to pretrained VGGT features.

Parameter Source. We next fix the homography parameterization and vary the feature source used to predict its parameters. We compare a learnable predictor trained from scratch, DINOv2 (Oquab et al., 2024), DepthAnything3 (Lin et al., 2025), VGGT (Wang et al., 2025), and VGGT-Ω (Wang et al., 2026). Fig. 4(b) shows that the proposed parameterization improves over PRoPE even when its parameters are predicted from scratch or from DINOv2 features. Notably, DINOv2† retains a 2D architecture, but its features are fine-tuned within VGGT's 3D pretraining pipeline and therefore acquire implicit 3D awareness, which helps explain their stronger performance than that of off-the-shelf DINOv2. Performance further improves with stronger pretrained feature extractors, with the geometry-aware VGGT backbone achieving the highest PSNR. These results show that the gains of HRoPE do not depend on a particular pretrained backbone, while more informative geometric features provide additional improvements.

4.2.2 Analysis of Local Surface Geometry

Local Geometry with Calibrated Cameras. To isolate the effect of local geometry, we start from PRoPE's GT camera projection P and compose it with the per-patch tangent-frame transformation F

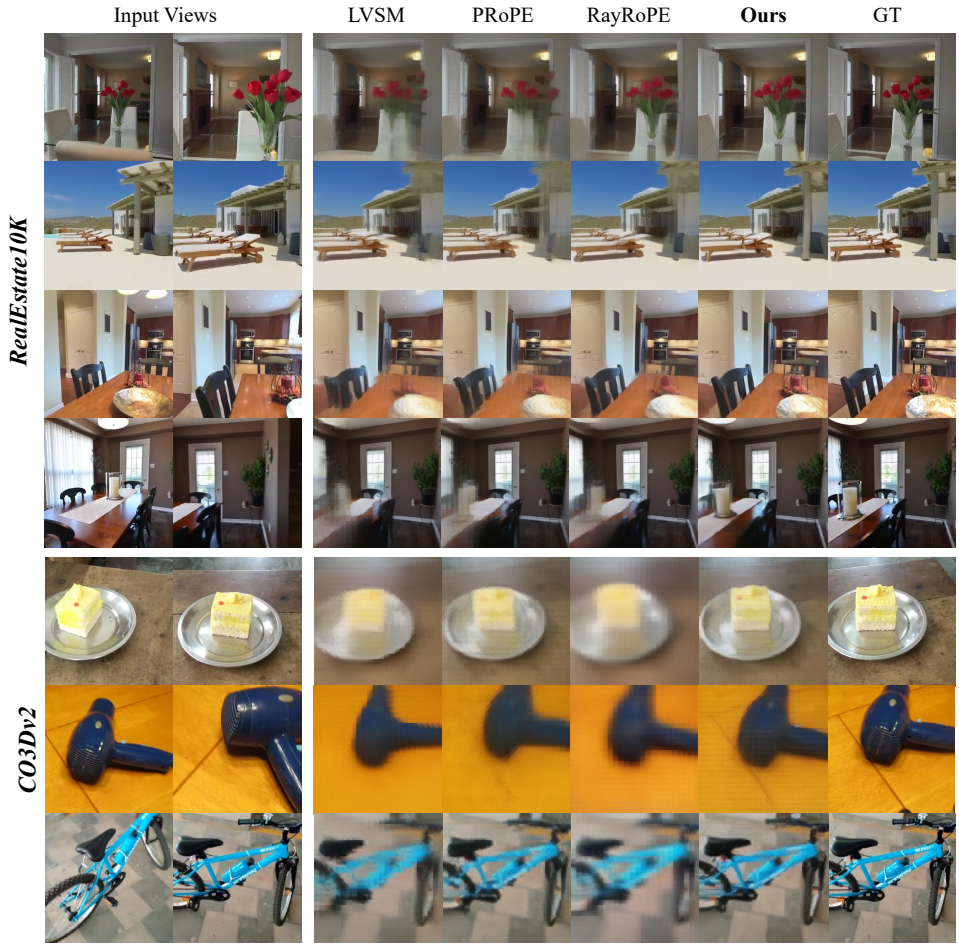


Figure 3: **Qualitative novel-view synthesis on RealEstate10K and CO3Dv2.** Each row shows one scene or object. From left to right, the columns contain two input views, predictions from LVSM, PRoPE, RayRoPE, and HRoPE, and the ground-truth target view (GT).

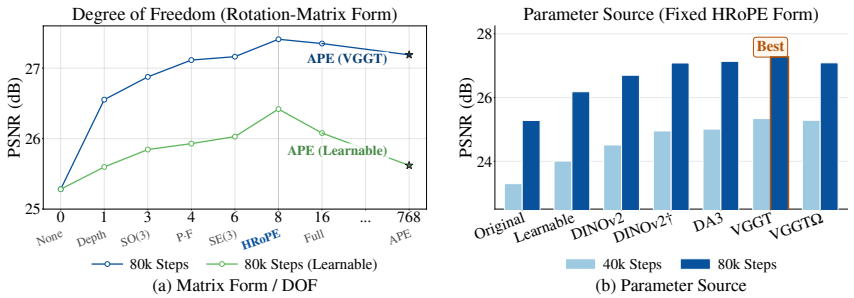


Figure 4: **Ablations on RealEstate10K, measured by PSNR. (a)** Transformation parameterizations using frozen VGGT features (blue) or a predictor trained from scratch (green), evaluated after 80k training steps. The x-axis labels denote the LVSM baseline (None), a depth term (Depth), a rotation ($\mathrm{SO}(3)$), the calibrated projection composed with a per-patch tangent-frame transform ($P \cdot F$), a rigid transformation ($\mathrm{SE}(3)$), our homography parameterization (HRoPE), and an unconstrained 4×4 matrix (Full). Stars denote the corresponding 768-dimensional additive positional encodings (APE). **(b)** Feature sources with the HRoPE parameterization fixed, evaluated after 40k and 80k training steps. DINOv2† denotes the fine-tuned DINOv2 features from VGGT.

derived from predicted plane parameters, forming the calibrated $P \cdot F$ variant. We evaluate PRoPE, $P \cdot F$, and the fully predicted HRoPE on RealEstate10K using the setup in Sec. 4.1. As shown in Table 3, adding F improves PRoPE from 26.142 to 27.115 in PSNR and from 0.826 to 0.840

Table 3: **Local surface geometry on RealEstate10K (Left) & Oracle geometry on ASE (Right).** On RealEstate10K, PRoPE ($P{\cdot}F$) and HRoPE both use VGGT features for parameter prediction. On ASE, HRoPE uses either VGGT-predicted parameters or ground-truth camera and plane parameters. Best and second-best results are highlighted in green and blue.

Method	PSNR↑	SSIM↑	LPIPS↓
PRoPE	26.142	0.826	0.111
PRoPE ($P \cdot F$)	27.115	0.840	0.104
HRoPE	27.410	0.848	0.091

Method	PSNR↑	SSIM↑	LPIPS↓
PRoPE	27.829	0.802	0.264
HRoPE (w. VGGT)	28.555	0.809	0.198
HRoPE (w. GT)	28.751	0.820	0.192

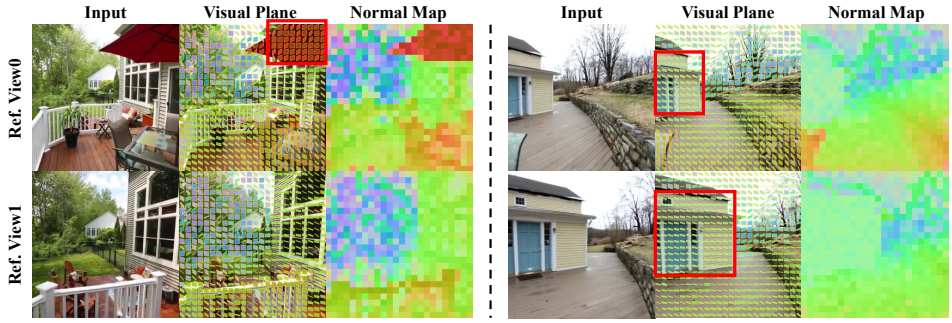


Figure 5: **Predicted local planes on RealEstate10K.** Two scenes are shown side by side; within each scene, the two rows correspond to the reference views. For each view, the columns show the input image, predicted plane orientations overlaid on the image (Visual Plane), and per-patch normals encoded as a color map (Normal Map). The red boxes mark the same approximately planar surface across views, where the predicted orientations remain locally coherent and cross-view consistent.

in SSIM, while reducing LPIPS from 0.111 to 0.104. Because PRoPE and $P \cdot F$ share the same projection P, these gains support the contribution of the local surface transformation. HRoPE with predicted transformations further improves all three metrics over the calibrated $P \cdot F$ variant.

Predicted Local Geometry. Fig. 5 shows predicted orientations for the two input views using overlays and per-patch normal maps. Patches belonging to approximately planar regions exhibit similar predicted orientations, while corresponding regions across reference views show qualitatively consistent surface orientations when expressed in a common coordinate frame. The visualizations suggest coherent local structure without direct plane supervision.

Oracle Geometry. Finally, we test whether more accurate, physically grounded parameters further improve HRoPE. We use Aria Synthetic Environments (Avetisyan et al., 2024), which provides ground-truth depth, surface normals, and camera parameters, and construct the transformation A in Eq. 5 using either VGGT-predicted parameters or ground-truth geometry. As shown in Table 3, replacing the VGGT predictions with ground-truth geometry improves PSNR from 28.555 to 28.751, while also improving SSIM and LPIPS. This result suggests that HRoPE benefits from more accurate geometric inputs under the evaluated oracle protocol.

5 Conclusion

In this paper, we investigated local surface geometry as a prior for positional encoding in multi-view transformers. Inspired by the slanted support windows of PatchMatch Stereo, we introduced HRoPE, which uses the structure of plane-induced homographies to parameterize per-patch transformations and modulate attention through their relative composition. The transformation parameters are predicted from patch features and learned jointly with the view-synthesis model. Experiments on RealEstate10K, CO3Dv2, Objaverse, and DL3DV demonstrate improved novel view synthesis over the evaluated positional encodings. Ablations support the contribution of the transformation structure, with gains observed for predictors trained from scratch and further improvements from pretrained features. A calibrated extension also improves PRoPE by incorporating local planar frames while retaining its camera projections. These findings support local surface geometry as a useful inductive bias for multi-view attention.

AI USE STATEMENT

During the preparation of this manuscript, we used ChatGPT solely to improve language, grammar, and readability. We did not use generative AI tools to generate or process datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims or proofs; propose or refine hypotheses; design research methods or experiments; implement methods; translate content; conduct qualitative or thematic analyses; or interpret results. The authors reviewed and revised all AI-assisted edits to ensure accuracy and preserve the intended meaning, and take full responsibility for the final content of this manuscript.

REPRODUCIBILITY STATEMENT

We describe the transformation parameterization, closed-form inverse, and attention formulation in Sec. 3.2, and parameter prediction and the calibrated variant in Sec. 3.3. Additional derivations, attention integration details, and numerical safeguards are provided in Appendices A.2, A.4, and A.5. The datasets, backbone configurations, and evaluation metrics are specified in Sec. 4.1, with ablation settings and local-geometry comparisons in Secs. 4.2.1 and 4.2.2. Protocols for out-of-distribution and transfer evaluations are described in Appendices A.6 and A.7. We will release our code upon acceptance.

REFERENCES

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *ECCV*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3), 2009.

Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, 2011.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024a.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos. In *ICLR*, 2025.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024b.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *CVPR*, 2026.

Ryan Griffiths and Donald G Dansereau. Rore: Rotary ray embedding for generalised multi-modal scene understanding. In *ICLR*, 2026.

Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *CVPR*, 2020.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *ICCV*, 2025.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *ICLR*, 2025.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *CVPR*, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. Rerope: Repurposing rope for relative camera control. *arXiv preprint arXiv:2602.08068*, 2026.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *NeurIPS*, 38, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022.

Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Technical Report RR-6303, INRIA, 2007.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *ICLR*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. *CVPR*, 2022.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018.

Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1), 1950.

Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 34, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *CVPR*, 2022.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021.

Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, 2021a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 2004.

Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *ICCV*, 2023.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. Rayrope: Projective ray positional encoding for multi-view attention. In *ECCV*, 2026.

Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. Urope: Universal relative position embedding across geometric spaces. *ECCV*, 2026.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *CVPR*, 2025.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018.

Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. In *3DV*, 2024.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *CVPR*, 2026.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024a.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024c.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

Qi Zuo, Xiaodong Gu, Yuan Dong, Zhengyi Zhao, Weihao Yuan, Lingteng Qiu, Liefeng Bo, and Zilong Dong. High-fidelity 3d textured shapes generation by sparse encoding and adversarial decoding. In *ECCV*, 2024.

A Appendix

We first expand the notation and derivations behind the geometric positional encodings summarized in Sec. 3.1 (Appendix A.1). Appendix A.2 then derives the plane-induced homography that computes a pixel in view i directly from a pixel in view j, and explains why a purely projective encoding cannot express this correspondence. Appendix A.3 then writes out in full the block-diagonal operator that applies this homography across the head dimension.

A.1 Background on Geometric Positional Encodings

Notation. For a patch token u in view i, let $\mathbf{q}_{i,u}$, $\mathbf{k}_{i,u}$, and $\mathbf{v}_{i,u}$ denote its query, key, and value vectors. The calibrated camera has intrinsics K_i and a world-to-camera transform

$$E_i = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \mathbf{X}_i = R_i \mathbf{X} + \mathbf{t}_i. \tag{8}$$

Here, $R_i \in \mathrm{SO}(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$ describe the camera pose. These calibrated quantities are distinct from the patch-specific parameters predicted by HRoPE in Sec. 3.3.

Rotary positional encoding. Let $\rho(u)$ denote RoPE's block-diagonal rotation for position u. Applying it to a query at u and a key at v gives

$$\big(\rho(u)\mathbf{q}\big)^\top \big(\rho(v)\mathbf{k}\big) = \mathbf{q}^\top \rho(u)^\top \rho(v)\mathbf{k} = \mathbf{q}^\top \rho(v-u)\mathbf{k}, \tag{9}$$

so the attention score depends on relative rather than absolute position. The usual 2D extension applies this construction separately along the two image axes.

Camera-relative encodings. GTA (Miyato et al., 2024) and CaPE (Kong et al., 2024) extend the same principle from image-grid offsets to camera motion: their feature transformations make the interaction depend on the relative extrinsic transform $E_i E_j^{-1}$, with method-specific representations of its action on features. PRoPE (Li et al., 2025) additionally includes intrinsics by lifting K_i to homogeneous coordinates,

$$\bar{K}_i = \begin{bmatrix} K_i & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad P_i = \bar{K}_i E_i, \qquad P_i P_j^{-1} = \bar{K}_i E_i E_j^{-1} \bar{K}_j^{-1}. \tag{10}$$

It applies $P_i^\top$ to queries and P_j^{-1} to keys, yielding the relative projective factor. This factor is invariant to the choice of world frame: replacing every E_i by $E_i T^{-1}$ for a shared rigid transform T leaves $P_i T^{-1}(P_j T^{-1})^{-1} = P_i P_j^{-1}$ unchanged.

A.2 Derivation of the Plane-Induced Homography

Let view j be the reference camera, with the world frame aligned to it, so a 3D point $\mathbf{X}_j \in \mathbb{R}^3$ (in camera-j coordinates) projects to the homogeneous pixel $\tilde{\mathbf{x}}_j$ of view j as

$$z_j\, \tilde{\mathbf{x}}_j = K_j\, \mathbf{X}_j \quad \Longleftrightarrow \quad \mathbf{X}_j = z_j\, K_j^{-1} \tilde{\mathbf{x}}_j, \tag{11}$$

where $z_j > 0$ is the (unknown) depth along the optical axis of camera-j. Let $(R_{i\leftarrow j}, \mathbf{t}_{i\leftarrow j})$ be the rigid transform from the camera-j frame to the camera-i frame; the same point projects to view i as

$$z_i\, \tilde{\mathbf{x}}_i = K_i\, (R_{i\leftarrow j}\, \mathbf{X}_j + \mathbf{t}_{i\leftarrow j}). \tag{12}$$

Suppose that $\mathbf{X}_j$ lies on a local plane with unit normal $\mathbf{n}_j$ at orthogonal distance $d_j > 0$ in the camera-j frame,

$$\mathbf{n}_j^\top \mathbf{X}_j = d_j \quad \Longleftrightarrow \quad \tfrac{1}{d_j}\, \mathbf{n}_j^\top \mathbf{X}_j = 1. \tag{13}$$

The plane assumption is what removes the unknown depth. Substituting $\mathbf{X}_j = z_j K_j^{-1} \tilde{\mathbf{x}}_j$ from Eq. 11 into Eq. 13 determines the depth at the ray–plane intersection,

$$z_j = \frac{d_j}{\mathbf{n}_j^\top K_j^{-1} \tilde{\mathbf{x}}_j}. \tag{14}$$

The key algebraic step is to rewrite the translation in Eq. 12 using the plane identity $\frac{1}{d_j}\mathbf{n}_j^\top \mathbf{X}_j = 1$, so that $\mathbf{t}_{i\leftarrow j} = \mathbf{t}_{i\leftarrow j}\,(\frac{1}{d_j}\mathbf{n}_j^\top \mathbf{X}_j)$ and the affine map becomes linear in $\mathbf{X}_j$:

$$R_{i\leftarrow j}\,\mathbf{X}_j + \mathbf{t}_{i\leftarrow j} = R_{i\leftarrow j}\,\mathbf{X}_j + \mathbf{t}_{i\leftarrow j}\left(\tfrac{1}{d_j}\mathbf{n}_j^\top \mathbf{X}_j\right) = \Big(\underbrace{R_{i\leftarrow j} + \tfrac{1}{d_j}\,\mathbf{t}_{i\leftarrow j}\,\mathbf{n}_j^\top}_{A}\Big)\mathbf{X}_j. \tag{15}$$

Combining Eqs. 11, 12, and 15 and eliminating the depth-dependent scale yields the plane-induced homography between the two pixels,

$$\tilde{\mathbf{x}}_i \;\sim\; H_{i\leftarrow j}\,\tilde{\mathbf{x}}_j, \qquad H_{i\leftarrow j} \;=\; K_i\Big(R_{i\leftarrow j} + \tfrac{1}{d_j}\,\mathbf{t}_{i\leftarrow j}\,\mathbf{n}_j^\top\Big)K_j^{-1} \;=\; K_i\,A\,K_j^{-1}, \tag{16}$$

Here, $\sim$ denotes equality up to a nonzero scale. For points in front of both cameras, $z_i, z_j > 0$, and the scale z_j/z_i is positive. The inner factor has the same rotation-plus-rank-one structure as the parameterization in Eq. 5, with $\kappa = 1/d_j$. This connection motivates the use of local plane parameters in our learned per-patch transformations. (The minus-sign form $R_{i\leftarrow j} - \frac{1}{d_j}\mathbf{t}_{i\leftarrow j}\,\mathbf{n}_j^\top$ arises under the alternative plane convention $\mathbf{n}_j^\top \mathbf{X}_j + d_j = 0$.)

Eq. 16 defines a pixel-to-pixel mapping conditioned on a local plane. In comparison, the relative projection $P_i P_j^{-1}$ used by PRoPE depends only on camera parameters and does not generally determine a unique correspondence between image pixels. A source pixel $\tilde{\mathbf{x}}_j$ specifies a viewing ray $\mathbf{X}_j = z_j K_j^{-1}\tilde{\mathbf{x}}_j$, but its depth z_j remains unknown. For a nonzero camera baseline, this depth ambiguity generally constrains the corresponding pixel in view i to an epipolar line. Given a plane hypothesis $(\mathbf{n}_j, d_j)$, Eq. 14 determines the depth at the ray–plane intersection, provided that the intersection is finite and lies in front of the source camera. Substituting this depth into the projection equation yields the homography in Eq. 16, which maps points on the plane between the two views. Unlike the camera-only relative projection used by PRoPE, a local plane resolves the depth ambiguity along a viewing ray and induces a pixel-to-pixel homography on that plane. This motivates HRoPE's incorporation of local surface geometry into its per-patch transformations. The same principle underlies the slanted support windows in PatchMatch Stereo.

Global-frame invariance. In the image-based instantiation of HRoPE, the per-patch transforms $A_{i,u}$ are predicted from image features without using externally specified world-frame extrinsics. Changing the world coordinate frame re-expresses the cameras and scene geometry but leaves the images and intrinsics unchanged. Consequently, the predicted transforms and the relative factors $H_{i,u}H_{j,v}^{-1}$ remain unchanged. This establishes invariance of the predicted geometric encoding; invariance of the full attention computation additionally requires its input features to be independent of the world-frame choice.

For transforms constructed from explicit camera and plane geometry, the scope is more restricted. When both factors describe a shared physical plane, consistently changing coordinates gives $H_i' = H_i G^{-1}$ and $H_j' = H_j G^{-1}$ for a common invertible matrix G, so $H_i'(H_j')^{-1} = H_i H_j^{-1}$. For different patch planes, the induced right multiplier can depend on the plane, and this cancellation is not generally guaranteed. Thus, the relative-product form alone does not establish world-frame invariance for arbitrary geometry-derived transforms. In our GT instantiation, both camera and plane parameters are expressed in the first reference camera's coordinate frame and normalized to a common scale, removing dependence on the external world coordinate frame.

A.3 The full block-diagonal encoding matrix

Section 3.2 applies the encoding "block-diagonally to the features"; here we write out that block-diagonal operator in full. Let the per-head feature dimension be D. For a token in view i we split its D-dimensional query/key/value vector into three contiguous parts of sizes $D/2$, $D/4$, $D/4$. The first $D/2$ coordinates carry the geometric homography and the remaining $D/2$ carry a standard 2D rotary encoding of the patch's pixel coordinates (x, y), with $D/8$ frequencies for each axis. The

Algorithm 1 HRoPE multi-view self-attention layer (single head).

Require: queries/keys/values $Q, K, V \in \mathbb{R}^{(NP)\times D}$; per-patch matrices $\{A_{i,u}\}$, intrinsics $\{K_i\}$; RoPE coeffs (Φ_x, Φ_y)
Notation: $\mathcal{H}_{i,u}$ is the block-diagonal operator of Eq. 17 built from the per-patch 3×3 matrix $H_{i,u}$ and (Φ_x, Φ_y). $\mathrm{BD}(\mathcal{H}, f)$ applies the corresponding per-token operators to the features f.

for each token u in view i **do** ▷ build the per-token geometric factor
  if i is an input view **then** $H_{i,u} \leftarrow K_i\, A_{i,u}$
  else $H_{i,u} \leftarrow K_i$ ▷ target view: projective fallback, $A_{i,u} = I$
  end if
end for
$\tilde{Q} \leftarrow \mathrm{BD}(\mathcal{H}^\top, Q)$ ▷ query side: $H_{i,u}^\top$
$\tilde{K} \leftarrow \mathrm{BD}(\mathcal{H}^{-1}, K)$ ▷ key side: $H_{i,u}^{-1}$ via Sherman–Morrison
$\tilde{V} \leftarrow \mathrm{BD}(\mathcal{H}^{-1}, V)$
$S \leftarrow \tilde{Q}\tilde{K}^\top/\sqrt{D}$ ▷ logit $= \mathbf{q}^\top H_{i,u} H_{j,v}^{-1}\mathbf{k}$, Eq. 4
$O \leftarrow \mathrm{softmax}(S)\,\tilde{V}$
$O \leftarrow \mathrm{BD}(\mathcal{H}, O)$ ▷ back to query frame
return O

block-diagonal encoding matrix for patch (i, u) is

$$\mathcal{H}_{i,u} = \underbrace{H_{i,u}^{\oplus D/6}}_{\text{geometry: } D/2 \text{ dims}} \oplus \underbrace{\Phi_x}_{D/4} \oplus \underbrace{\Phi_y}_{D/4} = \begin{bmatrix} H_{i,u} & & & & & \\ & \ddots & & & & \\ & & H_{i,u} & & & \\ & & & \Phi_x & & \\ & & & & \Phi_y & \end{bmatrix}, \tag{17}$$

Here, $\oplus$ denotes a direct sum and $H^{\oplus r}$ contains r copies of the 3×3 matrix H. The geometric block is $D/6$ copies of the single 3×3 matrix $H_{i,u}$ (here $D/6 = 8$), because each copy acts on a 3-vector and $3\cdot(D/6) = D/2$; this is what makes "apply H to the feature" well-defined when $D/2 > 3$. The two rotary blocks are the usual 2D-RoPE rotations

$$\Phi_x = \bigoplus_{\ell=1}^{D/8} \begin{bmatrix} \cos(x\,\omega_\ell) & -\sin(x\,\omega_\ell) \\ \sin(x\,\omega_\ell) & \cos(x\,\omega_\ell) \end{bmatrix}, \qquad \omega_\ell = \theta^{-2\ell/(D/2)}, \tag{18}$$

and Φ_y is identical with the patch row coordinate y; together they encode the in-image patch position, complementing the cross-view geometry in the first block. Attention uses $\mathcal{H}_{i,u}^\top$ on the query and $\mathcal{H}_{j,v}^{-1}$ on the key, so that the geometric blocks contract to $H_{i,u}H_{j,v}^{-1}$ (Eq. 4) and the rotary blocks contract to a relative rotation in (x, y), recovering the standard 2D-RoPE relative property. For target-view tokens, which carry no predicted plane, the geometric block reduces to K_i alone (i.e. $A_{i,u} = I$), encoding only camera intrinsics while retaining the 2D rotary blocks.

A.4 Integrating HRoPE into multi-view attention

HRoPE modifies multi-view self-attention through token-specific linear operators while leaving the other transformer components unchanged. For N views with P patches each, the layer receives NP tokens, per-patch transforms $\{A_{i,u}\}$, and camera intrinsics $\{K_i\}$. We construct $\mathcal{H}_{i,u}$ as in Eq. 17 and apply $\mathcal{H}_{i,u}^\top$ to each query and $\mathcal{H}_{j,v}^{-1}$ to each key and value. The resulting attention logit for tokens (i, u) and (j, v) contains the relative operator $\mathcal{H}_{i,u}\mathcal{H}_{j,v}^{-1}$. After attention aggregation, the output is transformed by $\mathcal{H}_{i,u}$, expressing the aggregated features in the query token's coordinate system. The inverse of $A_{i,u} = R_{i,u} + \kappa_{i,u}\mathbf{t}_{i,u}\mathbf{n}_{i,u}^\top$ is computed in closed form using Eq. 6. Fig. 6 illustrates these operations, and Algorithm 1 specifies the forward pass.

A.5 Additional experiment details

Datasets. We use five datasets in our experiments: RealEstate10K, CO3Dv2, Objaverse, DL3DV, and Aria Synthetic Environments (ASE). We describe the data splits, preprocessing, and frame sampling for each dataset below.

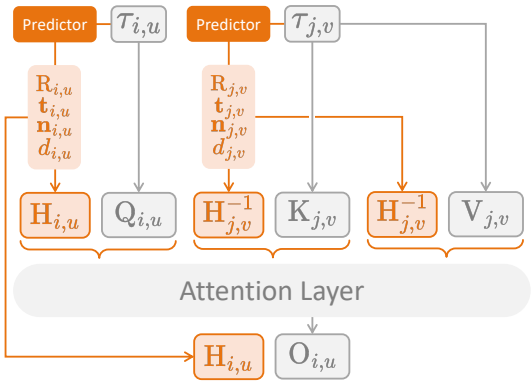


Figure 6: **Applying HRoPE to Multi-view Attention.** Here, $\tau_{i,u}$ denotes the token corresponding to image patch u in view i. A predictor maps each patch token $\tau_{i,u}$ to the transformation parameters $(R_{i,u}, \mathbf{t}_{i,u}, \mathbf{n}_{i,u}, \kappa_{i,u})$, from which $H_{i,u} = K_i A_{i,u}$ is constructed. Queries are transformed by $\mathcal{H}^{\top}_{i,u}$, while keys and values are transformed by $\mathcal{H}^{-1}_{j,v}$. The aggregated value is subsequently mapped to the query token's frame by $\mathcal{H}_{i,u}$. Only one query–key pair is shown for clarity.

- **RealEstate10K** (Zhou et al., 2018): frames are extracted from the source YouTube videos at the official timestamps, with the given per-sequence intrinsics and per-frame world-to-camera poses converted into the common format above. During training we sample input/target frames with an index gap in $[25, 100]$.
- **CO3Dv2** (Reizenstein et al., 2021): we select a higher-quality subset and split it into a 31k-scene training set and a 1k-scene test set, both covering all 52 categories uniformly. We use a frame gap in $[25, 50]$ during training.
- **Objaverse** (Deitke et al., 2023): from the G-buffer Objaverse set (Zuo et al., 2024) we keep objects with fewer than 20k faces (96k objects) and render 30 views each at 336 resolution in Blender. The camera looks down at the object from a fixed pitch of about $45°$, at radius 2.0 and field of view $40°$; for each object we sample views uniformly within a randomly placed $180°$ azimuth window and with random perturbations to azimuth and elevation. We render on a transparent background composited onto white, and sample views with an angular gap in $[30°, 60°]$ during training.
- **DL3DV** (Ling et al., 2024): we discard portrait-orientation videos and train on all remaining 10k scenes. For evaluation we use the RayZer benchmark (the 10k scenes not included in the benchmark are used for training) (Jiang et al., 2025). Training frame gaps follow the input regime: $[25, 50]$ for the 2- and 4-view models and $[25, 100]$ for the 8-view model.
- **Aria Synthetic Environments** (Avetisyan et al., 2024): we split the scenes at random into a 95k-scene training set and a 5k-scene test set, and use a frame gap in $[8, 12]$ during training. This synthetic dataset renders a depth map for every frame, from which we derive the ground-truth plane parameters used in Sec. 4.2.2. Since a patch spans many pixels, we fit one plane per 14×14 patch by least squares on inverse depth rather than reading a single depth sample, writing it as $\mathbf{n}_c^{\top}\mathbf{X}_c = d_c$ in the camera frame and fixing $d_c > 0$ so that normals are oriented consistently across views. Patches that are more than half invalid, or whose median relative residual exceeds 5% and are therefore poorly described by a single plane, fall back to a fronto-parallel plane at their median depth.

Training configuration. We train every model on four NVIDIA RTX 4090 (48 GB) GPUs with a per-GPU batch size of 16, using a cosine learning-rate schedule with peak rate 4×10^{-4} and a

Table 4: Training cost of the parameter-source variants (Sec. 4.2.1), all under the same 80k-iteration schedule; PSNR is the final RealEstate10K score. DINOv2† is the fine-tuned DINOv2 taken from inside VGGT. Best in **bold**, second underlined.

Source	PSNR ↑	GPU-hours
Original (LVSM)	25.282	33
Learnable	26.419	33
DINOv2 (Oquab et al., 2024)	26.701	46
DINOv2† (Oquab et al., 2024)	27.085	72
DA3 (Lin et al., 2025)	<u>27.137</u>	48
VGGT (Wang et al., 2025)	**27.410**	76
VGGT-Ω (Wang et al., 2026)	27.090	64

warmup of 2500 iterations. All scene-level models are trained for 80k iterations at a fixed resolution of 336×336 obtained by center cropping. Camera poses are normalized following the convention that matches the input regime. With few input views (e.g. 2 views) we follow PRoPE (Li et al., 2025): the first reference camera is set to the identity (its frame becomes the world origin) and all translations are scaled by the distance between the first and second reference cameras, so that this distance is unit length. With many input views we follow the original LVSM (Jin et al., 2025) convention: the scene is translated and rotated to align with the mean camera center and mean viewing direction, then rescaled by the maximum absolute camera-center coordinate times a fixed factor of 1.35 to a common scale. Our objective combines a photometric term with a perceptual loss; we weight the perceptual term at 0.5 for scene datasets and lower it to 0.1 for the object dataset.

Numerical stability of the predicted transformation. To stabilize training, we clamp the predicted translation $\mathbf{t}$ and inverse plane distance κ to $[-0.5, 0.5]$, normalize $\mathbf{n}$, and construct R to be orthonormal. This bounds the rank-one update in $\mathrm{A} = \mathrm{R} + \kappa \mathbf{t}\mathbf{n}^\top$ and guarantees $\det \mathrm{A} \geq 1 - \sqrt{3}/4 \approx 0.57$. We additionally clamp the Sherman–Morrison denominator to 0.1 and the entries of A and A^{-1} to $[-5, 5]$. Both matrices are computed in `float32` and then cast to the training precision.

Training cost. Table 4 reports the final RealEstate10K PSNR and training cost of the parameter-source variants, all trained for 80k iterations. At the same cost of 33 GPU-hours, the learned predictor improves PSNR from 25.282 to 26.419, showing that the gain does not depend solely on an expensive pretrained source. Among the pretrained sources, DA3 achieves 27.137 PSNR in 48 GPU-hours, compared with 27.410 PSNR in 76 GPU-hours for VGGT.

A.6 Out-of-Distribution Generalization

Focal-length Variation. We first evaluate robustness to changes in camera intrinsics by varying the target-camera focal length at test time. Specifically, we scale the focal length from $1\times$ to $5\times$ its original value, where $1\times$ corresponds to the standard evaluation setting. As shown in Fig. 7, HRoPE maintains strong performance as the zoom factor increases and achieves the lowest LPIPS across the evaluated focal lengths. In contrast, the calibrated $P \cdot F$ variant degrades more substantially under large zoom factors.

Target-view Extrapolation. We next extend the target-view sampling interval by 50% at both ends, allowing target views to fall outside the interval bounded by the two reference views. Fig. 7 shows that HRoPE maintains its advantage over the baseline positional encodings as the target view moves beyond the training sampling regime.

Input-view Spacing. We additionally vary the temporal spacing between the two reference frames. We shift the input-frame interval from the training range of $[25, 100]$ to $[75, 150]$ at test time. HRoPE remains robust as the reference views become farther apart and achieves lower LPIPS across the evaluated intervals.

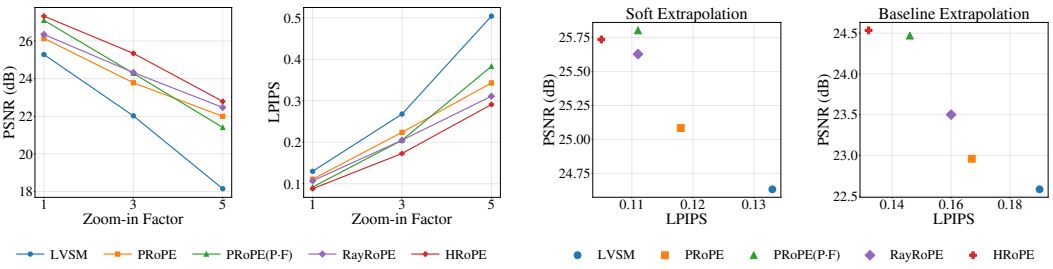


Figure 7: **Out-of-distribution generalization on RealEstate10K. Left pair:** PSNR (↑) and LPIPS (↓) as the unseen target-camera zoom factor increases from $1\times$ to $5\times$. **Right pair:** the PSNR–LPIPS trade-off under soft target-view extrapolation and an enlarged baseline between the two input views.

Table 5: Transferring HRoPE to EscherNet (Kong et al., 2024), an ~ 800M-parameter multi-view diffusion model, on Objaverse. We replace EscherNet's native CaPE encoding with HRoPE and evaluate with 1, 5, and 10 input views. Best in green.

	1 view			5 views			10 views		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
EscherNet + CaPE	17.689	0.858	0.181	22.065	0.899	0.114	23.666	0.914	0.089
EscherNet + HRoPE	19.900	0.883	0.135	23.075	0.911	0.092	24.515	0.923	0.074

A.7 Generalization Beyond Architectures

HRoPE is a positional encoding rather than a feature of any one architecture, so as a positional encoding, HRoPE is architecture-agnostic. We validate this in two directions: a generative multi-view diffusion model, which has a different decoder and a different native encoding, and a spatial multimodal large language model.

Multi-view Diffusion Model. As a first step outside that setting, we transfer HRoPE to a generative architecture with a different decoder and a different native encoding. We test on EscherNet (Kong et al., 2024), a large (~ 800M-parameter) multi-view diffusion model whose native encoding is the camera positional encoding CaPE, by replacing CaPE with HRoPE while leaving the rest of the model and its training recipe untouched. We train on Objaverse (Deitke et al., 2023) (96k objects) following EscherNet's setting, feeding 3 reference views during training, and evaluate the trained models with 1, 5, and 10 input views to probe both sparse- and denser-view regimes.

Table 5 shows that HRoPE improves EscherNet on every metric at every view count, supporting the applicability of the encoding beyond LVSM and feed-forward decoding.

Spatial Reasoning in Multimodal LLMs. We further evaluate how HRoPE enhances the spatial reasoning ability of multimodal LLMs. Visual tokens in these models are indexed only by their 2D raster location within a frame and their ordinal position in the frame sequence, so self-attention has no explicit representation of the 3D geometric relationship between observations of the same surface. Our proposed HRoPE enriches the visual tokens of a pretrained MLLM with surface-aware positional information and cross-view correspondence cues, while modifying only their positional encoding. Concretely, we build on LLaVA-Video-7B (Zhang et al., 2024c), sampling 32 frames per video and processing them with a frozen VGGT (Wang et al., 2025) aggregator in parallel with the model's native vision tower. From VGGT's per-patch features, the prediction head described in Sec. 3.3 regresses $(R_{i,u}, \mathbf{t}_{i,u}, \mathbf{n}_{i,u}, \kappa_{i,u})$, which parameterize the per-patch matrix $A_{i,u}$ in Eq. 5. We then insert the block-diagonal operator in Eq. 17 into the language model's self-attention, applying it only to visual tokens in layers $L-10$ through $L-2$ of the 28-layer backbone ($L = 28$): the geometric halves of each head's query and key for token (i, u) are transformed by $H_{i,u}^{\top}$ and $H_{i,u}^{-1}$, respectively, where $H_{i,u} = K_i A_{i,u}$, while the remaining halves retain the native rotary encoding. In this way, the visual stream of the MLLM is explicitly conditioned on per-patch surface geometry, while text tokens and the temporal ordering of frames are left unchanged, making HRoPE a strictly additive augmentation to the pretrained VLM. Following VLM-3R (Fan et al., 2026), we fine-tune

Table 6: Transferring HRoPE to a video multimodal LLM, evaluated on VSI-Bench (Yang et al., 2025). Our model uses LLaVA-Video-7B-Qwen2 as the backbone with HRoPE applied to the visual tokens, fine-tuned on the VSI-Bench training split at 32 frames; the preceding row reports a fine-tuned 7B video model without any geometric encoding (Fan et al., 2026). Frame budgets for the remaining methods are given in the text. Rank is taken over the overall average. Best in green.

	Numerical question				Multiple-choice question					
Method	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	Avg. ↑	Rank
Proprietary models										
GPT-4o (Hurst et al., 2024)	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5	34.0	9
Gemini-1.5 Pro (Team et al., 2024)	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6	45.4	4
Open-source models										
LongVILA-8B (Chen et al., 2025)	29.1	9.1	16.7	0.0	29.6	30.7	32.5	25.5	21.6	13
LongVA-7B (Zhang et al., 2024b)	38.0	16.6	38.9	22.2	33.1	43.3	25.4	15.7	29.2	12
VILA-1.5-40B (Lin et al., 2024)	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9	31.2	11
Qwen2.5VL-7B (Bai et al., 2025)	40.9	14.8	43.4	10.7	38.6	38.5	33.0	29.8	33.0	10
InternVL2-40B (Chen et al., 2024b)	34.9	26.9	46.5	31.8	42.1	32.2	34.0	39.6	36.0	8
Qwen2.5VL-72B (Bai et al., 2025)	25.1	29.3	54.5	38.8	38.2	37.0	34.0	28.9	37.0	7
LLaVA-OneVision-72B (Li et al., 2024)	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6	40.2	6
LLaVA-Video-72B (Zhang et al., 2024c)	48.9	22.8	57.4	35.3	42.4	36.7	35.0	48.6	40.9	5
Spatial-MLLM-4B (Wu et al., 2025)	65.3	34.8	63.1	45.1	41.3	46.2	33.5	46.3	48.4	3
LLaVA-Video-7B (fine-tuned) (Fan et al., 2026)	70.6	43.6	70.8	63.7	64.9	68.9	40.7	38.5	57.7	2
LLaVA-Video-7B + HRoPE (ours)	70.9	51.6	69.8	65.9	64.1	76.0	46.4	31.6	59.5	1

the model for 3 epochs on its released VSI-Bench training split, which combines spatial question–answer pairs over ScanNet (Dai et al., 2017) scenes with embodied route-planning instances, while keeping VGGT frozen throughout.

We report the evaluation results on VSI-Bench (Yang et al., 2025) in Table 6. Our model and the fine-tuned control take 32 frames per video as input. For Spatial-MLLM (Wu et al., 2025) and the Qwen2.5VL series (Bai et al., 2025), we use 16 frames as input. For the other open-source methods and GPT-4o (Hurst et al., 2024), we follow the VSI-Bench evaluation protocol to select the number of input frames (ranging from 8 to 32 frames). For Gemini-1.5 Pro (Team et al., 2024), it samples video frames at 1 FPS. HRoPE reaches an average of 59.5, ahead of every model in the table and 1.8 points above the same backbone fine-tuned on the identical data without it. The gains concentrate in the categories that require relating a surface across frames, including absolute distance (+8.0), relative direction (+7.1), and route planning (+5.7), while the categories answerable from a single frame barely move, which is what one expects of an encoding that supplies surface orientation rather than semantics. The exception is appearance order, which falls 6.9 points: it is the one purely temporal task in the suite, and HRoPE spends half of each visual token's head dimension on the plane-induced map, weakening the ordinal cue it depends on. Since the language backbone, the fine-tuning data, and the frame budget are held fixed across the two rows, both the improvements and this regression are attributable solely to the positional encoding. This controlled comparison substantiates our central claim: the plane-induced homography benefits self-attention whenever it must establish correspondence between two observations of the same surface, irrespective of whether those observations are rendered, denoised, or sampled from video.

A.8 Additional Qualitative Results

Fig. 8 and 9 provide additional novel-view comparisons against the baselines on RealEstate10K and CO3Dv2, complementing the qualitative results in the main text. The comparisons use the same LVSM decoder architecture and input views; the feature sources associated with each encoding are specified in the main experiments. Consistent with the quantitative results, HRoPE recovers sharper structure on the surfaces where a tilted local plane matters most: oblique walls, floors, and table tops on RealEstate10K, and the slanted faces and supporting surfaces of objects on CO3Dv2, where the projective and ray-based encodings tend to blur the geometry or smear texture across depth discontinuities. Fig. 10 further shows results under the EscherNet setting on Objaverse, where HRoPE produces cleaner geometry and more faithful appearance than CaPE from the same input views.

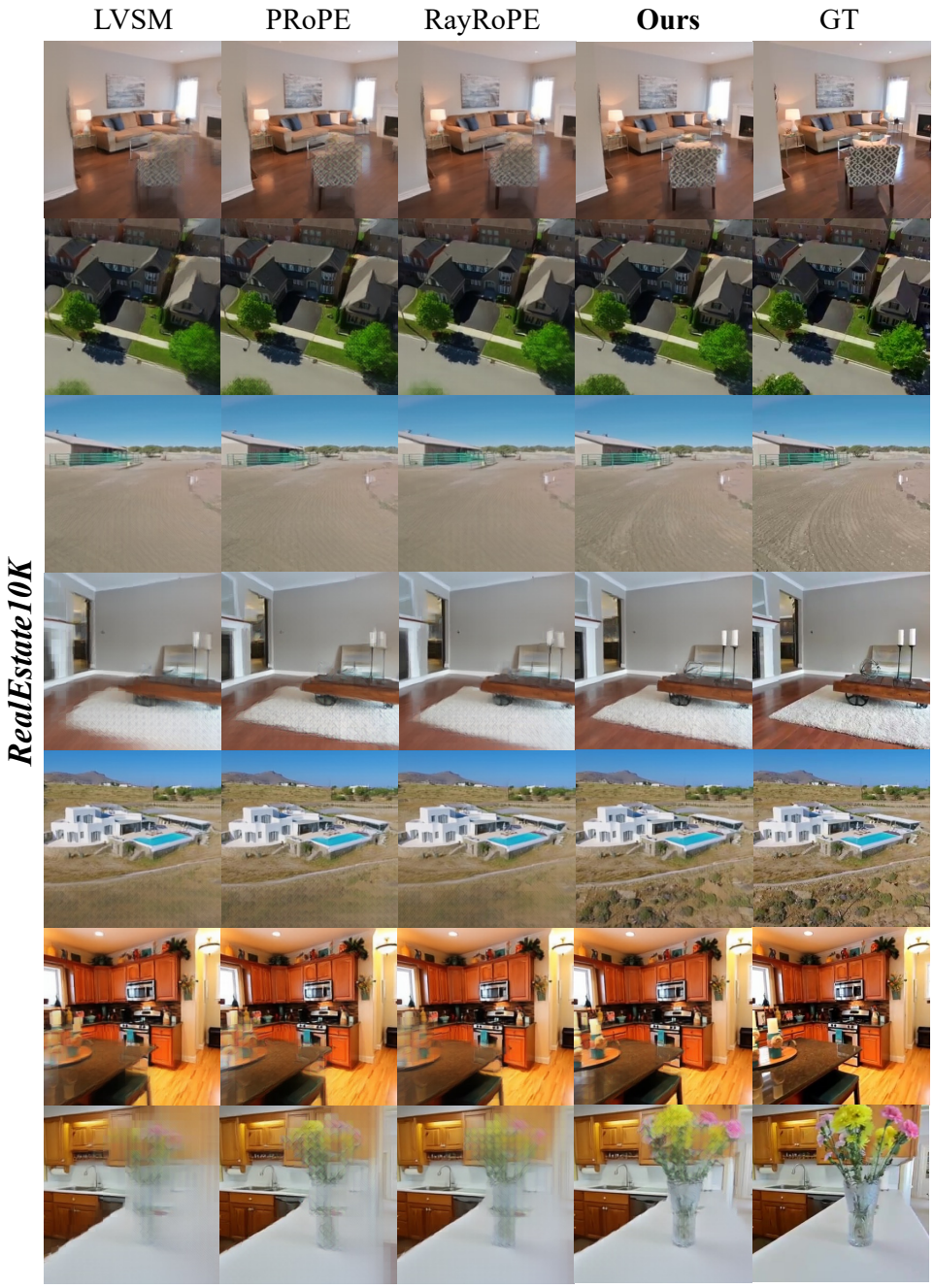


Figure 8: **Additional qualitative comparison on RealEstate10K.** Each row is one scene. From left to right, the columns show novel views synthesized by LVSM, PRoPE, RayRoPE, and HRoPE (Ours), followed by the ground-truth target view (GT). All methods use the same backbone and inputs and differ only in their positional encoding.

A.9 Limitations

HRoPE encourages correct cross-view correspondence through the structure of its rectified matching, but does not guarantee it: because the plane parameters are predicted per patch and per view, the encoding offers no formal guarantee that the same physical surface is encoded identically across views, unlike methods that enforce such uniqueness by construction. In practice the learned planes

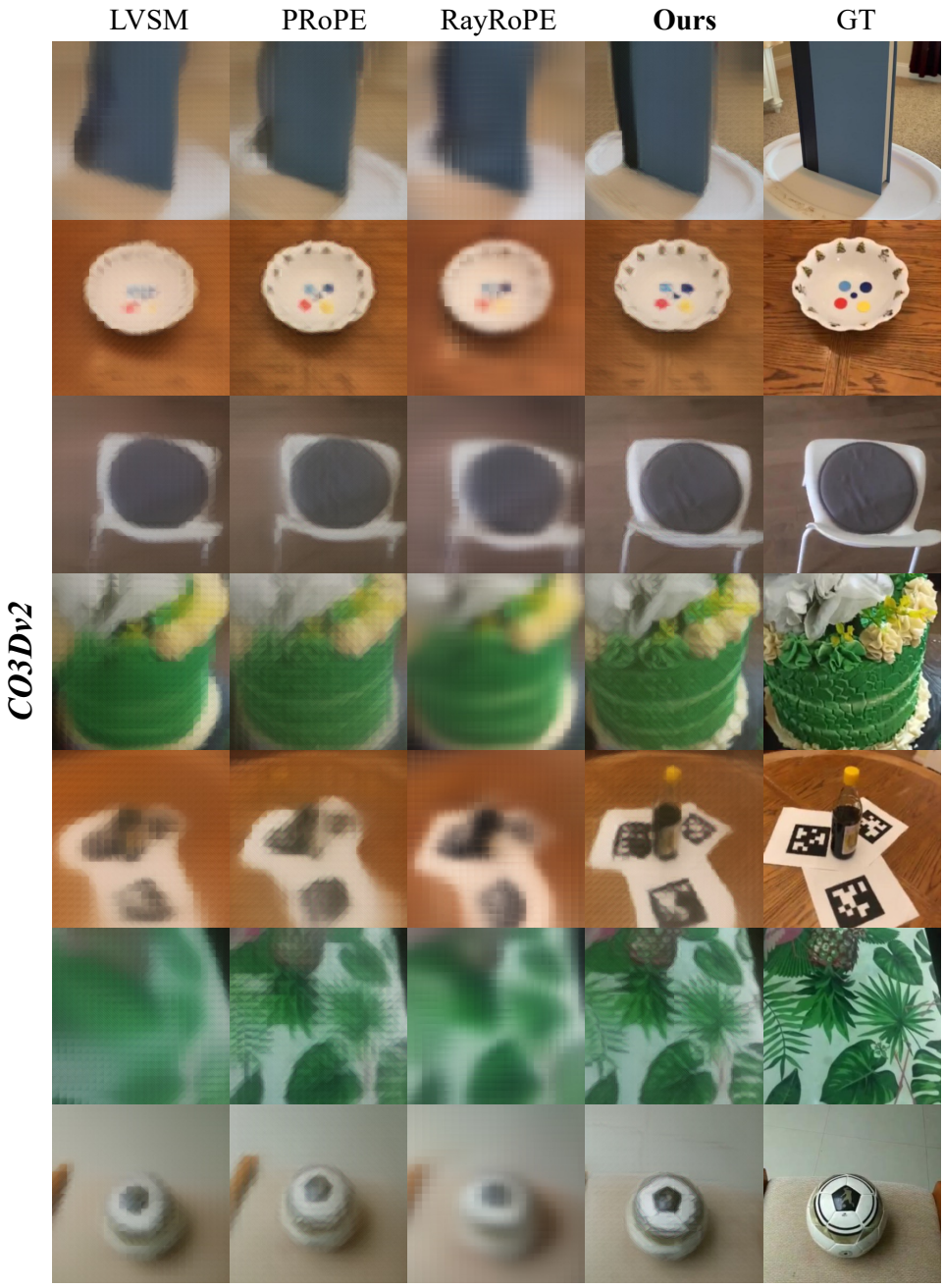


Figure 9: **Additional qualitative comparison on CO3Dv2.** Each row is one object. From left to right, the columns show novel views synthesized by LVSM, PRoPE, RayRoPE, and HRoPE (Ours), followed by the ground-truth target view (GT). All methods use the same backbone and inputs and differ only in their positional encoding.

are cross-view consistent (Sec. 4.2.2) and the calibrated $P \cdot F$ bridge corroborates the underlying geometry, but a hard guarantee remains future work.

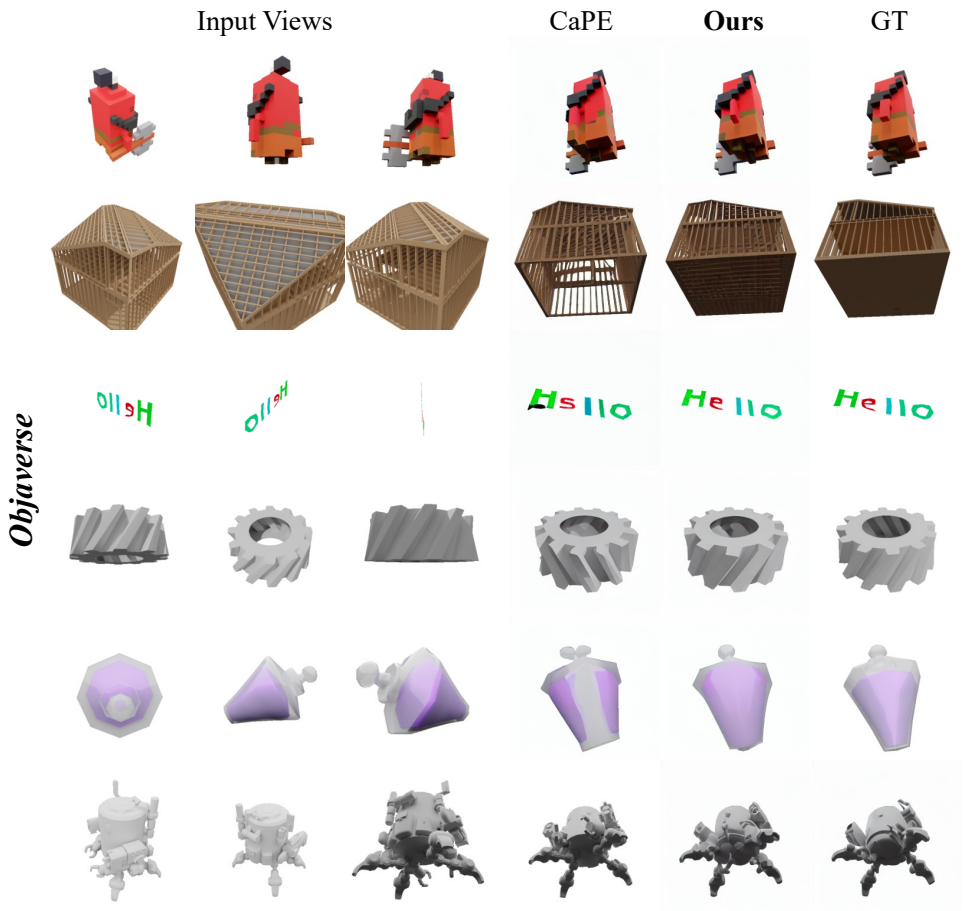


Figure 10: **Additional qualitative comparison with EscherNet on Objaverse.** Each row is one object. From left to right, the columns show three input views, the novel views synthesized with CaPE and HRoPE (Ours), and the ground-truth target view (GT). Both methods use the same EscherNet backbone and inputs and differ only in their positional encoding.