DPPE: Rethinking Camera-Based Positional Encoding for Scaling Multi-View Transformers

Shun Kenney*
Keio University
Yokohama, Japan
shunkenney@keio.jp

Teppei Suzuki
SB Intuitions Corp.
Tokyo, Japan
teppei.suzuki@sbintuitions.co.jp

Abstract

The remarkable scalability of Transformers has expanded their application to 3D computer vision, where camera-aware positional encoding is crucial for providing spatial cues in multi-view geometry. Recent advancements have established the practice of using camera parameters—such as extrinsics or projection matrices—as relative positional encoding into the query, key, and value vectors of the attention mechanism. However, when scaling up the training recipe of novel view synthesis (NVS) models with the camera-based positional encoding, we observe a significant issue: model performance stagnates in the late stages of training.

In this paper, we investigate the cause of the performance bottleneck when scaling up and demonstrate that storing rotation and translation given by the positional encoding in the same dimensions of the value vector causes indeterminacy in their independent identification, hindering training scalability. To address this, we propose Decoupled Pose Positional Encoding (DPPE), a novel camera-based positional encoding that explicitly decouples rotation and translation. Extensive evaluations on NVS tasks demonstrate that DPPE enables stable long-term training even in scaled-up training setup. Furthermore, it exhibits superior generalization performance in extrapolation settings, such as handling an increased number of viewpoints and zoom-in scenarios. Code is available at https://github.com/sbintuitions/dppe.

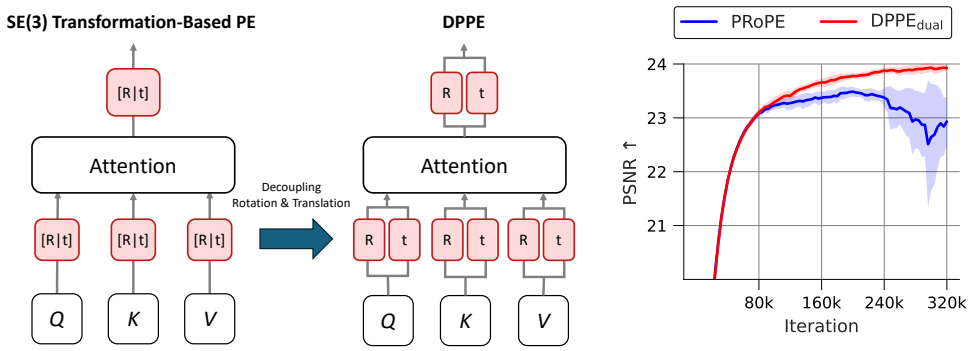
1 Introduction

The remarkable scalability of Transformers [1] has led to their dominance not only in natural language processing but also in numerous other areas, including the 3D computer vision field [2–6]. The attention function [1], which governs the information exchange between tokens in the Transformer, is implemented with positional encoding, which explicitly injects positional information into tokens for tasks where sequence order matters, since the attention function is itself permutation-invariant. There are various positional encoding methods [1, 3, 7–22], and in recent years, RoPE [10] is widely adopted because of its valuable properties such as the flexibility of sequence length and the capability of equipping the linear self-attention with relative position encoding.

For the tasks depending on multi-view geometry, camera-based positional encodings [19, 20, 3, 22, 23, 21, 24] are employed to provide spatial cues to tokens. In contrast to pixel-aligned approaches, which adds Plücker rays or raymaps to RGB [2, 25–29], several methods [3, 19, 20, 22, 24, 21, 23] have been proposed that apply camera-based transformations (*e.g.*, camera extrinsics or projection matrices) as relative positional encoding within attention. Notably, GTA [20] aligned the coordinate systems of tokens in the feature space by extending the camera-based geometric transformations to the values alongside the queries and keys. This geometry-aware mechanism enables GTA to

*Work was done during an internship at SB Intuitions.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) **Methods comparison** (b) **NVS results on MVImgNet2**

Figure 1: **DPPE prevents the late-stage performance degradation observed in PRoPE.** (a) Conceptual comparison between existing projection matrix–based positional encoding (left) and our proposed DPPE (right). Existing methods apply a single coupled SE(3) transformation $[R \mid t]$ uniformly to the queries (Q), keys (K), and values (V) before attention. In contrast, DPPE, our proposed method, decouples the rotation R and translation t, and injects them into Q, K, and V according to their distinct geometric roles. (b) Novel view synthesis on MVImgNet2: Results represent the mean over three seeds, and the shaded regions indicate the standard deviation. PRoPE's PSNR stagnates and degrades after 240k iterations, whereas DPPE remains stable.

significantly improve both the learning efficiency and overall performance of 3D vision tasks without requiring any additional learnable parameters, and the introduction of positional encoding to the values is widely used in several methods [3, 22, 24, 21, 23].

Despite the advantage of camera-based positional encoding and its empirically verified effectiveness, our validation on the novel view synthesis (NVS) task using a scaled-up setting, *i.e.*, a larger model trained on more data for an extended number of iterations, revealed a limitation: the performance of PRoPE, a natural extension of GTA, stagnates in the late stages of training (see Fig. 1(b)).

To investigate the cause of this performance stagnation, we systematically isolate the distinct factors within PRoPE [3], which is a projection matrix-based positional encoding, and analyze their specific impacts on the NVS task that strictly requires 3D spatial understanding. From those experiments, we obtain the following insight into the incorporation of camera information into the values in the attention function: *In camera-based positional encoding for the values, performance degrades if the rotation and shift components of the camera are coupled within the same dimensions.*

Based on these findings, we propose Decoupled Pose Positional Encoding (DPPE), a camera-based positional encoding in which rotation and translation are decoupled, as shown in Fig. 1(a). The proposed method enables stable training even in scaled-up configurations and demonstrates better generalization than baseline methods, including robustness to viewpoints beyond the training distribution and to zoom-in scenarios.

2 Related Work

Positional Encoding in Transformers. Absolute positional encoding (APE) directly injects per-token positional information, originating with the sinusoidal embeddings of [1] and extended via learned and conditional variants [8, 13]. Relative positional encoding (RPE) instead encodes pairwise offsets [7, 14–16, 9], among which Rotary Position Embedding (RoPE) [10] has become predominant due to its flexibility with variable sequence lengths and compatibility with linear attention, and has been adapted to vision and multimodal settings [11, 12, 17, 18].

Camera-Based Positional Encoding for Multi-View Transformers. Extending Transformers to multi-view 3D tasks that require explicit reasoning over camera geometry demands positional encodings capable of conveying such geometry. Existing methods fall into two broad categories. Pixel-aligned approaches concatenate Plücker rays or raymaps to the input RGB, providing per-pixel geometric cues without modifying attention; this strategy can be viewed as an APE in the multi-view

geometry setting [2, 25–29]. Attention-based approaches integrate camera transformations directly into attention as RPEs, and can be further classified by where in the attention mechanism the encoding is applied: (i) QK-only methods such as CaPE [19] inject camera information only into queries and keys, biasing attention weights without altering value features; (ii) QKV methods including GTA [20], PRoPE [3], and several recent extensions [22, 24] additionally transform values and outputs, aligning feature content across coordinate frames. Concurrent works also explore ray-based and unified relative formulations [21, 23].

Scope of This Work. GTA [20] introduced the principle of extending camera-based positional encoding from the queries and keys to the values and outputs (vo-pe), aligning feature coordinate systems across views. PRoPE [3] generalizes this to projection matrices and applies the same encoding function to both query-key positional encoding (qk-pe) and value-output positional encoding (vo-pe), as do subsequent works in this family. Despite their functionally distinct roles—qk-pe shaping attention similarity, vo-pe transforming aggregated feature content—this uniform treatment has not been systematically examined, particularly under scaled-up training. This paper investigates that question, focusing on the GTA/PRoPE family. We adopt PRoPE [3] as the exclusive baseline because (i) PRoPE generalizes GTA (the two coincide under fixed intrinsics, as confirmed in [3]), so findings on PRoPE extend to GTA; (ii) RayRoPE [21] introduces a learnable depth predictor and addresses ray-based geometry rather than qk-pe/vo-pe uniformity; (iii) methods for generative or video-retake tasks [22, 24] operate under fundamentally different paradigms.

3 Preliminaries

Here, we review the formulation of PRoPE, our baseline method. Given a projection matrix $\boldsymbol{P}_i \in \mathbb{R}^{4\times4}$ for the viewpoint of the image associated with the i-th token, the embedding matrix of the projective positional encoding in PRoPE is described as follows:

$$\gamma^{\mathrm{Proj}}_{d/4}(i) = \boldsymbol{I}_{d/4} \otimes \boldsymbol{P}_i, \tag{1}$$

where $\otimes$ denote the Kronecker product and $\boldsymbol{I}_k \in \{0,1\}^{k\times k}$ be the identity matrix, and $\gamma^{\mathrm{Proj}}_{d/4}(i) \in \mathbb{R}^{d\times d}$ is a block diagonal matrix. For clarity, the dimension of the identity matrix is indicated in the subscript of $\gamma^{\mathrm{Proj}}(i)$, but we omit it hereafter unless necessary.

Given the d dimensional queries, keys, and values, $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v} \in \mathbb{R}^{N\times d}$, where N is the number of tokens, the self-attention with the projective positional encoding for the i-th query is defined as follows:

$$\begin{aligned}\boldsymbol{o}'_i &= \gamma^{\mathrm{Proj}}(i)\,\mathrm{SelfAttn}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}, \gamma^{\mathrm{Proj}}) \\ &= \gamma^{\mathrm{Proj}}(i) \sum_j \alpha_{ij}\, \gamma^{\mathrm{Proj}}(j)^{-1} \boldsymbol{v}_j,\end{aligned} \tag{2}$$

where the attention weight α_{ij} is defined as:

$$\alpha_{ij} = \frac{\exp\left(s_{ij}/\sqrt{d}\right)}{\sum_l \exp\left(s_{il}/\sqrt{d}\right)}, \quad s_{ij} = \boldsymbol{q}_i^\top \gamma^{\mathrm{Proj}}(i)\, \gamma^{\mathrm{Proj}}(j)^{-1} \boldsymbol{k}_j. \tag{3}$$

Since $\gamma^{\mathrm{Proj}}(i)$ applied to the output of the self-attention is independent of the index j, it can be moved inside the summation and combined with $\gamma^{\mathrm{Proj}}(j)^{-1}$ applied to $\boldsymbol{v}_j$. This allows them to be computed together in advance as $\gamma^{\mathrm{Proj}}(i)\gamma^{\mathrm{Proj}}(j)^{-1} = \boldsymbol{I}_{d/4} \otimes \boldsymbol{P}_i {\boldsymbol{P}_j}^{-1}$, which serves as a coordinate system alignment [20, 3]. This role differs fundamentally from that of the positional encoding for query-key pairs, whose primary purpose is to inject biases reflecting token similarity.

PRoPE is used in conjunction with RoPE [10] as $\gamma^{\mathrm{PRoPE}} = \begin{bmatrix} \gamma^{\mathrm{Proj}}_{d/8} & 0 \\ 0 & \gamma^{\mathrm{RoPE}}_{d/4}(x,y) \end{bmatrix} \in \mathbb{R}^{d\times d}$, where $\gamma^{\mathrm{RoPE}}_{d/4}(x,y) \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ is the rotary embeddings based on the 2D patch coordinates x and y, *i.e.*, $\gamma^{\mathrm{RoPE}}_{d/4}(x,y) = \begin{bmatrix} \gamma^{\mathrm{RoPE}}_{d/4}(x) & 0 \\ 0 & \gamma^{\mathrm{RoPE}}_{d/4}(y) \end{bmatrix}$, where $\gamma^{\mathrm{RoPE}}_{d/4}(\cdot)$ constructs $\frac{d}{4} \times \frac{d}{4}$ rotary embeddings. We can define the original formulation of PRoPE by replacing γ^{Proj} in Eq. (2) with γ^{PRoPE}.

4 Investigating the Bottleneck of Camera-Based Positional Encoding

In this section, we analyze the functionality of PRoPE [3] using the novel view synthesis (NVS) task. We decompose the positional information constituting PRoPE into 2D patch coordinates, the camera intrinsics, and camera extrinsics to determine which parameters are responsible for the performance stagnation observed under scaled-up training.

Training Setup We evaluate all combinations of the parameters across three datasets: MVImgNet2 [30], RealEstate10K [31], and SpatialVidHQ [32], using standard RGB inputs. To investigate the effect of additional geometric input, we conduct an extended study on MVImgNet2 by concatenating Plücker raymaps to the input RGB in addition to a simple RGB input. For the model, we adopt the large-scale training configuration used in PRoPE [3], *i.e.*, a Transformer with 12 layers, a hidden dimension of 768, and an MLP dimension of 3072. We scale up the number of training iterations to 320k to accommodate the scaled-up dataset. The number of input views is set to two views to simplify the analysis. Detailed experimental settings are provided in App. C.1.

4.1 What is the Bottleneck in PRoPE?

In PRoPE, embeddings of 2D patch positions and the projection matrices are applied to both query-key and value-output pairs in self-attention. As a result, the positional encoding affects the attention pattern and the aggregated output, yet the precise role of each component within each encoding has not been thoroughly examined.

Therefore, to clarify the distinct properties of QK and VO positional encodings (qk-pe and vo-pe) and to assess the individual contributions of 2D patch coordinate and camera information, we evaluate all well-posed combinations of these encodings.

Results We show PSNR, SSIM [33], and VGG [34]-based LPIPS [35] of each combination on each dataset in Tab. 1. For RealEstate10K and SpatialVidHQ, no significant performance differences are observed across the various combinations of positional encodings. Similarly, on MVImgNet2, the choice of QK positional encoding does not yield substantial differences. In contrast, for the value-output positional encoding on MVImgNet2, we observe performance improvements when using the "2D" or "None" settings.

This result suggests that, on certain datasets, the positional encoding using projection matrix for the value-output pair negatively impacts performance. A similar trend is also observed in Tab. 2, which shows the metrics when Plücker raymaps are concatenated to the RGB input for MVImgNet2. This explicitly rules out the absence of geometric input as the cause of degradation, indicating that applying projection-matrix-based positional encoding to the value-output pair is itself harmful. We investigate the underlying cause in the next section.

4.2 Why Does the Projection Matrix in Value-Output Positional Encoding Cause Performance Degradation?

We explore the cause of the performance degradation on MVImgNet2 [30] by decomposing the projection matrix in vo-pe into its constituent elements: camera intrinsics $\boldsymbol{K}$, rotation $\boldsymbol{R}$, and translation $\boldsymbol{t}$. Note that for the query-key pair, we maintain the "2D + $\boldsymbol{P}$" configuration.

Results As shown in Tab. 3, each metric drops in the combination of $(\boldsymbol{K}, \boldsymbol{R}, \boldsymbol{t})$, which is projection matrix itself, and $(\boldsymbol{R}, \boldsymbol{t})$ settings. The common characteristic of these settings is that rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ are applied simultaneously.

To formulate a hypothesis for why the simultaneous application of $\boldsymbol{R}$ and $\boldsymbol{t}$ degrades performance, we expand VO positional encoding in Eq. (2). The relative transformation between i-th and j-th tokens is described as follows by extracting only the top four dimensions of the values $\boldsymbol{v}_j \in \mathbb{R}^d$:

$$\boldsymbol{P}_i \boldsymbol{P}_j^{-1} \boldsymbol{v}_{j,[1:4]} = \begin{pmatrix} \boldsymbol{K}_i \boldsymbol{R}_i \boldsymbol{R}_j^T \boldsymbol{K}_j^{-1} \boldsymbol{v}_{j,[1:3]} - \boldsymbol{K}_i \boldsymbol{R}_i \boldsymbol{R}_j^T \boldsymbol{t}_j \boldsymbol{v}_{j,4} + \boldsymbol{K}_i \boldsymbol{t}_i \boldsymbol{v}_{j,4} \\ \boldsymbol{v}_{j,4} \end{pmatrix}, \tag{4}$$

where $\boldsymbol{x}_{[i:j]} \in \mathbb{R}^{j-i+1}$ denotes elements from i-th to j-th of $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{K}_i \in \mathbb{R}^{3\times3}$, $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$, and $\boldsymbol{t}_i \in \mathbb{R}^3$ denote the camera intrinsic matrix, the rotation matrix, and the translation vector for

Table 1: **Quantitative results of various positional encoding combinations on various datasets.** "$\boldsymbol{P}$" and "2D" denote the projective positional encoding and the 2D patch positional encoding, respectively. For each metric, the best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively. We report PSNR↑/SSIM↑/LPIPS↓.

(a) MVImgNet2

qk-pe	vo-pe			
	None	2D	$\boldsymbol{P}$	2D+$\boldsymbol{P}$
None	-	-	-	23.10 / 0.699 / 0.211
2D	-	-	23.90 / 0.728 / 0.181	23.50 / 0.713 / 0.194
$\boldsymbol{P}$	-	24.21 / 0.739 / 0.174	-	23.98 / 0.731 / 0.183
2D+$\boldsymbol{P}$	24.09 / 0.733 / 0.172	24.22 / 0.738 / 0.171	23.81 / 0.725 / 0.184	23.73 / 0.725 / 0.183

(b) RealEstate10K

qk-pe	vo-pe			
	None	2D	$\boldsymbol{P}$	2D+$\boldsymbol{P}$
None	-	-	-	25.94 / 0.891 / 0.099
2D	-	-	26.01 / 0.892 / 0.099	26.20 / 0.894 / 0.097
$\boldsymbol{P}$	-	26.11 / 0.893 / 0.097	-	25.93 / 0.892 / 0.098
2D+$\boldsymbol{P}$	26.15 / 0.891 / 0.098	26.24 / 0.898 / 0.094	26.06 / 0.892 / 0.098	26.18 / 0.895 / 0.095

(c) SpatialVidHQ

qk-pe	vo-pe			
	None	2D	$\boldsymbol{P}$	2D+$\boldsymbol{P}$
None	-	-	-	19.65 / 0.649 / 0.248
2D	-	-	19.62 / 0.646 / 0.249	19.68 / 0.649 / 0.246
$\boldsymbol{P}$	-	19.58 / 0.647 / 0.251	-	19.66 / 0.651 / 0.249
2D+$\boldsymbol{P}$	19.58 / 0.645 / 0.248	19.58 / 0.644 / 0.250	19.64 / 0.647 / 0.247	19.65 / 0.648 / 0.251

Table 2: **Quantitative results for various positional encoding combinations with Plücker raymaps on MVImgNet2.** Notations and metrics follow Tab. 1.

qk-pe	vo-pe			
	None	2D	$\boldsymbol{P}$	2D+$\boldsymbol{P}$
None	23.78 / 0.713 / 0.190	23.95 / 0.720 / 0.183	23.83 / 0.715 / 0.194	23.99 / 0.726 / 0.187
2D	23.50 / 0.700 / 0.190	23.83 / 0.707 / 0.189	24.47 / 0.739 / 0.170	24.12 / 0.727 / 0.189
$\boldsymbol{P}$	24.40 / 0.738 / 0.174	24.70 / 0.749 / 0.170	24.18 / 0.727 / 0.177	24.29 / 0.734 / 0.176
2D+$\boldsymbol{P}$	24.60 / 0.745 / 0.165	24.64 / 0.744 / 0.165	24.07 / 0.727 / 0.182	24.28 / 0.733 / 0.180

the viewpoint corresponding to the i-th token, respectively. This equation indicates that rotation $\boldsymbol{R}_i$ and translation $\boldsymbol{t}_i$ are coupled within the same top three dimensions. For datasets such as RealEstate10K and SpatialVidHQ, where typically only one of $\boldsymbol{R}_i$ or $\boldsymbol{t}_i$ varies (*e.g.*, during a shift or pan), this coupling poses no issue, and no performance degradation is observed. However, given that performance degradation occurred solely on MVImgNet2, we consider the following hypothesis: *The coupling of rotation and translation hinders the modeling of scenes where these two components change simultaneously.* This hypothesis suggests that although vo-pe is originally intended for coordinate-system alignment, in practice it may be implicitly required to embed camera information in an identifiable form.

As formally shown in App. B, the projection matrix-based positional encoding is insufficient to identify the rotation and translation contributions at the per-token level. Since identifying camera

Table 3: **Results for various camera parameter combinations on MVImgNet2.** We evaluate the effects of each camera parameter, the intrinsics $\boldsymbol{K}$, the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t}$ in the value-output positional encoding (vo-pe). All values represent the mean over 3 seeds. For each metric, the best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively. Performance decreases whenever $\boldsymbol{R}$ and $\boldsymbol{t}$ are applied simultaneously.

vo-pe	$\boldsymbol{KRt}$	$\boldsymbol{Rt}$	$\boldsymbol{KR}$	$\boldsymbol{Kt}$	$\boldsymbol{K}$	$\boldsymbol{R}$	$\boldsymbol{t}$	None
PSNR↑	23.30	23.63	24.18	24.22	24.19	24.17	24.23	24.18
SSIM↑	0.711	0.723	0.737	0.738	0.736	0.737	0.739	0.736
LPIPS↓	0.201	0.189	0.173	0.172	0.174	0.175	0.171	0.173

parameters is essential for multi-view stereo tasks such as NVS, this non-identifiability forces the model to disambiguate camera parameters indirectly by aggregating contextual cues across tokens, complicating the learning process. Therefore, we propose a positional encoding that ensures per-token identifiability of camera parameters.

5 Method

Building on the hypothesis from Sec. 4.2, we propose Decoupled Pose Positional Encoding (DPPE), a camera-based positional encoding that retains all camera information while explicitly decoupling rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$. We introduce two variants: $\text{DPPE}_{\text{tAdd}}$, which explicitly partitions the feature space into dedicated dimensions for storing $\boldsymbol{R}$ and $\boldsymbol{t}$, and $\text{DPPE}_{\text{dual}}$, which uses the dual form of the projection matrix. As in PRoPE [3], we also use RoPE [10] for the 2D patch coordinates; we omit that below for brevity.

5.1 $\text{DPPE}_{\text{tAdd}}$: Decoupling via Additive Translation

$\text{DPPE}_{\text{tAdd}}$ explicitly decouples $\boldsymbol{R}$ and $\boldsymbol{t}$ by dividing the token dimensions into two equal halves and applying rotation- and translation-based encodings to each half, respectively. The embeddings of rotation-based positional encoding, $\gamma_{\text{DPPE}_{\text{tAdd},\boldsymbol{R}}}$ are defined using a transformation by the intrinsic matrix $\boldsymbol{K}_i \in \mathbb{R}^{3\times3}$ and the rotation matrix $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ for the i-th token as follows:

$$\gamma_{d/6}^{\text{DPPE}_{\text{tAdd},\boldsymbol{R}}}(i) = \boldsymbol{I}_{d/6} \otimes \boldsymbol{K}_i \boldsymbol{R}_i. \tag{5}$$

This embedding is then multiplied by the rotation-block components of queries, keys, values, and outputs as in Eq. (2). Importantly, the attention weights for the rotation block are computed solely from the rotation-block similarity, independently of the translation block (see App. A for the full self-attention formulation). The embeddings of translation-based positional encoding are defined as follows:

$$\gamma_{d/6}^{\text{DPPE}_{\text{tAdd},\boldsymbol{t}}} = \mathbf{1}_{d/6} \otimes \boldsymbol{t}_i, \tag{6}$$

where $\boldsymbol{t}_i \in \mathbb{R}^3$ denotes the translation vector for the i-th token and $\mathbf{1}_{d/6} \in \mathbb{R}^{d/6}$ is the all-ones column vector. Then, $\gamma^{\text{DPPE}_{\text{tAdd},\boldsymbol{t}}}$ is simply added to queries, keys, values, and outputs.

The detailed formulation of self-attention with $\text{DPPE}_{\text{tAdd}}$ can be found in App. A.

5.2 $\text{DPPE}_{\text{dual}}$: Decoupling via the Dual Projection Matrix

We also propose $\text{DPPE}_{\text{dual}}$, which applies the dual form of the projection matrix $\boldsymbol{P}_i^{-T} \in \mathbb{R}^{4\times4}$ [36] in place of $\boldsymbol{P}_i$ in Eq. (1): $\gamma^{\text{DPPE}_{\text{dual}}}(i) = \boldsymbol{I}_{d/4} \otimes \boldsymbol{P}_i^{-T}$. To examine its properties, we extract the top four dimensions of the value-output transformation, analogously to Eq. (4):

$$\boldsymbol{P}_i^{-T} \boldsymbol{P}_j^T \boldsymbol{v}_{j,[1:4]} = \begin{pmatrix} \boldsymbol{K}_i^{-T} \boldsymbol{R}_i \boldsymbol{R}_j^T \boldsymbol{K}_j^T \boldsymbol{v}_{j,[1:3]} \\ -\boldsymbol{t}_i^T \boldsymbol{R}_i \boldsymbol{R}_j^T \boldsymbol{K}_j^T \boldsymbol{v}_{j,[1:3]} + \boldsymbol{t}_j^T \boldsymbol{K}_j^T \boldsymbol{v}_{j,[1:3]} + \boldsymbol{v}_{j,4} \end{pmatrix} \tag{7}$$

Although translation $\boldsymbol{t}$ is not entirely decoupled from rotation $\boldsymbol{R}$, the first three dimensions of Eq. (7) contain no $\boldsymbol{t}$ terms, so the influence of $\boldsymbol{t}$ is isolated within the fourth dimension and can be uniquely

Table 4: **Effect of applying DPPE to different attention components on MVImgNet2.** (a)DPPE applied only to the value-output pair. (b) DPPE applied to both the query-key and value-output pairs. All values represent the mean over 3 seeds.

(a) Value-output only.

Method	PSNR↑	SSIM↑	LPIPS↓
PRoPE	22.91	0.696	0.217
$\text{DPPE}_{\text{tAdd}}$	**24.02**	**0.730**	0.180
$\text{DPPE}_{\text{dual}}$	23.91	0.725	**0.178**

(b) Both query-key and value-output.

Method	PSNR↑	SSIM↑	LPIPS↓
PRoPE	22.91	0.696	0.217
$\text{DPPE}_{\text{tAdd}}$	23.32	0.703	0.201
$\text{DPPE}_{\text{dual}}$	**23.92**	**0.726**	**0.178**

Table 5: **Comparison of camera-based positional encodings across datasets.** We evaluate DPPE and PRoPE on MVImgNet2, RealEstate10K, and SpatialVidHQ. All values represent the mean over 3 seeds. The best scores are shown in **bold**.

Method	MVImgNet2			RealEstate10K			SpatialVidHQ		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PRoPE [3]	22.91	0.696	0.217	24.14	0.851	0.137	19.21	0.626	0.268
$\text{DPPE}_{\text{dual}}$	23.92	0.726	**0.178**	24.16	0.850	0.138	19.26	0.629	0.266
$\text{DPPE}_{\text{tAdd}}$	**24.02**	**0.730**	0.180	**24.27**	**0.855**	**0.133**	**19.28**	**0.630**	**0.265**

identified there. We provide a formal proof of this property in App. B. Consequently, this design is expected to yield practically the same effect as being fully decoupled, and we verify its effectiveness in the next section.

6 Experiments

In this section, we verify the effectiveness of DPPE in the NVS task, especially for the scenes where camera rotation and translation change simultaneously. As we discussed in Sec. 2, we compare DPPE with PRoPE [3] as our baseline method, and we also compare DPPE with GTA [3] in App. D. We focus the main evaluation on NVS because it requires precise underlying 3D reconstruction and cannot be solved without accurate camera information, making it particularly well-suited for validating camera-based positional encodings. We also report the results on the multi-view depth estimation tasks in App. E.2, where DPPE likewise demonstrates equivalent to or better than the baseline.

Training Setup For the model, we adopt the large-scale training configuration used in PRoPE [3], a 12-layer Transformer with a hidden dimension of 768 and an MLP dimension of 3072, and train the model for 320k iterations with a variable number of multi-view inputs ranging from 2 to 4. All experiments were conducted using three different random seeds. Detailed experimental settings are provided in App. C.2.

6.1 Evaluation of DPPE as the Value-Output Positional Encoding

In this experiment, we applied $\text{DPPE}_{\text{tAdd}}$ and $\text{DPPE}_{\text{dual}}$ to the value-output pair and evaluate them on MVImgNet2 [30]. Note that the positional encoding for the query-key pair is the same as in PRoPE.

Results As shown in Tab. 4a, both DPPE variants outperform PRoPE in PSNR, SSIM, and LPIPS, demonstrating that decoupling rotation and translation in the value-output positional encoding leads to clear performance gains. The fact that $\text{DPPE}_{\text{dual}}$ also improves performance—despite not fully separating the rotation- and translation-storing dimensions—indicates that what matters is making camera parameters identifiable, rather than fully disjoint storage.

We also show the qualitative results in Fig. 2. Compared to PRoPE, DPPE successfully synthesizes more detailed textures. When the target viewpoint's camera information is ambiguous, recovering coarse structure may still be possible, but reproducing fine textures becomes highly challenging. This

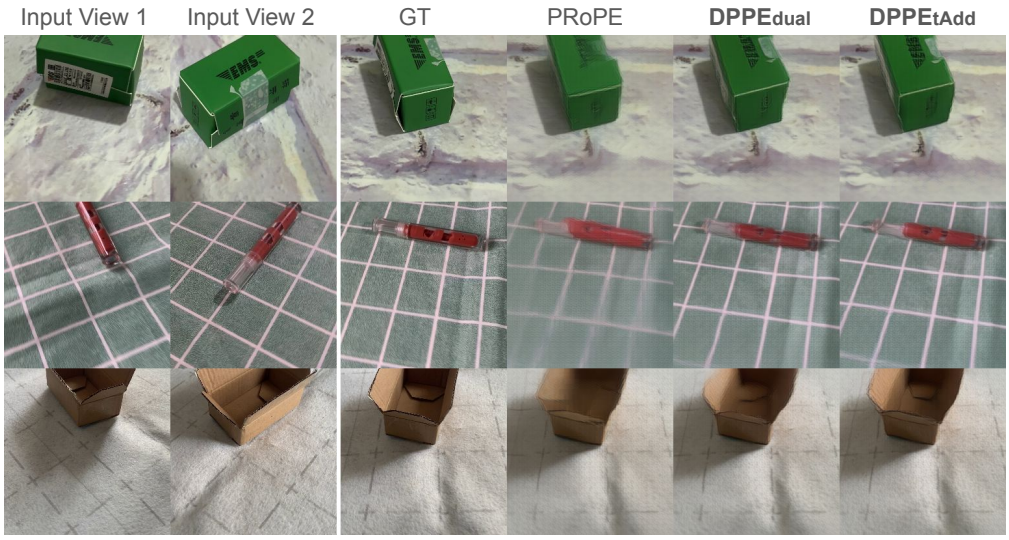


Figure 2: **Qualitative comparison of DPPE and PRoPE on MVImgNet2.** DPPE recovers more detailed textures compared to PRoPE, indicating an improved ability to disambiguate camera parameters. $\text{DPPE}_{\text{tAdd}}$ produces slightly finer details than $\text{DPPE}_{\text{dual}}$, reflecting its more explicit decoupling of rotation and translation.

observation suggests that DPPE facilitates the model in identifying camera parameters more easily. A comparison between the two variants further shows that $\text{DPPE}_{\text{tAdd}}$ reproduces slightly finer details than $\text{DPPE}_{\text{dual}}$, which we attribute to its more direct decoupling of rotation and translation.

6.2 Evaluation of DPPE as the Query-Key Positional Encoding

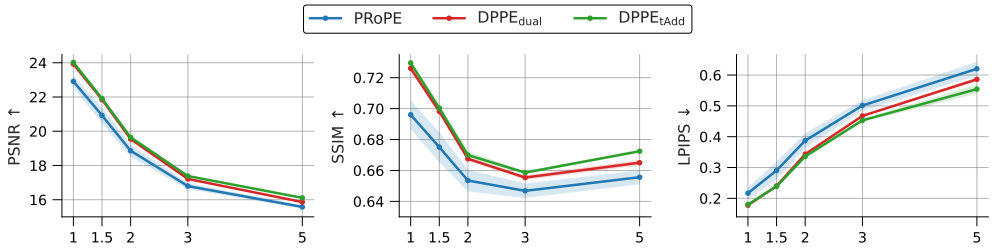
We evaluate DPPE by applying each variant to both query-key and value-output pairs on MVImgNet2. This allows us to verify whether DPPE works not only as an effect on the values and outputs, but also as a similarity bias in the computation of the attention matrix.

Results As shown in Tab. 4b, $\text{DPPE}_{\text{tAdd}}$ still outperforms PRoPE, but its performance is lower than when it is applied solely to the value-output pair. $\text{DPPE}_{\text{dual}}$, in contrast, maintains its performance under this setting. This asymmetric behavior suggests that the positional encodings for query-key and value-output pairs serve distinct roles.

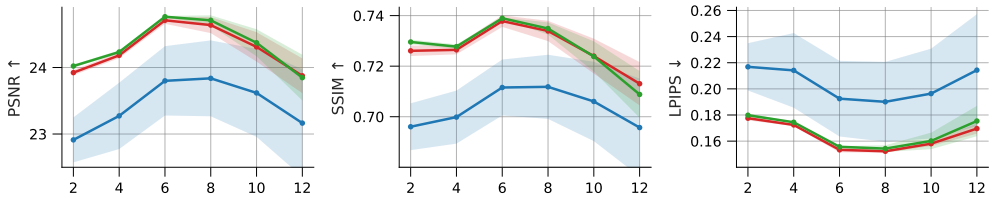
Unlike the value-output positional encoding, the positional encodings for queries and keys are expected to bias attention by token similarity; however, additive translation injection in $\text{DPPE}_{\text{tAdd}}$ introduces a term $\boldsymbol{t}_i^\top \boldsymbol{t}_j$ in the query-key dot product that depends only on absolute camera positions and grows with the scene scale. This scene-scale-dependent bias overwhelms the genuine similarity signal in scenes with large translations, undermining the intended role of the query-key positional encoding. Still, it is noteworthy that $\text{DPPE}_{\text{tAdd}}$ remains superior to PRoPE. In contrast, since $\text{DPPE}_{\text{dual}}$ uses the dual form of the projection matrix, it is expected to behave similarly to PRoPE, which uses the primal form. Thus, $\text{DPPE}_{\text{dual}}$ keeps its performance even when it is also applied to queries and keys. Based on these findings, in the subsequent experiments we configure $\text{DPPE}_{\text{tAdd}}$ to use PRoPE for the query-key positional encodings (and $\text{DPPE}_{\text{tAdd}}$ for the value-output positional encoding), while $\text{DPPE}_{\text{dual}}$ is used for both the query-key and value-output positional encodings.

6.3 Generalization and Extrapolation

In this experiment, we compare DPPE with PRoPE [3] on MVImgNet2, RealEstate10K, and SpatialVidHQ to evaluate generalization performance across datasets. We also evaluate the extrapolation performance—that is, the performance under conditions deviating from the training setup—regarding the number of multi-view inputs and increased focal lengths (i.e., zoom-in) on MVImgNet2.



(a) Evaluation results with zooming-in $\times$ [1.0, 5.0].



(b) Evaluation results with a variable number of multi-view inputs from 2 to 12.

Figure 3: **Evaluation under extrapolation settings.** We evaluate extrapolation capability with respect to the number of viewpoints and the zoom-in factor on MVImgNet2. DPPE constantly outperforms PRoPE in both settings. All values represent the mean over 3 seeds, and the shaded regions indicate the standard deviation.

Results As shown in Tab. 5, DPPE matches or outperforms PRoPE across all three datasets. On MVImgNet2 in particular, PSNR improves by more than 1.2 dB, confirming effectiveness on data where rotation and translation vary significantly together. The marginal gap on SpatialVidHQ is likely due to its predominantly simple camera trajectories, where the benefit of decoupling rotation and translation is inherently limited.

We also show the results for the zoom-in setting in Fig. 3(a) and the results for the various number of viewpoints setting in Fig. 3(b). In both settings, DPPE consistently outperforms PRoPE, indicating strong generalization to unseen test conditions.

7 Limitations

As shown in Tab. 5, the proposed method does not yield significant performance improvements over PRoPE on SpatialVidHQ. We attribute this to the predominantly simple camera trajectories in the dataset, where rotation and translation rarely change simultaneously and thus the rotation–translation coupling in PRoPE's value-output positional encoding does not manifest as a critical bottleneck. We acknowledge that such simple-trajectory regimes (*e.g.*, pure panning, dolly shots, or near-planar motions) cover a non-trivial portion of practical applications, including indoor walkthroughs, drone footage with stabilized gimbals, and many video generation scenarios. In these settings, the benefit of decoupling rotation and translation is inherently limited, and DPPE offers no clear advantage over existing methods. Nevertheless, as the field moves toward larger-scale and more diverse training corpora, scenes with complex coupled motion will become increasingly prevalent, and the limitations of PRoPE identified in this work are likely to emerge as a critical bottleneck.

8 Conclusion

In this study, we proposed Decoupled Pose Positional Encoding (DPPE), a camera-based positional encoding method with two variants: DPPE$_{tAdd}$ and DPPE$_{dual}$. Our method is motivated by the finding that storing rotation and translation in the same dimensions causes indeterminacy in their independent identification, hindering training scalability. DPPE$_{tAdd}$ explicitly assigns camera rotation and translation to separate dimensions, while DPPE$_{dual}$ uses the dual formulation of the projection matrix. DPPE$_{tAdd}$ achieves slightly better performance than DPPE$_{dual}$, but DPPE$_{dual}$ exhibits favorable camera-geometric properties and works effectively as a positional encoding for both the value-output

and the query-key pairs. DPPE enables stable scaled-up training on NVS and also achieves better view synthesis in extrapolation scenarios involving zoom-in and varying numbers of viewpoints, compared to PRoPE.

References

[1] 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.

[2] 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.

[3] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025.

[4] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[5] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[6] Haotong Lin, Sili Chen, Junhao 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.

[7] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, 2018.

[8] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[9] 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.

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

[11] 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.

[12] Chengcheng Wang, Jianyuan Guo, Hongguang Li, Yuchuan Tian, Ying Nie, Chang Xu, and Kai Han. Circle-rope: Cone-like decoupled rotary positional embedding for large vision-language models. *arXiv preprint arXiv:2505.16416*, 2025.

[13] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13569–13578, 2021.

[14] 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.

[15] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*, 2020.

[16] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[17] Haoyu Liu, Sucheng Ren, Tingyu Zhu, Peng Wang, Cihang Xie, Alan Yuille, Zeyu Zheng, and Feng Wang. Spiral rope: Rotate your rotary positional embeddings in the 2d plane. *arXiv preprint arXiv:2602.03227*, 2026.

[18] Yoav Gelberg, Koshi Eguchi, Takuya Akiba, and Edoardo Cetin. Extending the context of pretrained llms by dropping their positional embeddings. *arXiv preprint arXiv:2512.12167*, 2025.

[19] 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.

[20] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. *arXiv preprint arXiv:2310.10375*, 2023.

[21] Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. Rayrope: Projective ray positional encoding for multi-view attention. *arXiv preprint arXiv:2601.15275*, 2026.

[22] Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. *arXiv preprint arXiv:2512.07237*, 2025.

[23] Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. Urope: Universal relative position embedding across geometric spaces. *arXiv preprint arXiv:2604.18747*, 2026.

[24] Byeongjun Park, Byung-Hoon Kim, Hyungjin Chung, and Jong Chul Ye. Redirector: Creating any-length video retakes with rotary camera encoding. *arXiv preprint arXiv:2511.19827*, 2025.

[25] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[26] 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 *European Conference on Computer Vision*, pages 1–19. Springer, 2024.

[27] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024.

[28] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[29] 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.

[30] Xiaoguang Han, Yushuang Wu, Luyue Shi, Haolin Liu, Hongjie Liao, Lingteng Qiu, Weihao Yuan, Xiaodong Gu, Zilong Dong, and Shuguang Cui. Mvimgnet2. 0: A larger-scale dataset of multi-view images. *arXiv preprint arXiv:2412.01430*, 2024.

[31] 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.

[32] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025.

[33] 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): 600–612, 2004.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

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

[37] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna B Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *Advances in Neural Information Processing Systems*, 37:76232–76264, 2024.

[38] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

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

[40] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[41] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[42] Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[43] Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patino, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, et al. Smollm3: smol, multilingual, long-context reasoner. *Hugging Face Blog*, 2025.

[44] Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

[45] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

[46] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22389, 2024.

[47] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, et al. Megasynth: Scaling up 3d scene reconstruction with synthesized data. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16441–16452, 2025.

[48] 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. *arXiv preprint arXiv:2304.07193*, 2023.

[49] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

A Formulation of Self-Attention with DPPE$_{\text{tAdd}}$

We provide the explicit formulation of the self-attention with DPPE$_{\text{tAdd}}$. For clarity, we omit the RoPE component for 2D patch coordinates and focus solely on the rotation- and translation-based positional encodings.

A.1 Notation and Feature Decomposition

DPPE$_{\text{tAdd}}$ partitions the d-dimensional token features into two halves of equal dimension $d/2$, dedicated to rotation and translation respectively (we assume d is divisible by 6). For the i-th token, we write

$$\boldsymbol{q}_i = \begin{bmatrix} \boldsymbol{q}_i^R \\ \boldsymbol{q}_i^t \end{bmatrix}, \quad \boldsymbol{k}_i = \begin{bmatrix} \boldsymbol{k}_i^R \\ \boldsymbol{k}_i^t \end{bmatrix}, \quad \boldsymbol{v}_i = \begin{bmatrix} \boldsymbol{v}_i^R \\ \boldsymbol{v}_i^t \end{bmatrix} \in \mathbb{R}^d, \tag{8}$$

where the superscripts R and t denote the first $d/2$ dimensions (rotation block) and the last $d/2$ dimensions (translation block), respectively. The positional encodings are

$$\gamma^R(i) := \gamma^{\mathrm{DPPE}_{\mathrm{tAdd},R}}(i) = I_{d/6} \otimes (K_i R_i) \in \mathbb{R}^{(d/2)\times(d/2)}, \tag{9}$$

$$\gamma^t(i) := \gamma^{\mathrm{DPPE}_{\mathrm{tAdd},t}}(i) = \mathbf{1}_{d/6} \otimes \boldsymbol{t}_i \in \mathbb{R}^{d/2}, \tag{10}$$

where $\mathbf{1}_{d/6} \in \mathbb{R}^{d/6}$ is the all-ones column vector.

A.2 Block-wise Attention Computation

Since γ^R and γ^t act on disjoint dimension blocks, DPPE$_{\text{tAdd}}$ computes the attention score as a *concatenation* of two block-wise scores rather than as their sum:

$$\boldsymbol{s}_{ij} \;=\; \begin{bmatrix} s_{ij}^R \\ s_{ij}^t \end{bmatrix} \in \mathbb{R}^2, \tag{11}$$

$$s_{ij}^R \;=\; (\boldsymbol{q}_i^R)^\top \gamma^R(i)\, \gamma^R(j)^{-1} \boldsymbol{k}_j^R \,/\, \sqrt{d/2}, \tag{12}$$

$$s_{ij}^t \;=\; \big(\boldsymbol{q}_i^t + \gamma^t(i)\big)^\top \big(\boldsymbol{k}_j^t + \gamma^t(j)\big) \,/\, \sqrt{d/2}. \tag{13}$$

Block-wise softmax. The attention weights for the two blocks are computed by independent softmax normalization over the source token index, so that each block has its own attention pattern:

$$\alpha_{ij}^R \;=\; \frac{\exp\big(s_{ij}^R\big)}{\sum_{j'} \exp\Big(s_{ij'}^R\Big)}, \qquad\qquad \alpha_{ij}^t \;=\; \frac{\exp\big(s_{ij}^t\big)}{\sum_{j'} \exp\Big(s_{ij'}^t\Big)}. \tag{14}$$

Block-wise output. The output $\boldsymbol{o}_i' = [\,(\boldsymbol{o}_i')^R\,;\,(\boldsymbol{o}_i')^t\,]$ is computed block-wise using the corresponding attention weights:

$$(\boldsymbol{o}_i')^R \;=\; \gamma^R(i) \sum_j \alpha_{ij}^R\, \gamma^R(j)^{-1} \boldsymbol{v}_j^R, \tag{15}$$

$$(\boldsymbol{o}_i')^t \;=\; \gamma^t(i) + \sum_j \alpha_{ij}^t \big(\boldsymbol{v}_j^t - \gamma^t(j)\big). \tag{16}$$

This block-wise decoupling ensures that the rotation pattern α^R depends only on rotation-related similarities (via γ^R) and the translation pattern α^t depends only on translation-related similarities (via γ^t); the two computations never share information through the attention weights.

A.3 Per-Token Value Transformation

The per-token value-side contributions used in App. B are recovered as follows. For the rotation block,

$$\gamma^R(i)\, \gamma^R(j)^{-1} \boldsymbol{v}_j^R \;=\; \big(I_{d/6} \otimes K_i R_i R_j^\top K_j^{-1}\big)\, \boldsymbol{v}_j^R, \tag{17}$$

which depends on (R_i, R_j, K_i, K_j) but not on $(\boldsymbol{t}_i, \boldsymbol{t}_j)$. For the translation block, since $\sum_j \alpha_{ij}^t = 1$,

$$(\boldsymbol{o}_i')^t \;=\; \sum_j \alpha_{ij}^t \big(\boldsymbol{v}_j^t + \mathbf{1}_{d/6} \otimes (\boldsymbol{t}_i - \boldsymbol{t}_j)\big), \tag{18}$$

so that the per-token contribution is $\boldsymbol{v}_j^t + \mathbf{1}_{d/6} \otimes (\boldsymbol{t}_i - \boldsymbol{t}_j)$, depending on $(\boldsymbol{t}_i, \boldsymbol{t}_j)$ but not on (R_i, R_j). The disjoint dependence is the basis of the injectivity proof in Prop. 2.

B Identifiability Analysis of Camera Parameters in the Value-Output Positional Encoding

This appendix formally shows that PRoPE's value-output positional encoding is non-injective in $(\boldsymbol{R}, \boldsymbol{t})$ at the per-token level, while DPPE restores identifiability. We use the notation of Sec. 4: $\boldsymbol{P}_i = \boldsymbol{K}_i[\boldsymbol{R}_i \mid \boldsymbol{t}_i]$ is the projection matrix lifted to $\mathbb{R}^{4\times4}$, and $\boldsymbol{v}_{j,[1:3]}$, $\boldsymbol{v}_{j,4}$ denote the first three and fourth entries of $\boldsymbol{v}_j$. The analysis concerns the top four dimensions; the remaining blocks (cf. Eq. (1)) follow by identical construction.

B.1 Non-Identifiability of PRoPE's Value-Output Positional Encoding

Proposition 1 (Non-identifiability of PRoPE's value-output positional encoding) *Let $\boldsymbol{y}$ denote the top three dimensions of PRoPE's value-output positional encoding output (Eq. (4)). For any $\boldsymbol{v}_j$ with $\boldsymbol{v}_{j,4} \neq 0$ and any tangent perturbation $\delta\boldsymbol{R}_i$, the translation perturbation*

$$\delta\boldsymbol{t}_i = -\delta\boldsymbol{R}_i \boldsymbol{R}_j^\top \left(\frac{\boldsymbol{K}_j^{-1} \boldsymbol{v}_{j,[1:3]}}{\boldsymbol{v}_{j,4}} - \boldsymbol{t}_j \right) \tag{19}$$

satisfies $\delta\boldsymbol{y} = \mathbf{0}$.

Proof 1 *Differentiating Eq. (4) with respect to $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ gives $\delta\boldsymbol{y} = \boldsymbol{K}_i \delta\boldsymbol{R}_i \boldsymbol{R}_j^\top \big(\boldsymbol{K}_j^{-1}\boldsymbol{v}_{j,[1:3]} - \boldsymbol{t}_j \boldsymbol{v}_{j,4}\big) + \boldsymbol{K}_i \delta\boldsymbol{t}_i \boldsymbol{v}_{j,4}$. Since $\boldsymbol{K}_i$ is invertible and $\boldsymbol{v}_{j,4} \neq 0$, setting $\delta\boldsymbol{y} = \mathbf{0}$ yields Eq. (19).*

The compensating $\delta\boldsymbol{t}_i$ depends on $\boldsymbol{v}_j$ through both $\boldsymbol{v}_{j,[1:3]}$ and $\boldsymbol{v}_{j,4}$, so no fixed downstream transformation can recover $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ from a single token; disambiguation requires aggregating across tokens with diverse value content.

B.2 Identifiability of DPPE

Proposition 2 (DPPE$_{\text{tAdd}}$ is injective) *In DPPE$_{tAdd}$, rotation acts on the upper half via $\boldsymbol{KR}$ (Eq. (5)) and translation acts additively on the lower half (Eq. (6)). Hence rotation and translation perturbations affect disjoint output coordinates and cannot cancel; the map $(\boldsymbol{R}, \boldsymbol{t}) \mapsto T(\boldsymbol{R}, \boldsymbol{t}; \boldsymbol{v}_j)$ is injective for any $\boldsymbol{v}_j$ with $\boldsymbol{v}_{j,[1:d/4]} \neq \mathbf{0}$.*

Proof 2 *From Eq. (5), the rotation block of the value-domain output (the upper half) is*

$$T_R = \big(\boldsymbol{I}_{d/6} \otimes \boldsymbol{K}_i \boldsymbol{R}_i \boldsymbol{R}_j^\top \boldsymbol{K}_j^{-1}\big) \boldsymbol{v}_j^R, \tag{20}$$

which depends on $(\boldsymbol{R}_i, \boldsymbol{R}_j)$ but not on $(\boldsymbol{t}_i, \boldsymbol{t}_j)$. From Eq. (6), the translation block (the lower half) is

$$T_t = \boldsymbol{v}_j^t + \boldsymbol{I}_{d/6} \otimes (\boldsymbol{t}_i - \boldsymbol{t}_j), \tag{21}$$

which depends on $(\boldsymbol{t}_i, \boldsymbol{t}_j)$ but not on $(\boldsymbol{R}_i, \boldsymbol{R}_j)$. Differentiating yields

$$\delta T_R = \boldsymbol{I}_{d/6} \otimes \boldsymbol{K}_i \big(\delta\boldsymbol{R}_i \boldsymbol{R}_j^\top + \boldsymbol{R}_i \delta\boldsymbol{R}_j^\top\big) \boldsymbol{K}_j^{-1} \boldsymbol{v}_j^R, \tag{22}$$

$$\delta T_t = \boldsymbol{I}_{d/6} \otimes (\delta\boldsymbol{t}_i - \delta\boldsymbol{t}_j). \tag{23}$$

Since T_R and T_t occupy disjoint coordinate blocks, their perturbations cannot cancel one another. In particular, any non-zero rotation perturbation produces a non-zero δT_R (provided $\boldsymbol{v}_j^R \neq \mathbf{0}$ and $\boldsymbol{K}_i, \boldsymbol{K}_j$ invertible), which no translation perturbation can compensate; symmetrically for translations.

Proposition 3 (DPPE$_{\text{dual}}$ is sequentially identifiable) *The first three of the top four dimensions of DPPE$_{dual}$'s output (Eq.* (7)*) depend only on rotations and intrinsics, with no translation contribution. Hence rotations are uniquely identifiable from these dimensions; conditional on rotations, the fourth dimension is an affine function of $(\boldsymbol{t}_i, \boldsymbol{t}_j)$ with known coefficients, making translations identifiable.*

Proof 3 *From Eq.* (7)*, the top three output dimensions are*

$$\boldsymbol{y}_R := \boldsymbol{K}_i^{-\top} \boldsymbol{R}_i \boldsymbol{R}_j^\top \boldsymbol{K}_j^\top \boldsymbol{v}_{j,[1:3]}, \tag{24}$$

which depends on $(\boldsymbol{R}_i, \boldsymbol{R}_j)$ but contains no $(\boldsymbol{t}_i, \boldsymbol{t}_j)$ terms. Differentiating with respect to translations yields $\delta \boldsymbol{y}_R = \boldsymbol{0}$, so no translation perturbation can affect $\boldsymbol{y}_R$, and rotations are identifiable from $\boldsymbol{y}_R$ independently of translations.

Conditional on $(\boldsymbol{R}_i, \boldsymbol{R}_j)$ being fixed, the fourth output dimension becomes an affine function of $(\boldsymbol{t}_i, \boldsymbol{t}_j)$:

$$y_t = -\boldsymbol{t}_i^\top \boldsymbol{a} + \boldsymbol{t}_j^\top \boldsymbol{b} + \boldsymbol{v}_{j,4}, \tag{25}$$

where $\boldsymbol{a} := \boldsymbol{R}_i \boldsymbol{R}_j^\top \boldsymbol{K}_j^\top \boldsymbol{v}_{j,[1:3]}$ and $\boldsymbol{b} := \boldsymbol{K}_j^\top \boldsymbol{v}_{j,[1:3]}$ are known constants once rotations and intrinsics are fixed. Provided $\boldsymbol{v}_{j,[1:3]} \neq \boldsymbol{0}$ and the intrinsics are invertible, both $\boldsymbol{a}$ and $\boldsymbol{b}$ are non-zero, so y_t varies non-trivially with $(\boldsymbol{t}_i, \boldsymbol{t}_j)$ and the translation contribution is identifiable from y_t.

Connection to training dynamics. The propositions above are per-token statements: for any single edge (i, j), PRoPE's value-output transformation cannot uniquely identify $(\boldsymbol{R}_i, \boldsymbol{t}_i)$, whereas DPPE can. The full PRoPE model can in principle disambiguate $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ by aggregating across edges via the weighted sum $\sum_j \alpha_{ij} \gamma^{\mathrm{Proj}}(j)^{-1} \boldsymbol{v}_j$ in Eq. (2). We hypothesize, however, that the absence of per-token identifiability shifts the burden of disambiguation onto the rest of the network and manifests as the late-stage training degradation of PRoPE in Fig. 4. DPPE removes this burden by providing per-token identifiability of $(\boldsymbol{R}_i, \boldsymbol{t}_i)$. Furthermore, DPPE$_{\text{tAdd}}$ enforces a stronger form of decoupling at the architectural level: the rotation and translation blocks maintain independent attention patterns α_{ij}^R and α_{ij}^t (App. A), so that translation-induced biases in attention (*e.g.*, spurious bonds between distant tokens do not contaminate the aggregation of the rotation block).

C Experiment Details

For all tables in Secs. 4 and 6, we present the values smoothed by applying an exponential moving average (EMA) with a decay factor of 0.75 to the evaluation results from checkpoints saved every 5k iterations between 275k and 320k iterations in order to mitigate evaluation fluctuations across iterations and ensure the reliability of the results. Furthermore, Fig. 1(b) shows the smoothed values using the same EMA with a decay factor of 0.75 across all evaluation steps from the start of training.

EMA mitigates iteration-wise variance while accurately reflecting the model's performance up to the final iteration. Therefore, in this study, we determined and applied an decay factor for each experiment to smooth fluctuations while properly capturing the true final performance.

C.1 Training Details of Exploring Camera-based Positional Encoding in Sec. 4

For the exploratory experiments on camera-based positional encodings, we generally followed the default settings provided in the official PRoPE repository [3]. The image resolution is fixed at 256×256, and the world batch size is set to 64. To facilitate analysis and reduce computational costs in these experiments, we retain the default number of input views 2 and target views 1.

We adopt a warmup-stable scheduler, which maintains a learning rate of 5×10^{-4} after a 500-step warmup period. This approach omits the decay phase from the warmup-stable-decay scheduler [37], which has recently become the standard for the long-term training of large language models [38–44]. Training was conducted via distributed learning on eight H100 GPUs, taking approximately 20 hours per model.

C.2 Training Details in Sec. 6

The training settings for these experiments are fundamentally identical to those described in Sec. 4. However, to evaluate more practical capabilities, we trained the models by randomly varying the

Table 6: Angular changes in camera viewing direction between consecutive frames in the training sets (degrees).

Dataset	Mean	Median	95th percentile
MVImgNet2	11.27	12.58	17.15
RealEstate10K	0.20	0.14	0.59
SpatialVidHQ	0.72	0.40	2.50

number of input views between 2 and 4, while keeping the number of target views fixed at 1. As a result of this modification, the training time increased to approximately 24 hours per model.

C.3 Multi-View Sampling Strategy

For the novel view synthesis (NVS) task, it is essential that a certain degree of subject overlap exists among the sampled images.

Regarding RealEstate10K, we used the configurations exactly as provided, since the sampling index ranges for training, along with the IDs and image indices for the evaluation data, are available in the official PRoPE [3] repository.

For MVImgNet2 and SpatialVidHQ, we design an image sampling method because they do not provide the sampling settings: An initial image is randomly sampled, and the selection probability is configured to increase as the angular and spatial distances from the reference camera position increase. Regarding the angle, we introduce an additional constraint: only images located within 60 degrees of any previously sampled camera are considered as candidates.

Consequently, for MVImgNet2, this approach enables the extraction of viewpoints with appropriate subject overlap while maintaining an angular difference of 60 degrees or less. For SpatialVidHQ, this ensures that spatially distant images are appropriately sampled for camera shift, while images maintaining subject overlap are suitably selected for camera panning.

During training, this sampling process is iterated to acquire multiple images. A randomly selected non-edge camera among the sampled images is assigned as the target view, while the remaining images are used as input views. For evaluation, we prepare a split of approximately 6,000 instances in advance. By applying this sampling algorithm beforehand to fix the indices, we ensured consistent evaluations on the identical dataset across every evaluation iteration.

C.4 Descriptive Camera-Motion Statistics

To partially assess the motion-based hypothesis in Sec. 4.2, we analyzed the training-set camera trajectories. Since the camera poses are estimated without metric scale, translation magnitudes are not directly comparable across datasets. We therefore measured the angular difference between camera viewing directions of consecutive frames, pooling all consecutive-frame pairs across videos.

As shown in Tab. 6, MVImgNet2 exhibits substantially larger frame-to-frame changes in viewing direction than RealEstate10K and SpatialVidHQ. While these statistics reveal differences in camera motion, they do not directly characterize joint rotation–translation variation or establish a causal explanation for the performance differences.

D Performance Degradation in Scaled-up Training

When performance begins to degrade during training, selecting the best-performing checkpoint (*i.e.*, early stopping) is a common strategy to maintain high performance. Fig. 4, plotting the raw metrics on MVImgNet2 from the experiments in Tab. 5 and an additionally conducted experiment for GTA [20], demonstrates that the performance of PRoPE begins to drop around 240k iterations, with PSNR and LPIPS peaking at 184k iterations, and SSIM peaking at 200k iterations. Accordingly, we evaluate PRoPE using checkpoints at 184k, 200k, and 240k iterations. Similarly, GTA—which is PRoPE without intrinsic parameter information and with coupled rotation and translation—exhibits degradation similar to PRoPE in the later stages of training, as also indicated in the $(\boldsymbol{R}, \boldsymbol{t})$ row of

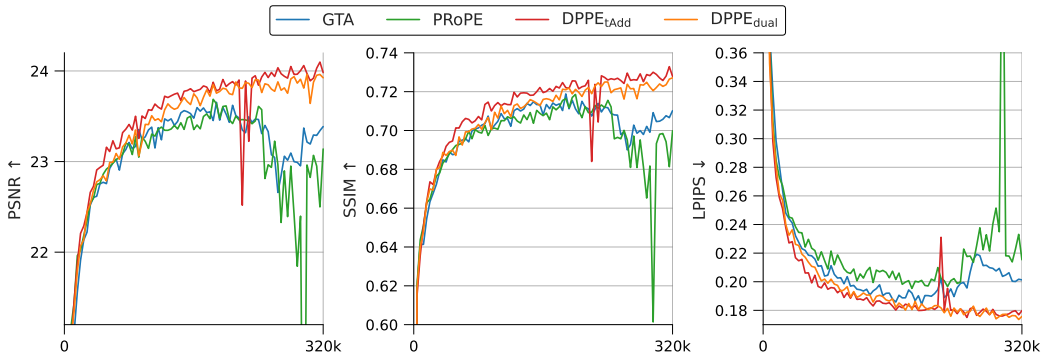


Figure 4: **Evaluation results of novel view synthesis on the MVImgNet2 dataset at each step.** While the performance of GTA [20] and PRoPE [3] degrades, DPPE consistently improves the performance.

Table 7: Comparison using the early stopping.

Method	PSNR↑	SSIM↑	LPIPS↓
GTA$_{@188k}$	23.65	0.719	0.187
GTA$_{@196k}$	23.60	0.716	0.184
GTA$_{@240k}$	23.46	0.712	0.196
GTA$_{@320k}$	23.39	0.710	0.201
PRoPE$_{@184k}$	23.67	0.715	0.195
PRoPE$_{@200k}$	23.61	0.719	0.197
PRoPE$_{@240k}$	23.61	0.716	0.199
PRoPE$_{@320k}$	23.14	0.700	0.215
DPPE$_{dual@320k}$	23.92	0.727	**0.176**
DPPE$_{tAdd@320k}$	**23.98**	**0.728**	0.180

Tab. 3. For GTA, PSNR and SSIM peak at 188k iterations, and LPIPS peaks at 196k iterations. The evaluation results using these optimal checkpoints are presented in Tab. 7.

As shown in Tab. 7, even when using their respective best checkpoints, both PRoPE and GTA fall short of the performance achieved by DPPE at the final iteration. Furthermore, determining the training schedule based solely on the behavior of the positional encoding—which is merely one component of the entire architecture—prevents the overall model from realizing its full potential and is therefore impractical.

E Additional Experiments

E.1 Comprehensive Dataset Extension and Generalization to Ray-Space Encodings

To verify the robustness and generalizability of the proposed decoupling framework, we extend our evaluation across two distinct axes: dataset diversity and alternative geometric encodings.

Evaluation Across Six Diverse Datasets We evaluate the novel view synthesis (NVS) performance of DPPE against existing baselines across six datasets featuring highly varied scene structures, scale scales, and motion dynamics: MVImgNet2 [30], RealEstate10K [31], SpatialVidHQ [32], CO3D [45], WildRGBD [46], and MegaSynth [47]. This broad suite tests the models under both object-centric constraints (e.g., MVImgNet2, CO3D) and complex trajectory-based environments (e.g., RealEstate10K, SpatialVidHQ).

Generalization to Token-Level Ray Spaces (UCPE) We investigate whether the benefits of decoupling apply beyond image-level camera matrices. We implement our decoupling strategy on Unified Camera Positional Encoding (UCPE) [22], a formulation that instantiates a ray space for each

Table 8: **Comparison of camera-based positional encodings across datasets.** We evaluate DPPE and PRoPE on various datasets. All values represent the mean over 3 seeds. For each metric, the best, second-best, and third-best results are highlighted in red and orange, respectively, within each block separated by dashed lines.

Dataset	Metric	Plücker	GTA	PRoPE	$\text{DPPE}_{\text{dual}}$	$\text{DPPE}_{\text{tAdd}}$	UCPE	$\text{UCPE}_{\text{dual}}$	$\text{UCPE}_{\text{tAdd}}$
MVImgNet2	PSNR↑	23.07	23.28	22.91	23.92	24.02	23.27	23.93	23.95
	SSIM↑	0.682	0.706	0.696	0.726	0.730	0.704	0.726	0.725
	LPIPS↓	0.224	0.203	0.217	0.178	0.180	0.205	0.179	0.176
RealEstate10K	PSNR↑	26.33	24.15	24.14	24.16	24.27	23.05	23.66	23.75
	SSIM↑	0.849	0.847	0.851	0.850	0.855	0.831	0.842	0.843
	LPIPS↓	0.135	0.141	0.137	0.138	0.133	0.158	0.146	0.146
SpatialVidHQ	PSNR↑	19.23	19.00	19.21	19.26	19.28	18.88	18.76	19.13
	SSIM↑	0.610	0.613	0.626	0.629	0.630	0.612	0.606	0.624
	LPIPS↓	0.278	0.278	0.268	0.266	0.265	0.286	0.294	0.274
CO3D	PSNR↑	21.54	20.06	19.91	20.55	20.81	20.56	20.87	20.92
	SSIM↑	0.688	0.689	0.686	0.702	0.709	0.703	0.712	0.711
	LPIPS↓	0.291	0.290	0.295	0.256	0.245	0.260	0.244	0.245
WildRGBD	PSNR↑	16.39	16.88	17.20	17.33	17.27	17.13	17.42	17.23
	SSIM↑	0.526	0.553	0.564	0.570	0.565	0.558	0.571	0.563
	LPIPS↓	0.422	0.363	0.325	0.321	0.338	0.362	0.317	0.345
MegaSynth	PSNR↑	18.07	18.18	18.20	18.37	18.33	17.78	18.44	18.31
	SSIM↑	0.513	0.538	0.540	0.549	0.547	0.512	0.556	0.549
	LPIPS↓	0.354	0.335	0.333	0.320	0.324	0.371	0.317	0.323

patch token rather than applying a global camera matrix per image. Following the principles detailed in App. A, we construct $\text{UCPE}_{\text{dual}}$ and $\text{UCPE}_{\text{tAdd}}$ to assess if separating rotation and translation components resolves late-stage training degradation in token-level ray encodings.

Comparison with Explicit Geometric Baselines For comprehensive reference, we compare these attention-based positional encodings against a non-attention baseline using Plücker coordinates. This baseline embeds local ray geometry directly into the token features without modifying the attention matrix via projection operations.

As reported in Tab. 8 and Fig. 5, $\text{DPPE}_{\text{tAdd}}$ and $\text{DPPE}_{\text{dual}}$ consistently outperform or match the best baseline performance across all datasets, validating the core identifiability analysis. Crucially, the performance gains observed when transitioning from UCPE to $\text{UCPE}_{\text{dual}}$ and $\text{UCPE}_{\text{tAdd}}$ mirror the improvements seen between PRoPE and DPPE. This demonstrates that architectural decoupling is a generalizable solution for stabilizing camera-based positional encodings, irrespective of whether the geometry is parameterized at the image level or the token level.

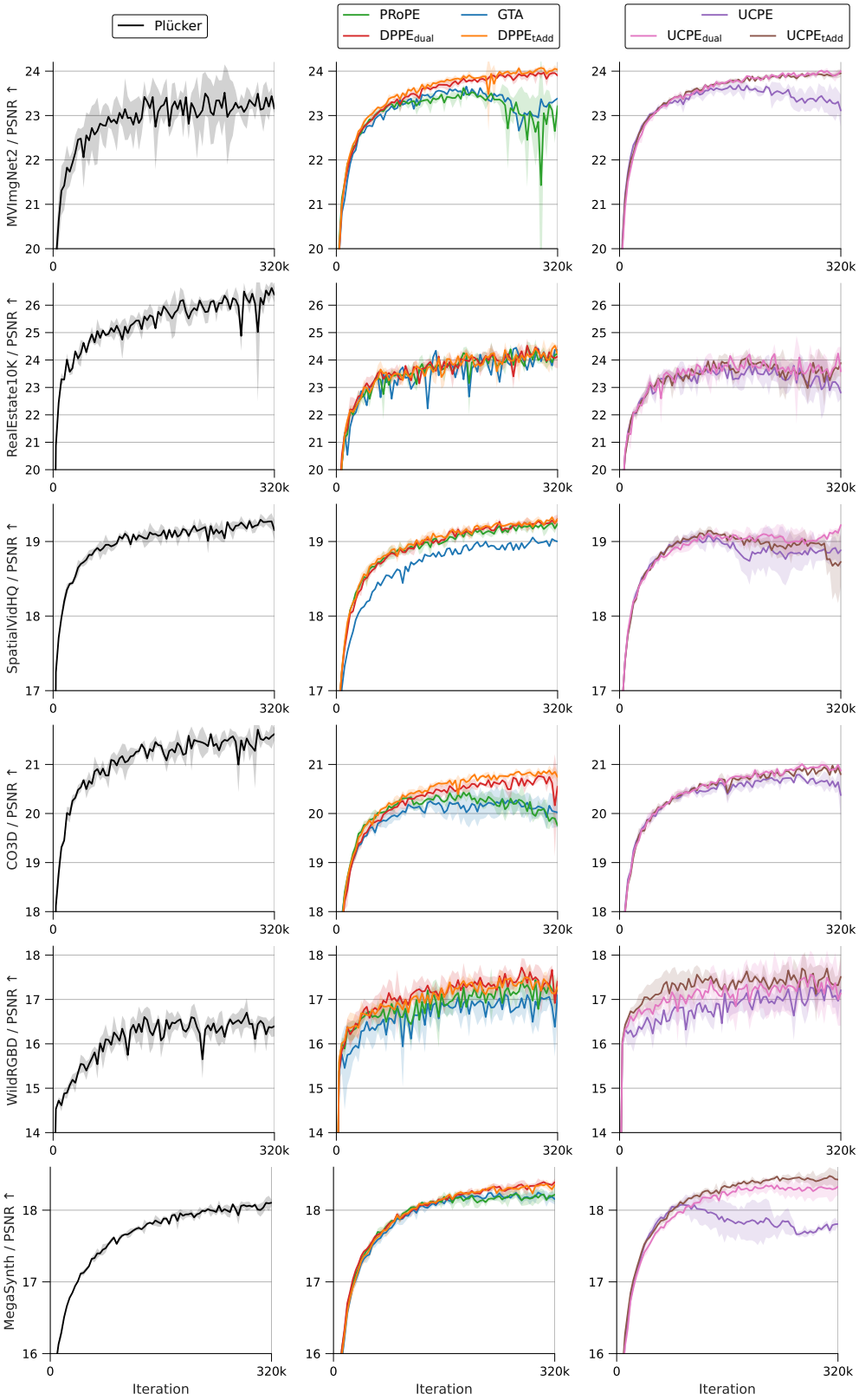


Figure 5: **Extended NVS performance (PSNR) across six datasets.** Results represent the mean over three seeds, and the shaded regions indicate the standard deviation. The learning curves show that our decoupled variants (DPPE and UCPE's) consistently achieve superior or competitive robustness and training stability compared to coupled baselines.

Table 9: Depth estimation performance measured by L1 loss. The best scores are shown in **bold**.

Method	ARKitScenesWAI	CO3DWAI	MEGASYNTHWAI
RoPE	3.272	0.793	0.110
PRoPE	2.735	0.490	0.071
$\text{DPPE}_{\text{tAdd}}$	2.827	0.499	0.070
$\text{DPPE}_{\text{dual}}$	**2.734**	**0.466**	**0.068**

E.2 Large Scale Multi-View Depth Estimation

To demonstrate that DPPE achieves strong performance regardless of the target task, we apply and compare PRoPE [3] and DPPE within VGGT [4], a multi-view depth estimation model.

Experiment Details We conduct experiments using the VGGT [4] architecture. All model parameters are consistent with the original, and the pretrained DINOv2 [48] encoder remains frozen during training. Training is performed for 240k iterations with a learning rate of 1.0×10^{-4} and a warmup-stable-only scheduler (without decay). The minimum learning rate is set to 1.0×10^{-6}, with a warmup period of 1k iterations and a weight decay of 0.05. The training dataset consists of CO3D [45] (34,998 scenes), MegaSynth [47] (65,200), WildRGBD [46] (23,280), and ARKitScenes [49] (4,498). For validation, we use the official train/val split for ARKitScenes. For CO3D and MegaSynth, as no official splits were available, we randomly sampled 2,000 scenes from each for validation. WildRGBD was excluded from the validation set due to significant missing pixels in its ground-truth depth, although it remains suitable for training.

Modified Depth Loss In the original VGGT, depth is predicted in a space normalized such that the mean depth of all valid pixels equals 1. However, since the model does not have prior information about valid pixel locations, we modified this normalization for consistency. In our setup, we set the reference camera to identity and normalize the mean distance of all camera centers to 1. Depth is then predicted at this scale and evaluated using a direct L1 loss. This approach provides a deterministic scale for both the data and the model. Additionally, we apply the four-stage multi-scale gradient loss used in the original VGGT. Camera loss and point tracking loss are omitted as they are not applicable to our specific experimental setting. Metrics in the plots are smoothed using a decay factor of 0.95 every 1k iterations.

Results and Discussion We compare four positional encoding methods: RoPE, which is used in original VGGT, PRoPE, $\text{DPPE}_{\text{tAdd}}$, and $\text{DPPE}_{\text{dual}}$. While stable evaluation on ARKitScenes is challenging due to its small proportion in the training data, consistent patterns emerge in CO3D and MegaSynth.

Consistent with the results on the similarly object-centric MVImgNet2 [30] dataset shown in Tab. 5, $\text{DPPE}_{\text{dual}}$ demonstrates steady performance improvements on CO3D even in the later stages of training. This confirms the effectiveness of $\text{DPPE}_{\text{dual}}$ not only for NVS tasks but also for multi-view depth estimation. Furthermore, in MegaSynth—a dataset characterized by significant viewpoint changes rather than object-centric scenes—the performance gains of PRoPE saturate during late-stage training, whereas $\text{DPPE}_{\text{dual}}$ maintains consistent improvement. We also observe that $\text{DPPE}_{\text{tAdd}}$ contributes significantly to performance enhancements in the final stages of MegaSynth training. These results indicate that DPPE generalizes well beyond object-centric scenarios, demonstrating robust performance on datasets involving simultaneous changes in rotation and translation.

E.3 Further Scale-Up Training

To examine whether the benefit of DPPE persists in a more demanding training regime, we further scale up both the model size and the training horizon. Specifically, we double the network depth from 12 to 24 layers and extend the training schedule from 320k to 1M iterations. This setting is substantially more demanding than the configurations used in the preceding experiments, and at this scale training stability becomes a non-trivial concern. To establish a reliable foundation for such long training runs, we additionally introduce a cosine decay learning-rate scheduler with a short linear warmup, which we find to be essential for stable convergence over the extended schedule.

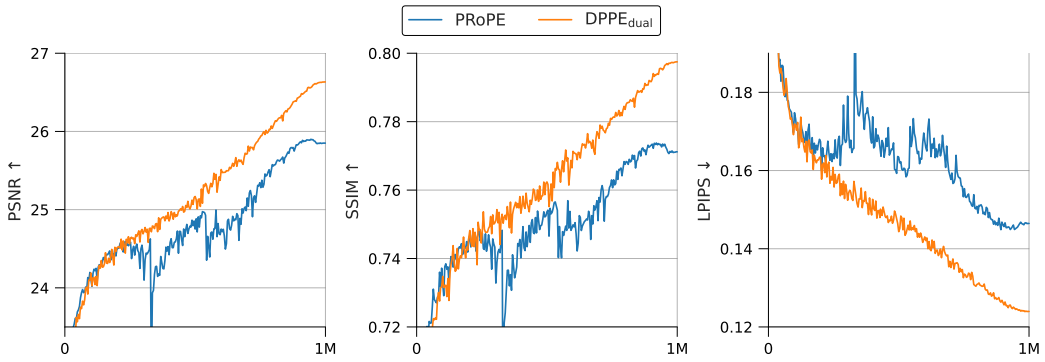


Figure 6: **Futher scaled-up training with PRoPE and DPPE.** DPPE shows stable training behavior throughout the entire schedule while PRoPE suffers from training instability.

We evaluate this scaled-up setting on MVImgNet2 for novel view synthesis (NVS), and compare DPPE against PRoPE [3] under identical conditions. Following our base protocol, the number of input views is fixed to two. The peak learning rate is set to 5e-4 with a linear warmup of 500 steps followed by cosine decay over the remaining iterations. All other hyperparameters are kept identical to those used in Sec. 6.

As reported in Fig. 6, DPPE exhibits stable training behavior throughout the entire schedule, with all evaluation metrics improving consistently up to the final iteration. In contrast, PRoPE suffers from training instability and its performance degrades in the late stage of training. This trend is consistent with the conclusion drawn in the main paper.

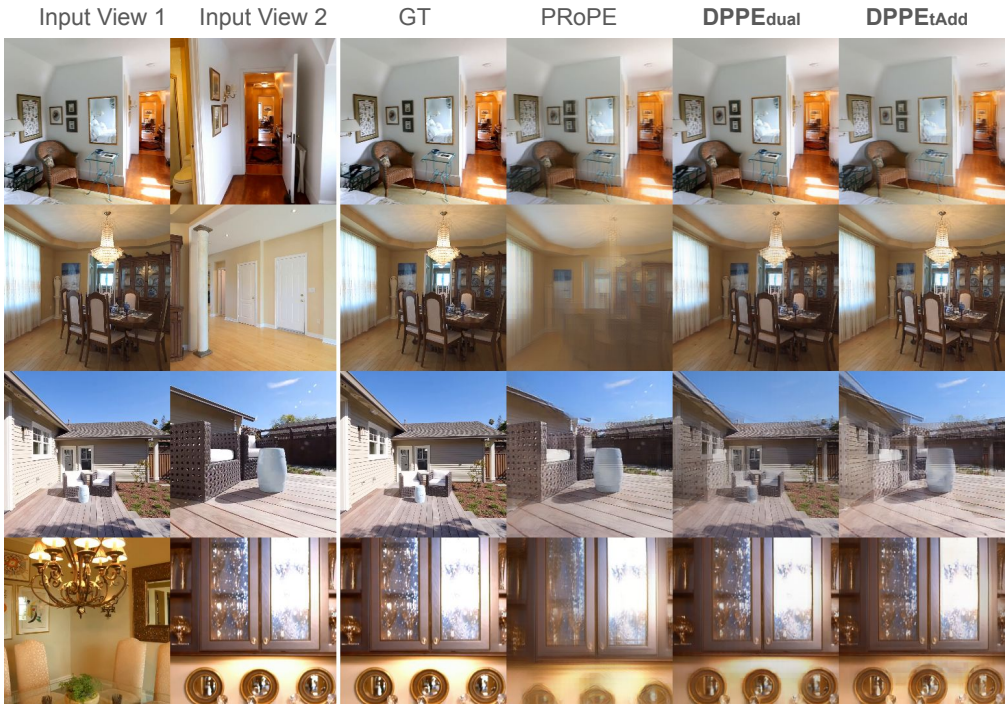


Figure 7: Qualitative comparison of DPPE and PRoPE for novel view synthesis on RealEstate10K.

E.4 Qualitative Comparison on RealEstate10K and SpatialVidHQ

Figs. 7 and 8 show the qualitative comparison of DPPE and PRoPE on RealEstate10K and SpatialVidHQ, respectively.

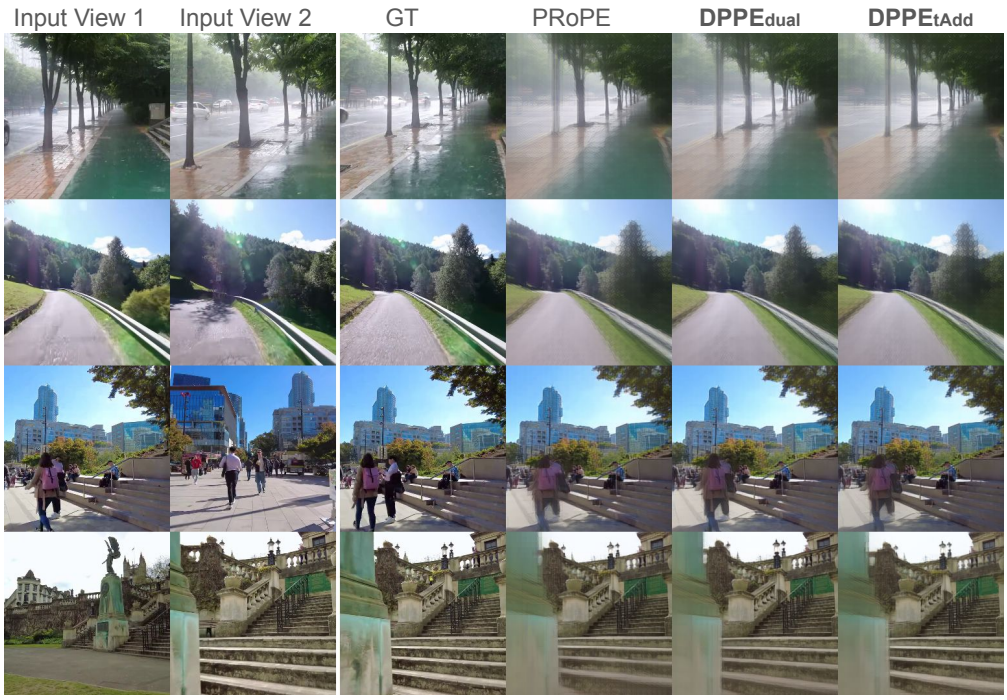


Figure 8: Qualitative comparison of DPPE and PRoPE for novel view synthesis on SpatialVidHQ.

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: The main claims made in the introduction are formally analyzed in Sec. 4, resolved by the method proposed in Sec. 5, and their effectiveness is demonstrated in Sec. 6.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss the limitations of our work in Sec. 7.

   Guidelines:

   - The answer [N/A] 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?

   Answer: [Yes]

   Justification: Theoretical explanations and underlying assumptions are provided in Sec. 5 and App. B.

   Guidelines:

   - The answer [N/A] 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: Key details for reproducing the results are provided in the "Training Setup" paragraphs within each experimental section and the appendix (Secs. 4 and 6 and App. C). In addition, we will make the code publicly available if the paper is accepted.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The code is publicly available at `https://github.com/sbintuitions/dppe`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: Detailed experimental settings and hyperparameters are provided in App. C.

Guidelines:

- The answer [N/A] 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: Due to the large model size and extensive training time, running multiple trials was computationally prohibitive. Instead, as detailed in App. C, we mitigated iteration variance by evaluating accuracy using an Exponential Moving Average (EMA) with a decay factor of 0.75 across checkpoints every 5k steps from 275k to 320k.

Guidelines:

- The answer [N/A] 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).
- 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: The compute resources used for each experiment are detailed in App. C.

Guidelines:

- The answer [N/A] 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: This research fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: This is foundational research in deep learning and computer vision without direct foreseeable risks of malicious use or negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release any models or datasets with a high risk of misuse.

    Guidelines:

    - The answer [N/A] 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: The licenses and terms of use for existing assets are discussed in App. C.3.

    Guidelines:

    - The answer [N/A] 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.
    - 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: [N/A]

    Justification: We do not introduce or release any new datasets or assets in this paper.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: This research does not involve human subjects, making IRB approval inapplicable.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a core component of the methodology; they were used solely for minor text formatting and editing assistance.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.