Cameras as Relative Positional Encoding

Ruilong Li[1,2*] **Brent Yi**[1*] **Junchen Liu**[1*] **Hang Gao**[1] **Yi Ma**[1,3] **Angjoo Kanazawa**[1]

[1]*UC Berkeley* [2]*NVIDIA* [3]*HKU*

Abstract

Transformers are increasingly prevalent for multiview computer vision tasks, where geometric relationships between viewpoints are critical for 3D perception. To leverage these relationships, multiview transformers must use camera geometry to ground visual tokens in 3D space. In this work, we compare techniques for conditioning transformers on cameras: token-level raymap encodings, attention-level relative pose encodings, and a new relative encoding—Projective Positional Encoding (PRoPE)—that captures complete camera frustums, both intrinsics and extrinsics, as a relative positional encoding. Our experiments begin by showing how relative conditioning methods improve performance in feedforward novel view synthesis, with further gains from PRoPE. This holds across settings: scenes with both shared and varying intrinsics, when combining token- and attention-level conditioning, and for generalization to inputs with out-of-distribution sequence lengths and camera intrinsics. We then verify that these benefits persist for different tasks, stereo depth estimation and discriminative spatial cognition, as well as larger model sizes. Code is available on our project webpage[2].

1 Introduction

Images of our world exist in the context of the viewpoints they were captured from. The geometry of these viewpoints—intrinsic and extrinsic parameters that give pixel coordinates their physical meaning—ground visual observations in 3D space. This spatial grounding is increasingly important in deep learning, especially as advances in 3D vision and embodied intelligence make multiview tasks more ubiquitous.

To solve multiview tasks with transformers, models must bind viewpoint information to patch tokens from each input image. This binding requires special care: just as naive positional encoding techniques for 1D sequences hinder performance for learning in language models [1], naive encodings of camera geometry may also be suboptimal for multiview vision models [2–4]. Advances in both settings can be summarized as transitions from absolute [5] to relative [6] encodings.

In this work, we study the problem of conditioning vision transformers on the camera geometry of input images. We survey existing techniques for addressing this, which include (i) absolute encodings in the form of pixel-aligned, token-level raymaps—these are the most common in recent state-of-the-art models [7–10]—and (ii) attention-level relative encodings based on SE(3) pose relationships [3, 4]. We then present a new camera conditioning technique, Projective Positional Encoding (PRoPE), that is designed to capture the complete geometry of cameras as a relative positional encoding. PRoPE models viewing *frustum* relationships that describe both intrinsics and extrinsics, while remaining easy to incorporate with standard transformer architectures and fused attention kernels [11].

Our experiments are on three tasks, which span six datasets. We begin with a series of studies comparing camera conditioning techniques for feedforward novel view synthesis using RealEstate10K [12] and Objaverse [13]. Our results highlight the advantages of relative encodings—particularly PRoPE—compared to absolute ones. We then verify that these benefits extend to other settings: we demonstrate improvements when integrating PRoPE into UniMatch [14] for stereo depth estimation across three

[*]Equal contribution

[2]`https://www.liruilong.cn/prope/`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-0540

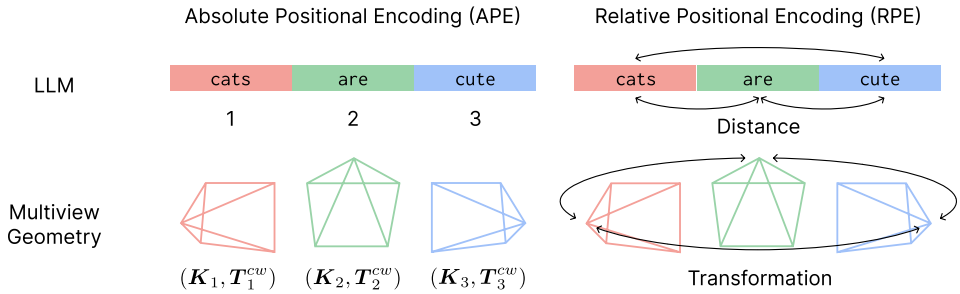


Figure 1: **Cameras as relative positional encoding.** Language models and multiview transformers must both bind "positional" information to input tokens, in terms of sequence position for language and camera parameters for multiview computer vision. We present a study on camera conditioning that includes absolute positional encodings (raymaps), relative pose encodings [2, 4], and a new method captures relative projective relationships between more complete camera *frustums*.

benchmarks, for a discriminative spatial cognition task using DL3DV [15], and when scaling to larger novel view synthesis models [7, 8].

The contributions of this paper are as follows:

(1) Survey. We survey both absolute raymap and relative SE(3) conditioning techniques for camera geometry in multiview transformers.

(2) Method. We propose PRoPE (Projective Positional Encoding), a new relative positional encoding technique that injects both camera intrinsics and extrinsics into a transformer's self-attention blocks.

(3) Evaluation. We present a series of novel view synthesis (NVS) experiments that compare camera conditioning techniques empirically. Our results highlight the advantages of relative pose encoding methods like CAPE [4] and GTA [3], while demonstrating further improvements from PRoPE across a range of settings: scenes with shared intrinsics, scenes with varying intrinsics, for hybrid conditioning that combines both token-level and attention-level representations, and in generalization for out-of-distribution test inputs.

(4) Task generalization. We show that the benefits of cameras as relative positional encoding generalize (i) to stereo depth estimation when integrated into UniMatch, (ii) to discriminative spatial cognition, and (iii) when scaling to larger model sizes.

2 Related Work

Absolute and relative positional encodings. Transformer architectures are permutation-invariant; they therefore require explicit position encoding to understand token order in sequential inputs [5]. Position encoding in sequence models has been an active area of research [1, 16–19]. While early works [20–27] focused on absolute positional encoding (APE), recent methods have increasingly adopted relative positional encoding (RPE), particularly RoPE [28], as a standard across domains, including natural language processing [29, 30, 1, 31] and computer vision [32–34, 28, 35]. Relative encodings aim to improve models by defining positions as relative offsets between token pairs. These offsets are injected into the pairwise interactions of standard dot product attention [5]:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{1}$$

where $Q, K, V \in \mathbb{R}^{T\times d}$. Position offsets can be injected using the pairwise nature of the $QK^\top \in \mathbb{R}^{T\times T}$ matrix, either via additive biases [6, 36, 37] or SO(2)-based rotation [1]. RPE offers important advantages over APE, including translation invariance, improved relationship modeling, and generalization to long sequences [38, 1, 39]. In this work, we study both absolute and relative encodings for conditioning transformers on camera geometry instead of 1D position.

Multiview transformers. Many computer vision tasks are multiview: they take multiple images and known camera geometry for each image as input. Examples exist in 3D reconstruction and view synthesis [40–44], pose estimation [45], depth prediction [14, 46, 47], 3D scene understanding [48, 49], robotics [50], and world models [51]. Many recent works leverage the improved scaling properties [22, 52, 53] of vision transformers for solving these tasks [8, 7, 9, 10, 54]. These models slice input images into patches, and use each patch as an independent visual token for the transformer. In this work, we use the model designs proposed by LVSM [8] and UniMatch [14] as a starting point for studying a critical design decision—how transformers are conditioned on camera geometry.

https://doi.org/10.52202/085713-0540

Camera conditioning in transformers. The dominant approach for conditioning multiview transformers on cameras is currently raymaps [45, 7, 8, 10]: per-pixel 6D embeddings that contain either ray origins and directions [44, 7] or Plücker coordinates [55, 45]. Concatenating these parameters to pixels enables conditioning on both camera intrinsics and extrinsics at the token level. Raymaps, however, require defining a frame of reference [56, 7, 57], which is problematic because the choice of world coordinate system is arbitrary and can hinder generalization. While this problem can be partially addressed by normalizing poses [58, 57, 8], prior works have shown that a more fundamental fix is possible through relative parameters [2–4, 59]. Notably, attention-level encodings that capture relative SE(3) poses don't require defining a consistent global frame, are compatible with fused attention kernels [11], and have been shown to improve novel view synthesis performance [3, 4]. In the following section, we survey both absolute raymap and relative SE(3) methods for encoding camera geometry for transformers. We then propose a new relative encoding method, PRoPE, that encodes relationships between more complete camera *frustums*.

3 Conditioning Transformers on Cameras

3.1 Preliminaries

We study transformers that take N images from known cameras as input:

$$\{(\boldsymbol{I}_i, \boldsymbol{K}_i, \boldsymbol{T}_i^{cw})\}_{i=1}^N, \tag{2}$$

where each $\boldsymbol{I}_i \in \mathbb{R}^{H\times W\times 3}$ is an image, $\boldsymbol{K}_i \in \mathbb{R}^{3\times 3}$ is the camera intrinsics, and $\boldsymbol{T}_i^{cw} = (\mathbf{R}_i^{\text{cw}}, \mathbf{t}_i^{\text{cw}}) \in \text{SE}(3)$ is the transformation for computing camera coordinates from world coordinates. The latter two terms encode the viewing frustum that corresponds to each image: intrinsics capture the shape and field-of-view of the frustum, while extrinsics capture position and orientation. Both intrinsics and extrinsics are encapsulated in the "world-to-image" projection matrix $\boldsymbol{P}_i \in \mathbb{R}^{3\times 4}$:

$$\boldsymbol{P}_i = \begin{bmatrix}\boldsymbol{K}_i & \mathbf{0}^{3\times 1}\end{bmatrix} \boldsymbol{T}_i^{cw}. \tag{3}$$

For notational convenience, these 3×4 projection matrices can be made invertible by lifting to 4×4 with the standard basis vector $\mathbf{e}_4 = (0,0,0,1)^\top$. This transformation maps 3D world coordinates to a projective image space defined by the frustum of camera i. It can be used to compute 2D image coordinates from world coordinates:

$$\tilde{\boldsymbol{P}}_i = \begin{bmatrix}\boldsymbol{P}_i \\ \mathbf{e}_4^\top\end{bmatrix}; \quad \begin{bmatrix}\tilde{\mathbf{x}}_i \\ 1\end{bmatrix} \propto \tilde{\boldsymbol{P}}_i \tilde{\mathbf{X}}_{\text{world}}, \tag{4}$$

where $\tilde{\mathbf{x}}_i \in \mathbb{R}^3$ and $\tilde{\mathbf{X}}_{\text{world}} \in \mathbb{R}^4$ are homogeneous coordinates in the image and world respectively. The inverse relationship can be used to compute ray directions in 3D space from 2D image coordinates. For homogeneous image coordinate $\tilde{\mathbf{x}}_i^{u,v} = (u,v,1)^\top$,

$$\begin{bmatrix}\alpha \mathbf{d}_i^{u,v} \\ 1\end{bmatrix} \propto \tilde{\boldsymbol{P}}_i^{-1} \begin{bmatrix}\tilde{\mathbf{x}}_i^{u,v} \\ 1\end{bmatrix}, \tag{5}$$

where $\alpha \in \mathbb{R}$ is a scalar magnitude and $\mathbf{d}_{\mathbf{i}}^{\mathbf{u},\mathbf{v}} \in \mathbb{S}^2$ is a unit-norm ray direction.

3.2 Pixel-aligned Camera Encoding

Token-level, pixel-aligned raymaps are the dominant method for encoding geometry in multiview transformers [7, 42, 10, 9]. Networks that employ raymaps concatenate images $\boldsymbol{I}_i \in \mathbb{R}^{H\times W\times 3}$ with per-pixel raymaps $\mathbf{M}_i \in \mathbb{R}^{H\times W\times R}$ along the channel dimension, which expands inputs to $\mathbb{R}^{H\times W\times(3+R)}$. There are two main approaches for computing these raymaps, which we refer to as "naive" and Plücker.

Naive raymaps. Naive raymaps [7] are computed as per-pixel origin and direction vectors:

$$\mathbf{M}_{i,\texttt{Naive}}^{u,v} = \begin{bmatrix}\mathbf{o}_i \\ \mathbf{d}_i^{u,v}\end{bmatrix} \in \mathbb{R}^6 \quad \text{for } (u,v) \in [1,W]\times[1,H] \tag{6}$$

$$\mathbf{o}_i = -(\mathbf{R}_i^{\text{cw}})^\top \mathbf{t}_i^{\text{cw}}, \tag{7}$$

where each ray direction $\mathbf{d}_i^{u,v}$ is computed using $\tilde{\boldsymbol{P}}_i$ by following Equation 5.

https://doi.org/10.52202/085713-0540

Plücker raymaps. Plücker raymaps [45, 8] can be implemented by replacing the origin term in naive raymaps with a moment term:

$$\mathbf{M}^{u,v}_{i,\texttt{Plücker}} = \begin{bmatrix} \boldsymbol{o}_i \times \boldsymbol{d}^{u,v}_i \\ \boldsymbol{d}^{u,v}_i \end{bmatrix} \in \mathbb{R}^6 \quad \text{for } (u,v) \in [1,W] \times [1,H]. \tag{8}$$

This moment term makes the ray representation invariant to the choice of origin along the ray.

Properties. Raymaps offer a simple approach for conditioning on both camera intrinsics and extrinsics. An important drawback, however, is that they are *absolute*: similar to early position encoding techniques for 1D sequences [5], raymaps are expressed in global terms. They are sensitive to the arbitrary choice of reference frame, which can hinder generalization.

3.3 Relative SE(3) Encoding

To remove the need for a global reference frame, recent works have introduced *relative* encodings for SE(3) camera poses. Two existing approaches fall under this category: CaPE [4] and GTA [3]. Given images i_1 and i_2, both aim to condition networks on $\boldsymbol{T}^{cw}_{i_1}(\boldsymbol{T}^{cw}_{i_2})^{-1}$ using modified self-attention blocks. This captures dense relationships—how each camera pose is situated relative to every other camera pose—and makes networks invariant to how the world frame w is defined.

Notation. To formalize the operations required for attention-level geometry encoding, we denote the batched matrix-vector product $\circledcirc$, Kronecker product $\otimes$, and identity matrices $\mathbf{I}_k \in \mathbb{R}^{k\times k}$. We use i for image/camera indices and t for patch/token indices. $i(t)$ is the index of the image that patch t belongs to. Rows of the $Q, K, V \in \mathbb{R}^{T\times d}$ matrices are subscripted $Q_t, K_t, V_t \in \mathbb{R}^d$. Batched matrix-vector products are defined as:

$$\mathbf{A} \in \mathbb{R}^{N\times d_1\times d_2}, B \in \mathbb{R}^{N\times d_2} \implies (\mathbf{A} \circledcirc B) \in \mathbb{R}^{N\times d_1} \tag{9}$$

$$(\mathbf{A} \circledcirc B)_{nj} = \sum_k \mathbf{A}_{njk} B_{nk}. \tag{10}$$

We use $\mathbf{D} \in \mathbb{R}^{T\times d\times d}$ to denote batches of block-diagonal matrices, where individual matrices in $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_T]$ are subscripted $\mathbf{D}_t \in \mathbb{R}^{d\times d}$.

CaPE [4]. CaPE injects relative SE(3) pose by transforming the Q and K matrices before they are passed to self-attention. CaPE can be formalized using per-token block-diagonal matrices $\mathbf{D}^{\texttt{CaPE}}_t$, which is computed by diagonally repeating the camera extrinsics:

$$\mathbf{D}^{\texttt{CaPE}}_t = \mathbf{I}_{d/4} \otimes \boldsymbol{T}^{cw}_{i(t)} \tag{11}$$

Like RoPE [1], transformations are then applied to the Q and K matrices before self-attention. This can be encapsulated into an augmented self-attention block:

$$\text{Attn}^{\texttt{CaPE}}(Q,K,V) = \text{Attn}((\mathbf{D}^{\texttt{CaPE}})^\top \circledcirc Q, (\mathbf{D}^{\texttt{CaPE}})^{-1} \circledcirc K, V). \tag{12}$$

The effect of this is that each $Q^\top_{t_1} K_{t_2} \in \mathbb{R}$ dot product in $QK^\top \in \mathbb{R}^{T\times T}$ is replaced with:

$$Q^\top_{t_1} \mathbf{D}^{\texttt{CaPE}}_{t_1} (\mathbf{D}^{\texttt{CaPE}}_{t_2})^{-1} K_{t_2} \in \mathbb{R}, \tag{13}$$

where $\mathbf{D}^{\texttt{CaPE}}_{t_1}(\mathbf{D}^{\texttt{CaPE}}_{t_2})^{-1} = \mathbf{I}_{d/4} \otimes \left[\boldsymbol{T}^{cw}_{i(t_1)}(\boldsymbol{T}^{cw}_{i(t_2)})^{-1}\right]$, thus conditioning outputs on relative pose.

GTA [3]. GTA proposes a formulation for per-token transformations with a similar high-level goal as CaPE. GTA's attention variant transforms the Q and K matrices the same way, while proposing to also transform the V matrix:

$$\text{Attn}^{\texttt{GTA}}(Q,K,V) = \mathbf{D}^{\texttt{GTA}} \circledcirc \text{Attn}((\mathbf{D}^{\texttt{GTA}})^\top \circledcirc Q, (\mathbf{D}^{\texttt{GTA}})^{-1} \circledcirc K, (\mathbf{D}^{\texttt{GTA}})^{-1} \circledcirc V). \tag{14}$$

This has the added benefit of injecting relative transformations into the attention operator's value aggregation. The attention layer output for each token t_1 becomes

$$\left[\text{Attn}^{\texttt{GTA}}(Q,K,V)\right]_{t_1} = \sum_{t_2} \alpha_{t_1,t_2} \mathbf{D}^{\texttt{GTA}}_{t_1} (\mathbf{D}^{\texttt{GTA}}_{t_2})^{-1} V_{t_2}, \tag{15}$$

where α_{t_1,t_2} is a softmax score computed from the transformed dot product (Equation 13). GTA's experiments [3] compare several SE(3)-based formulations for $\mathbf{D}^{\texttt{GTA}}$, with and without the value matrix transformation. The best-performing methods include the value transform, and both SE(3) for camera pose and RoPE [1] for 2D patch position. Our experiments include GTA using these terms.

3.4 Projective Position Encoding (PRoPE)

We introduce a new relative positional encoding method in our study, which we call PRoPE. The core observation of PRoPE is that the SE(3) poses considered by existing relative encoding techniques are only a partial representation of camera geometry. Instead of relating each camera i_1 and camera i_2 with only their poses $\boldsymbol{T}_{i_1}^{cw}$ and $\boldsymbol{T}_{i_2}^{cw}$, PRoPE uses the projective relationship between full *frustums*:

$$\tilde{\boldsymbol{P}}_{i_1}\tilde{\boldsymbol{P}}_{i_2}^{-1}. \tag{16}$$

This 4×4 matrix can be interpreted as a transformation between the local projective spaces defined by each camera; it encodes the complete geometric relationship between camera views. As we will see in Equation 20, it also retains the key global invariance property of SE(3)-based relative encodings.

To implement PRoPE, we define a new set of $\mathbf{D}_t^{\texttt{PRoPE}} \in \mathbb{R}^{d\times d}$ matrices and use GTA-style attention (Equation 14) to inject them into transformer blocks. We design these matrices to (1) encode frustum relationships *between* cameras—this uses the projective relationship in Equation 16—and (2) encode relative patch positions *within* cameras—this follows GTA [3] and uses RoPE terms. These goals are achieved with complementary submatrices, each with shape $\frac{d}{2} \times \frac{d}{2}$:

$$\mathbf{D}_t^{\texttt{PRoPE}} = \begin{bmatrix} \mathbf{D}_t^{\texttt{Proj}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_t^{\texttt{RoPE}} \end{bmatrix} \tag{17}$$

$$\mathbf{D}_t^{\texttt{Proj}} = \mathbf{I}_{d/8} \otimes \tilde{\boldsymbol{P}}_{i(t)} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}} \tag{18}$$

$$\mathbf{D}_t^{\texttt{RoPE}} = \begin{bmatrix} \text{RoPE}_{d/4}(x_t) & \mathbf{0} \\ \mathbf{0} & \text{RoPE}_{d/4}(y_t) \end{bmatrix} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}. \tag{19}$$

In these definitions, $\text{RoPE}_{d/4}(\cdot)$ constructs $\frac{d}{4} \times \frac{d}{4}$ rotary embeddings [1] for (x_t, y_t), which are the patch coordinates for token t.

3.5 Properties of PRoPE

PRoPE has several important properties, which become more evident when we expand the projective transformation:

$$\tilde{\boldsymbol{P}}_{i_1}\tilde{\boldsymbol{P}}_{i_2}^{-1} = \begin{bmatrix} \boldsymbol{K}_i & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \boldsymbol{T}_i^{cw}(\boldsymbol{T}_j^{cw})^{-1} \begin{bmatrix} \boldsymbol{K}_j^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{20}$$

Global frame invariance. Redefining the world frame is equivalent to right-multiplying both $\boldsymbol{T}^{cw}$ $SE(3)$ terms, which is algebraically eliminated in Equation 20.

Reduction to relative SE(3) attention. For cameras with identity intrinsics, Equation 20 reduces to the relative $SE(3)$ transformations utilized in CAPE and GTA. These methods can be interpreted as a case of PRoPE where the intrinsic matrices are set to identity.

Reduction to RoPE [28]. Equation 20 evaluates to identity for patches from the same image. For these token pairs, $\mathbf{D}_t^{\texttt{PRoPE}}$ contains only the RoPE terms used by single-image vision transformers.

4 Experiments

The goal of our experiments is to understand how camera conditioning techniques—including PRoPE—impact the performance of multiview transformers. To accomplish this, we present experiments comparing encoding strategies under several task and evaluation conditions.

4.1 Experiment Setup

We include metrics for several camera conditioning techniques—Naive and Plücker raymap encodings, CAPE [4], GTA [3], and PRoPE. In our experiments, GTA refers to the SE(3)+SO(2) variation studied by [3], where SO(2) refers to RoPE on patch positions. As discussed in Section 3.4, PRoPE adopts the self-attention mechanism and RoPE combination proposed by GTA. The primary difference between PRoPE and GTA is therefore the use of relative projective relationships instead of relative SE(3) relationships between cameras.

Our core experiments evaluate camera conditioning techniques using feedforward novel view synthesis (NVS). NVS is an ideal benchmarking task because it requires fine-grained geometric reasoning:

 https://doi.org/10.52202/085713-0540

Method	RealEstate10K [12]			Objaverse [13]		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Plücker Raymap	20.48	0.209	0.622	21.44	0.159	0.851
Naive Raymap	20.54	0.210	0.623	21.59	0.153	0.856
CAPE [4]	21.11	0.234	0.656	19.68	0.220	0.827
GTA [3]	22.51	0.164	0.707	**23.70**	**0.104**	**0.879**
PRoPE	**22.80**	**0.146**	**0.725**	**23.70**	**0.104**	**0.879**

Table 1: **Novel view synthesis comparison, with constant intrinsics in each scene**. We compare different camera conditioning approaches applied to the LVSM [8] framework.

Method	RealEstate10K [12]			Objaverse [13]		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Plücker Raymap	19.89	0.327	0.608	21.43	0.177	0.852
Naive Raymap	20.56	0.301	0.629	21.00	0.191	0.845
CAPE [4]	15.94	0.497	0.699	16.78	0.322	0.760
GTA [3]	15.77	0.512	0.641	18.00	0.257	0.775
PRoPE	**21.42**	**0.247**	**0.678**	**22.98**	**0.138**	**0.871**

Table 2: **Novel view synthesis, with varying intrinsics in each scene**. We compare the LVSM [8] model trained with different camera conditioning strategies, on intrinsics-augmented dataset variants.

models are trained to render scenes from target viewpoints, given only calibrated reference images and target camera parameters. We do this by reimplementing and training variants of LVSM [8], a state-of-the-art novel view synthesis method that originally encoded camera geometry with Plücker raymaps. We train and evaluate separately on the RealEstate10K [12] and Objaverse [13] datasets. Scenes in RealEstate10K are captured with constant intrinsics, but cameras vary between scenes. Objaverse renders use the same camera intrinsics across the entire dataset.

We take several steps to ensure fairness in evaluations. Models are trained in the same codebase, with matched hyperparameters and training steps. All main experiments use identical input, output, and overall model sizes ($\sim$25M parameters); we also validate larger models in Section 4.7. More details are provided in Appendix A.1.1.

4.2 Relative vs Absolute Positional Encodings

Novel view synthesis results are presented in Table 1 and discussed below.

Relative encodings outperform absolute ones. Consistent with prior work [4, 3], we observe that relative encodings for camera geometry consistently outperform absolute ones. CAPE, GTA, and PRoPE all yield improvements over widely used raymap encodings, with PRoPE (followed closely by GTA) producing the best results.

Projective positional encoding improves view synthesis quality. PRoPE consistently outperforms other encoding methods across metrics on RealEstate10K [12], despite the dataset's limited intrinsics variation. This confirms that capturing more complete camera information (both intrinsic and extrinsic) in our relative encoding is beneficial. We also find no loss of performance when the train and test images all have constant intrinsics: GTA [3] and PRoPE produce identical metrics for the Objaverse dataset, which verifies that PRoPE reduces to GTA when camera intrinsics are unimportant.

4.3 Attention-Level Intrinsics Conditioning

Real-world data often involves different cameras and focal lengths—consider multiview rigs on autonomous cars or zoom lenses on point-and-shoot cameras. To understand how effective PRoPE is at encoding intrinsics information, we evaluate each conditioning method on intrinsics-augmented versions of the RealEstate10K [12] and Objaverse [13] datasets. We augment RealEstate10K by applying a zoom factor sampled uniformly in $[1, 3]$ to each image. For Objaverse, we switch from constant field of views to uniformly sampled ones between 35 and 50 degrees. In contrast to Section 4.2, this means that cameras within scenes can vary in both extrinsics *and* intrinsics. We present quantitative results in Table 2 and qualitative results in Figure A.1.

PRoPE enables intrinsic-aware multiview understanding. We observe that PRoPE outperforms all alternative camera conditioning techniques for both the RealEstate10K and Objaverse datasets. Existing attention-based methods, which let networks condition on relative pose, perform poorly

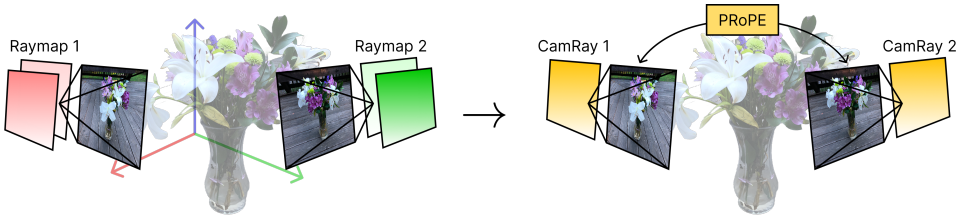


Figure 2: **Hybrid camera encoding.** *Left:* token-level conditioning using only raymaps, which capture both intrinsics and extrinsics. *Right:* hybrid encoding where attention-level PRoPE captures relationships between camera frustums, and token-level CamRay encodes local ray directions.

Method	RealEstate10K [12]			Objaverse [13]		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
GTA [3]	15.77	0.512	0.641	16.98	0.261	0.762
GTA+CamRay	21.41	0.238	0.673	22.69	0.123	0.869
PRoPE	21.42	0.247	0.678	22.82	0.119	0.872
PRoPE+CamRay	**21.78**	**0.211**	**0.692**	**22.98**	**0.114**	**0.874**

Table 3: **Novel view synthesis with *hybrid* camera encodings, with varying intrinsics in each scene**. Intrinsics can be conditioned by concatenating local frame camera rays to the network input.

without knowledge of intrinsics. While token-level raymaps carry sufficient camera information, they perform worse overall than PRoPE's relative conditioning formulation.

4.4 Hybrid Encoding Strategies

Token- and attention-level camera encodings require modifications to different parts of the transformer architecture. They are therefore compatible with each other: both conditioning styles can be used simultaneously. To compare PRoPE against a stronger baseline while simultaneously exploring these "hybrid" conditioning strategies (Figure 2), we train LVSM variations that couple relative encodings with local, *camera-frame* raymaps:

$$\mathbf{M}^{u,v}_{i,\texttt{CamRay}} = \mathbf{R}^{\text{cw}}_i \mathbf{d}^{u,v}_i \propto \boldsymbol{K}_i^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^\top \in \mathbb{R}^3 \tag{21}$$

We refer to this raymap as CamRay. CamRay shares many properties with existing raymaps (Section 3.2)—it encodes intrinsics, is pixel-aligned, and can be concatenated to input images—but it is not tied to an absolute coordinate system. It can therefore be used in conjunction with relative pose and camera encoding techniques without sacrificing global frame invariance. As we observe in Section 4.6, this provides empirical advantages over Plücker raymaps.

CamRay can be interpreted as a token-level encoding of camera intrinsics. We therefore evaluate it using the intrinsics-augmented NVS datasets described in Section 4.3. Results are reported in Table 3 and discussed below.

PRoPE effectively encodes camera geometry. On both RealEstate10K and Objaverse, we observe that PRoPE is comparable with or outperforms GTA+CamRay. This is true even though PRoPE is simpler: it is only applied attention-level, while GTA+CamRay includes both attention-level and token-level terms.

Token-level and attention-level conditioning techniques are complementary. GTA and PRoPE both benefit from the extra CamRay input. GTA benefits significantly more; this can be explained by the absence of intrinsics information in the standard SE(3)-based GTA formulation.

4.5 Out-of-distribution Robustness

One hypothesis for why relative camera encodings outperform absolute ones is improved generalization characteristics; this is similar to why RoPE [28] can improve performance for language modeling. To test this, we benchmark conditioning methods using test-time settings that introduce distribution shifts in sequence length and intrinsics (Figure 3).

Setting 1: Longer Input Sequences at Test Time. Inspired by test-time context length extrapolation [60, 61], our first setting deploys NVS models trained with a fixed number of input views (2 in our experiments) to significantly more views (up to 16) at test time. This is particularly important

https://doi.org/10.52202/085713-0540

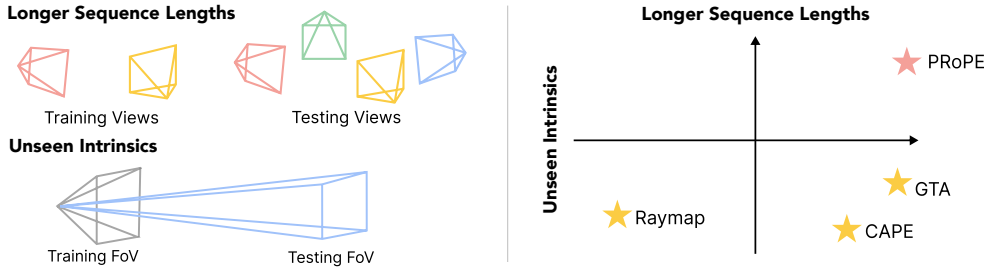
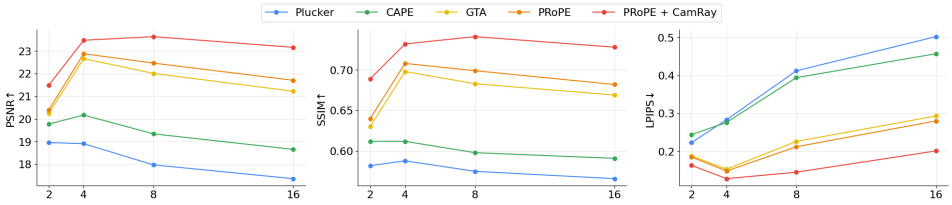
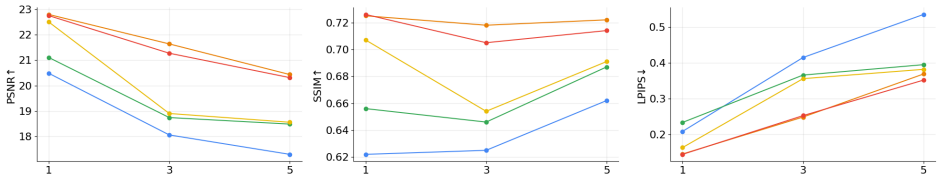


Figure 3: **Out-of-distribution tasks.** *Left:* We evaluate camera conditioning methods on both longer sequence lengths and unseen camera intrinsics. *Right:* PRoPE improves results for both unseen sequence lengths and unseen intrinsics.



(a) Testing with {2, 4, 8, 16} input views.



(b) Testing with {1, 2, 3}× zooming-in.

Figure 4: **Evaluation on RealEstate10K** [12]. Relative encoding methods demonstrate superior robustness on handling (a) varying numbers of input views and (b) different focal lengths at test time.

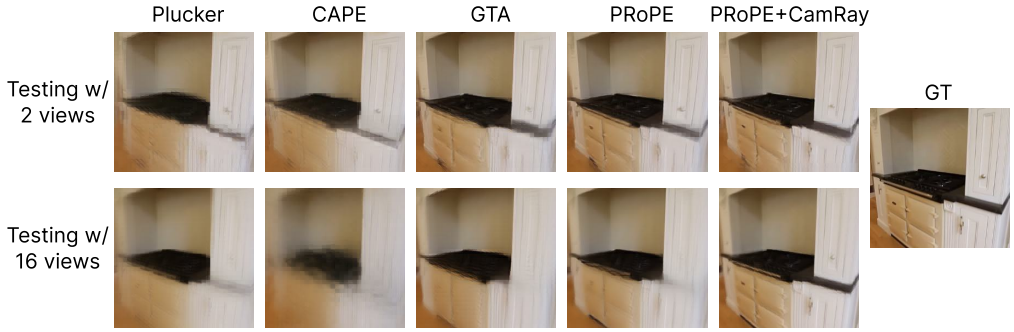


Figure 5: **Results of Longer Input Sequences at Test Time.** PRoPE demonstrates superior robustness when tested with varying numbers of input views, despite being trained with only 2 views.

Figure 6: **Results of Varying Focal Length at Test Time.** PRoPE explicitly models relative intrinsics—we find this makes it more robust to zoomed-in test views.

in real-world scenarios where the number of observations can vary substantially across different applications, and sometimes dynamically increase.

Setting 2: Out-of-distribution Intrinsics at Test Time. Our second setting evaluates a model's ability to handle varying focal lengths at test time. This is crucial as focal length can vary significantly across

https://doi.org/10.52202/085713-0540

Dataset	Model	Abs Rel	Sq Rel	RMSE	RMSE log
RGBD [62]	UniMatch	0.123	0.175†	0.678	0.203
	UniMatch + PRoPE	**0.105**	0.203†	**0.573**	**0.181**
SUN3D [63]	UniMatch	0.131	0.098	0.397	0.169
	UniMatch + PRoPE	**0.117**	**0.075**	**0.343**	**0.152**
Scenes11 [64]	UniMatch	0.065	0.085	0.575	0.126
	UniMatch + PRoPE	**0.049**	**0.063**	**0.474**	**0.104**

Table 4: **Performance Improvement on Stereo Depth Estimation Task with UniMatch [14]**. †The "Sq Rel" metric is less reliable on the RGBD dataset due to the imperfect depth and camera pose [64].

Figure 7: **Qualitative Results on Stereo Depth Estimation Task.** Attention-level camera conditioning in UniMatch [14] leads to significant estimation improvements.

Method	5 views	9 views	17 views
Plücker	69.1%	76.9%	74.6%
PRoPE+Plücker	81.1%	90.5%	91.8%
PRoPE+CamRay	**86.1%**	**93.0%**	**94.3%**

Table 5: **Spatial cognition results**. We report the accuracy of detecting inconsistent image-camera pairs on the DL3DV [15] dataset under varying numbers of input views. Both CamRay and PRoPE significantly help with performance, without introducing additional model parameters. An illustration of this task can be found in Figure A.3.

different cameras and zoom levels, and it is impractical to train a separate model for every possible focal length. We test our models with focal lengths ranging from $1\times$ to $5\times$ the training focal length, simulating scenarios where more zoomed-in images are seen during deployment.

Relative encodings improve generalization; PRoPE outperforms alternatives. Evaluated results on the RealEstate10K [12] dataset are summarized in Figure 4, with visuals provided in Figure 5 and 6. We make three main observations. First, while Plücker Raymap encodes more complete camera information than CAPE and GTA, it consistently underperforms across all settings—even when intrinsics information is critical. Second, PRoPE improves performance and robustness in both out-of-distribution settings, particularly when handling out-of-distribution focal lengths. This indicates that explicitly modeling the relative projective relationship between cameras is more effective than modeling only the relative SE(3) relationship, as done in GTA and CAPE. Finally, we found adding CamRay to PRoPE actually *hurts* performance for intrinsics extrapolation; this suggests that PRoPE is uniquely useful for intrinsics generalization.

4.6 Task Generalization

Like our results so far, prior studies on relative pose encodings [3, 4] for multiview transformers have focused experiments on novel view synthesis. To better understand how these conclusions generalize, we evaluate PRoPE in two new tasks: stereo depth estimation using UniMatch [14] and a spatial cognition task designed around DL3DV [15].

Stereo Depth Estimation. We set up this task using UniMatch [14], a pretrained multiview transformer that has seen wide adoption in downstream applications [65–68]. UniMatch was originally trained on

https://doi.org/10.52202/085713-0540

Method	1× Compute			100× Compute		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Plücker Raymap	20.48	0.622	0.209	25.64	0.809	0.084
PRoPE	**22.80**	**0.725**	**0.146**	**26.56**	**0.832**	**0.071**

Table 6: **Scaling LVSM with increased compute.** We compare LVSM [8] models trained with Plücker raymaps versus PRoPE on RealEstate10K [12] at two compute scales.

three different tasks; we focus on the stereo depth estimation task, which assumes known relative camera poses between input views. We incorporate camera information into UniMatch's cross-view attention mechanism using PRoPE, modifying only ∼50 lines of the official code. All models follow the exact same training protocols as described in the original paper.

Spatial Cognition. Next, we design a spatial cognition task inspired by [69]. In this task, a network is given multiple images of the same scene, each paired with camera information. The problem is designed so that it cannot be solved by analyzing the camera information alone, the images alone, or without reasoning about the multiview relationships among all inputs. One of the image-camera pairs is intentionally corrupted by assigning it an incorrect camera pose sampled from other frames. The network is then required to identify the incorrect image-camera pair based on geometric consistency. See Appendix A.1.3 for implementation details, and Figure A.3 for input-output examples.

PRoPE's benefits generalize across tasks. For depth estimation, we provide quantitative results in Table 4 and qualitative results in Figure 7. For spatial cognition, accuracy metrics are provided in Table 5. We find that PRoPE significantly improves multiview understanding across both tasks. In our spatial cognition task, we observe that performance with PRoPE continues to improve as the number of views increase during testing, whereas the Plücker raymaps do not exhibit the same trend. We also observe improvements when replacing Plücker with CamRay, which indicates that the absolute extrinsics information hinders the model's ability to generalize.

4.7 Scaling PRoPE

In our final set of experiments, we evaluate how the advantages of relative camera encoding extend to larger models with more computational resources. We conducted two experiments for this, which are discussed below.

PRoPE's benefits persist when scaling LVSM. We scale our LVSM training pipeline with approximately 100× more computational resources (details in Appendix A.1.1). We train two variants of this larger LVSM model: one that follows the original LVSM paper [8] and uses Plücker rays and one that incorporates PRoPE. Results are reported in Table 6, where we observe that the relative PRoPE encoding continues to improve model quality—by a smaller but still significant margin—on larger model variants trained with more resources.

PRoPE improves CAT3D results. With assistance from the original authors, we add PRoPE to and retrain CAT3D [7], a large multiview diffusion model conditioned on naive raymaps. We report metrics from this model in Table A.2. PRoPE produces consistent improvements across metrics, while introducing zero additional model parameters and negligible computational overhead.

5 Conclusion and Future Work

In this work, we highlight how representing cameras as relative positional encodings—particularly in a way that captures both camera intrinsics and extrinsics—improves multiview transformers across settings and tasks. Possibilities for future work include improving numerical stability when directly multiplying projective matrices with Q/K/V vectors; it may be possible, for example, for ill-conditioned matrices to emerge from telephoto focal lengths. It would also be interesting to extend PRoPE to distorted camera models, for example by applying PRoPE to per-patch projective approximations. Furthermore, incorporating "multi-frequency" [19, 1] encoding for camera parameters remains nontrivial due to the non-commutativity of projective transforms, a challenge shared by relative SE(3) attention methods [3, 4].

References

[1] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[2] Aleksandr Safin, Daniel Duckworth, and Mehdi SM Sajjadi. Repast: Relative pose attention scene representation transformer. *arXiv preprint arXiv:2304.00947*, 2023.

[3] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations (ICLR)*, 2024.

[4] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9503–9513, 2024.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[6] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[7] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

[8] 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. *arXiv preprint arXiv:2410.17242*, 2024.

[9] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025.

[10] Ethan Weber, Norman Müller, Yash Kant, Vasu Agrawal, Michael Zollhöfer, Angjoo Kanazawa, and Christian Richardt. Fillerbuster: Multi-view scene completion for casual captures, 2025. arXiv:2502.05175.

[11] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[12] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[13] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[14] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[16] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

[17] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[18] Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*, 2024.

 https://doi.org/10.52202/085713-0540

[19] Connor Schenck, Isaac Reid, Mithun George Jacob, Alex Bewley, Joshua Ainslie, David Rendleman, Deepali Jain, Mohit Sharma, Avinava Dubey, Ayzaan Wahid, et al. Learning the ropes: Better 2d and 3d position encodings with string. *arXiv preprint arXiv:2502.02562*, 2025.

[20] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16×16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[23] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[26] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[28] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[31] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[33] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[34] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[35] Genmo Team. Mochi 1. `https://github.com/genmoai/models`, 2024.

[36] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[37] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2022.

[38] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[39] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[40] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022.

[41] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, 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*, pages 6229–6238, 2022.

[42] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models. *arXiv:2411.18613*, 2024.

[43] Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024.

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[45] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *arXiv preprint arXiv:2402.14817*, 2024.

[46] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.

[47] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

[48] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9202–9212, 2023.

[49] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.

[50] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[51] Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. *arXiv preprint arXiv:2505.20171*, 2025.

[52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[53] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[54] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

[55] Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions of the Royal Society of London*, (155):725–791, 1865.

 https://doi.org/10.52202/085713-0540

[56] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019.

[57] Vitor Guizilini, Muhammad Zubair Irshad, Dian Chen, Greg Shakhnarovich, and Rares Ambrus. Zero-shot novel view and depth synthesis with multi-view geometric diffusion. *arXiv preprint arXiv:2501.18804*, 2025.

[58] Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision*, pages 198–216. Springer, 2022.

[59] Brent Yi, Vickie Ye, Maya Zheng, Yunqi Li, Lea Müller, Georgios Pavlakos, Yi Ma, Jitendra Malik, and Angjoo Kanazawa. Estimating body and hand motion in an ego-sensed world. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7072–7084, 2025.

[60] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624, 2023.

[61] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*, 2024.

[62] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[63] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013.

[64] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017.

[65] 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 *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[66] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2025.

[67] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259*, 2024.

[68] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. In *European Conference on Computer Vision*, pages 111–128. Springer, 2024.

[69] Tyler Bonnen, Stephanie Fu, Yutong Bai, Thomas O'Connell, Yoni Friedman, Nancy Kanwisher, Josh Tenenbaum, and Alexei Efros. Evaluating multiview object consistency in humans and image models. *Advances in Neural Information Processing Systems*, 37:43533–43548, 2025.

[70] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[71] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. *arXiv preprint arXiv:2310.17994*, 2023.

[72] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21551–21561, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our contribution as a theoretical grounded analysis and an new approach for camera conditioning in transformer that is more robust than previous works. This has been well discussed in the method section and proved in the experiment section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/085713-0540

Answer: [NA]

Justification: Our paper does involve plenty of math in section 3 but is not proving anything.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We outline the method section with detailed math. We will also provide Pseudo code in supplemental material of our algorithm.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code will be released soon.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details will be provided in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are expensive to run, and we are an academic group so can't afford multiple runs for error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/085713-0540

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Will be provided in supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We did our best to follow the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work is a foundational research on the camera encoding approaches and not tied to particular applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work is a foundational research on the camera encoding approaches so does not poses such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets in the paper are created by our own, except for the dataset which is properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/085713-0540

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-0540

A.1 Experiment details

A.1.1 LVSM-based Model Details

We adhere to the original LVSM [8] implementation specifications, maintaining consistency across most settings. For complete details regarding these configurations, we direct readers to the original LVSM paper [8]. The modifications we made include:

- We trained exclusively at 256×256 resolution and did not perform the additional fine-tuning at higher resolutions.
- Limited by academic-level resources, we use a smaller version of the LVSM model with 6 transformer blocks, and reduce the MLP channel dimension from 3072 to 1024. Our models are trained on 2x GPUs with a total batch size of 4, as opposed to 512 in the original paper. This applies to all experiments expect for the ones in Section 4.7.
- For the scaling-up experiments in Section 4.7, we use a LVSM model with 12 transformer blocks and keep the all other configurations including the MLP channel dimension (3072). These models are trained on 8x GPUs with a total batch size of 64.

A.1.2 UniMatch Modification Details

UniMatch [14]'s model consists of a cross-attention transformer to capture multi-view relationship, as well as a self-attention transformer that serves as a single-view image encoder/decoder. We inject the camera information into both transformers, on the Q/K/V/O vectors, using our formulation. Note that our formula exactly falls back to RoPE [28] in the single-view scenario. It therefore naturally works with both transformer networks in UniMatch.

A.1.3 Spatial Cognition Model Details

We formulate this task as a classification problem, where the number of classes corresponds to the number of input image-camera pairs, and the "ground-truth" class is the ID of the inconsistent image-camera pair. The training objective is to identify the inconsistent pair, which we optimize using the cross-entropy loss. The architecture of the model is largely similar to LVSM [8], where the only change is that the last linear layer is modified to output a single scalar for each token. Outputs are then averaged over all tokens for each input pair, resulting in a vector that represents the probability that each input pair is the "bad" one (over softmax). We provide more data exemplars we used for training and testing in Figure A.3.

A.2 Additional Results

A.2.1 Ablating PRoPE

As our proposed PRoPE includes two terms: the projective relationship between cameras ($\mathbf{D}_t^{\text{Proj}}$) and the patch coordinate relationship ($\mathbf{D}_t^{\text{RoPE}}$). We here ablate each of them to study their contribution. As shown in Table A.1, as a crucial component to the system, modeling the projective relationship between cameras alone already yields strong performance. Introducing RoPE [28] further enhances model's ability on understanding patch relationship, which is a minimal unit in vision transformer architecture. Notably, in PRoPE, we allocate half of the feature channels to encode each term. Ablating one term therefore means using all feature channels to encode the remaining one.

A.2.2 PRoPE v.s. GTA: More Analysis on the Affect of Intrinsic Modeling

The main difference between our proposed PRoPE and prior work GTA [3] is the introduce of camera intrinsic into the formulation. While GTA itself does not take into account camera intrinsics, as we have pointed out in Section 4.4, it is compatible with our proposed CamRay – which encodes the intrinsic at token level – to form a complete representation for camera conditioning. In this section, we run more thorough experiments focusing comparing PRoPE and GTA in differnet settings (with/without CamRay) as the camera conditioning techniques on both UniMatch for stereo depth

Encoding Method	PSNR↑	LPIPS↓	SSIM↑
w/o $\mathbf{D}_t^{\text{Proj}}$	16.04	0.505	0.509
w/o $\mathbf{D}_t^{\text{RoPE}}$	21.39	0.238	0.673
PRoPE	**21.78**	**0.211**	**0.692**

Table A.1: **Ablation Study on PRoPE.** $\mathbf{D}_t^{\text{Proj}}$ is crucial for encoding the relative camera information, and $\mathbf{D}_t^{\text{RoPE}}$ is also helpful to capture the relative patch coordinate. Experiments are conducted on RealEstate10K [12] with CamRay as input.

	3-view			6-view			9-view		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Zip-NeRF [70]	12.77	0.271	0.705	13.61	0.284	0.663	14.30	0.312	0.633
ZeroNeRF [71]	14.44	0.316	0.680	15.51	0.337	0.663	15.99	0.350	0.655
ReconFusion [72]	15.50	0.358	0.585	16.93	0.401	0.544	18.19	0.432	0.511
CAT3D [7]	16.62	0.377	0.515	17.72	0.425	0.482	18.67	0.460	**0.460**
CAT3D [7] + PRoPE	**16.93**	**0.382**	**0.505**	**18.01**	**0.443**	**0.479**	**18.98**	**0.474**	0.461

Table A.2: **Incorporating PRoPE into CAT3D.** Adding PRoPE to the CAT3D [7] multiview diffusion model yields improvements over the original model, with zero additional model parameters and negligible computational overhead.

Method	1×	3×	5×	7×
Plücker	19.89	21.03	20.75	20.32
GTA	15.77	18.14	18.12	18.29
GTA+CamRay	21.41	22.70	22.35	21.77
PRoPE	21.42	22.95	22.75	**22.40**
PRoPE+CamRay	**21.86**	**23.06**	**22.83**	**22.40**

PSNR (↑) across zoom-in levels.

Method	1×	3×	5×	7×
Plücker	0.608	0.694	0.732	0.755
GTA	0.512	0.626	0.683	0.721
GTA+CamRay	0.673	0.741	0.764	0.780
PRoPE	0.678	0.748	0.775	**0.794**
PRoPE+CamRay	**0.693**	**0.751**	**0.776**	**0.794**

SSIM (↑) across zoom-in levels.

Method	1×	3×	5×	7×
Plücker	0.327	0.272	0.308	0.345
GTA	0.641	0.488	0.439	0.403
GTA+CamRay	0.238	0.191	0.242	0.288
PRoPE	0.247	0.193	0.227	0.244
PRoPE+CamRay	**0.218**	**0.183**	**0.220**	**0.242**

LPIPS (↓) across zoom-in levels.

Table A.3: **Additional Novel View Synthesis Results on Out-of-distribution Intrinsics at Test Time.** Experiments are conducted with LVSM [8] on RealEstate10K [12] dataset with augmented intrinsics (1-3× zoom-in) as described in Section 4.3.

estimation in Table A.4 and LVSM for novel view synthesis in Table A.3, both with augmented intrinsics in training to highlight the benefits of PRoPE on intrinsic modeling.

The new results corroborate the benefits of PRoPE that were originally shown in the main paper. We find that PRoPE consistently outperforms GTA. Importantly, both PRoPE and PRoPE+CamRay outperform GTA+CamRay, despite the fact that these methods include the same information and are all invariant to the choice of global frame. This supports our finding that attention-level intrinsic conditioning (as in PRoPE) is important.

A.2.3 Qualitative Results for Stereo Depth Estimation

In Figure A.2 we show more qualitative results on the task of stereo depth estimation with UniMatch [14].

https://doi.org/10.52202/085713-0540

Method	1.0×	1.5×	2.0×	3.0×
UniMatch	0.234	0.228	0.238	0.299
+Plucker	-2.47%	-9.35%	-7.74%	-4.67%
+GTA	-28.83%	-33.48%	-32.15%	-12.55%
+GTA+CamRay	-29.90%	-35.25%	-32.42%	-13.65%
+PRoPE	**-33.75%**	**-36.66%**	**-33.85%**	-26.40%
+PRoPE+CamRay	-31.99%	-35.82%	-33.24%	**-28.21%**

abs_rel (↓) across zoom-in levels.

Method	1.0×	1.5×	2.0×	3.0×
UniMatch	0.357	0.376	0.408	0.613
+Plucker	-1.08%	-9.46%	-2.76%	-3.37%
+GTA	-31.77%	-41.52%	-41.16%	-11.88%
+GTA+CamRay	-33.54%	-44.06%	-42.34%	-13.09%
+PRoPE	**-45.13%**	**-46.79%**	**-44.53%**	-38.56%
+PRoPE+CamRay	-41.79%	-46.75%	-44.08%	**-39.88%**

sq_rel (↓) across zoom-in levels.

Method	1.0×	1.5×	2.0×	3.0×
UniMatch	1.059	1.031	1.035	1.188
+Plucker	+1.03%	-3.95%	-0.53%	+1.64%
+GTA	-19.55%	-26.12%	-25.68%	-4.50%
+GTA+CamRay	-20.41%	-27.44%	-25.72%	-6.12%
+PRoPE	**-28.64%**	**-29.79%**	**-28.25%**	-22.61%
+PRoPE+CamRay	-26.86%	-29.29%	-28.09%	**-23.74%**

rmse (↓) across zoom-in levels.

Method	1.0×	1.5×	2.0×	3.0×
UniMatch	0.322	0.322	0.336	0.421
+Plucker	+0.49%	-5.46%	-2.33%	+2.00%
+GTA	-23.29%	-30.33%	-30.03%	-3.24%
+GTA+CamRay	-24.08%	-31.64%	-30.23%	-4.34%
+PRoPE	**-32.01%**	**-33.56%**	**-32.67%**	-26.78%
+PRoPE+CamRay	-29.98%	-32.68%	-32.31%	**-27.71%**

rmse_log (↓) across zoom-in levels.

Table A.4: **Additional Stereo Depth Estimation Results on Out-of-distribution Intrinsics at Test Time.** Experiments are conducted with UniMatch [14]'s official code as described in Section 4.6 but with 1/8 of the training resources (2 GPUs x 50k steps).

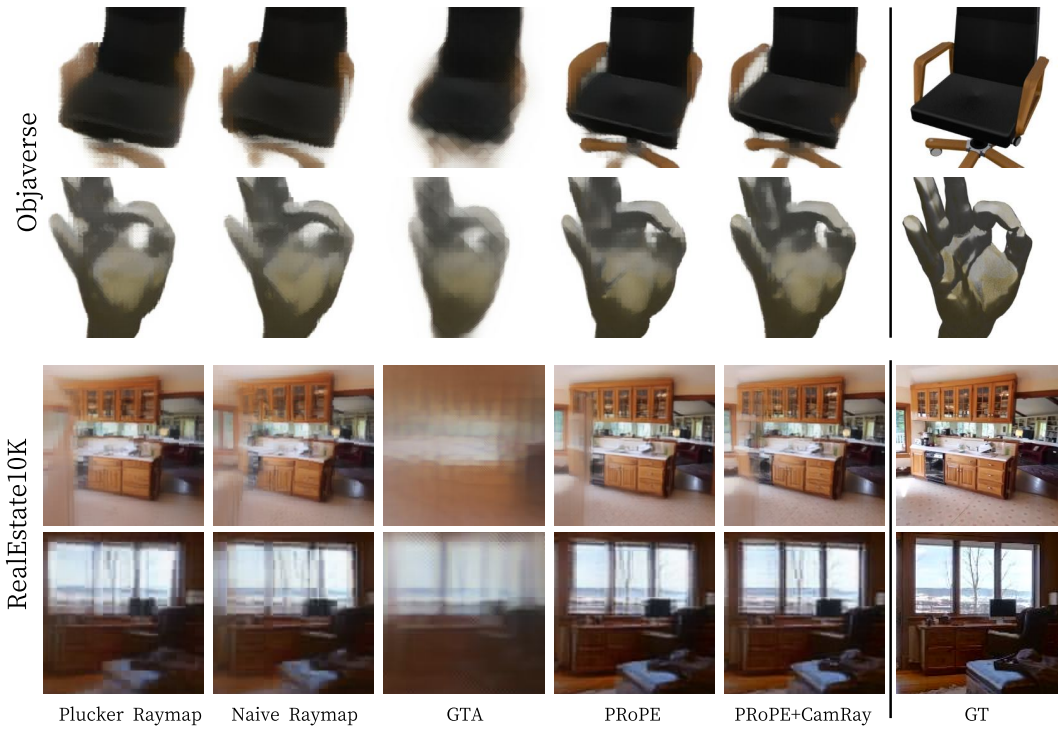


Figure A.1: **More Qualitative Results of Novel View Synthesis on RealEstate10K [12] and Objaverse [13].**

https://doi.org/10.52202/085713-0540

Figure A.2: **More Qualitative Results of Stereo Depth Estimation on RGBD [62], SUN3D [63] and Scenes11 [64].**

https://doi.org/10.52202/085713-0540

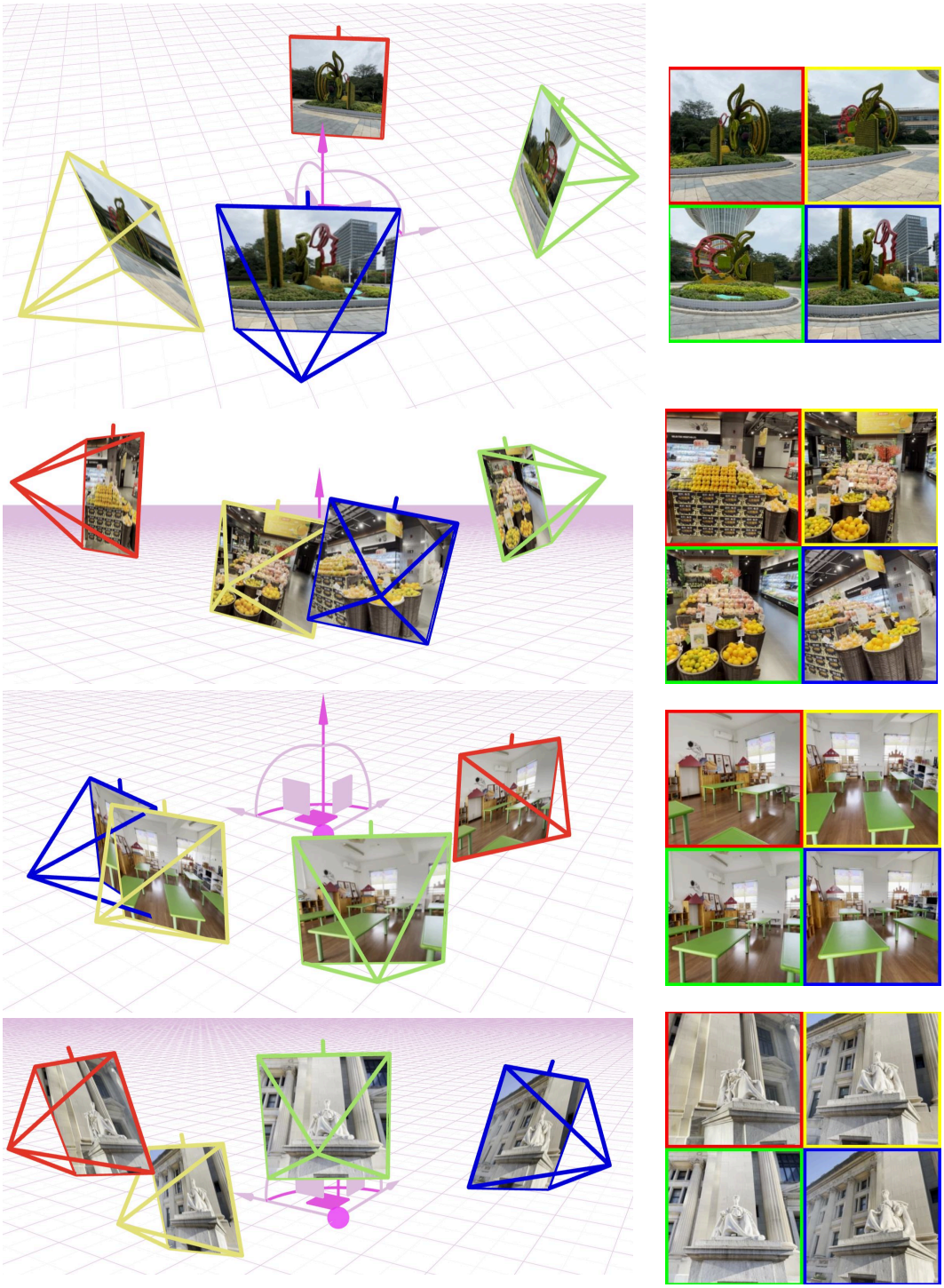


Figure A.3: **The Spatial Cognition Task.** The model takes multi-view images and corresponding camera information as input and aims to identify inconsistent image-camera pairs (yellow here). Understanding the cross-view relationships between images and cameras is crucial for this task.

https://doi.org/10.52202/085713-0540