RASTER-FREE VIEW SYNTHESIS WITH POINT-ANCHORED GEOMETRIC EMBEDDINGS

Anonymous authors
Paper under double-blind review

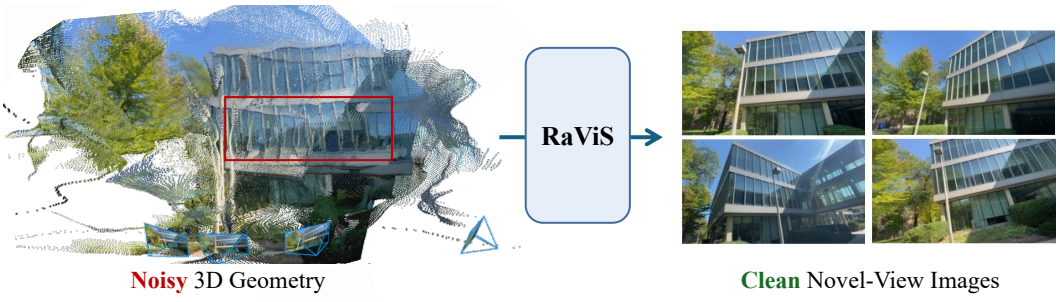

Figure 1: **RaViS** synthesizes novel views from the noisy 3D geometry estimated by a geometry foundation model (GFM). It never rasterizes this geometry into the image and uses it only to guide attention, so geometric errors do not turn into artifacts in the synthesized images.

ABSTRACT

Geometry foundation models (GFMs) such as VGGT estimate camera poses and depth from multi-view images in a single forward pass, and recent feed-forward novel-view synthesis (NVS) methods build on this geometry by rasterizing it into the target view. However, the estimated geometry inevitably contains small errors, and rasterization turns them directly into visible artifacts, such as objects appearing as overlapping copies. We propose **RaViS**, a raster-free view synthesis model built on a simple principle of attending to the geometry rather than rasterizing it. RaViS synthesizes the target view with a transformer and uses the estimated geometry only to tell attention how image patches are related in 3D. To this end, we introduce a point-anchored geometric embedding that places each patch at its own estimated 3D point, so that attention between two patches depends on their relative 3D position and camera rotation. Since the geometry only steers attention and never enters the output image, its errors merely shift where the model attends instead of appearing as artifacts. To further inject 3D structure and semantics, we feed the synthesis transformer with the latent features that the GFM already computes while estimating the geometry. Using only GFM-estimated camera poses, RaViS outperforms prior feed-forward methods on RealEstate10K and DL3DV and in zero-shot generalization. On RealEstate10K, it even outperforms similarly sized baselines that have access to ground-truth camera poses.

1 INTRODUCTION

Novel-view synthesis (NVS) from multi-view images is a long-standing problem in computer vision, with applications spanning virtual reality (Xu et al., 2023), autonomous driving (Yan et al., 2024), and robotics (Zheng et al., 2024). NeRF (Mildenhall et al., 2020) and 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) have become central to this area, achieving exceptional synthesis quality through differentiable rendering. In practice, however, these methods require lengthy structure-from-motion with COLMAP (Schönberger & Frahm, 2016) and per-scene optimization. Enabling low-latency and high-quality NVS remains an important problem for real-world applications.

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

Recently, geometry foundation models (GFMs) such as VGGT (Wang et al., 2025; 2026b) have drawn considerable attention by predicting geometry within seconds in a single forward pass. Instead of relying on 3D constraints such as triangulation, they predict camera poses and dense depth maps purely from data, and these together yield the 3D geometry of the scene. While their camera poses are sometimes less precise than those from constraint-based pipelines, scaling up training data and model size has made GFM geometry accurate enough to build on.

Following this success, many recent methods perform NVS on top of GFMs. Some rasterize 3D Gaussians predicted from GFM geometry (Jiang et al., 2025b; Lin et al., 2026; Liu et al., 2026), while others refine the rasterized point-cloud image with the input images (Wu et al., 2026b). Free of COLMAP and per-scene optimization, they bring low-latency NVS within reach. Still, both lines depend on rasterization and thus inherit the imperfections of the predicted geometry. Because the estimated camera poses and depth are not perfectly accurate, the 3D points unprojected from them are frequently misaligned across views. Rasterizing such geometry then turns these misalignments into visible artifacts, such as objects appearing as overlapping copies. This raises our central research question: *how can we synthesize realistic images from estimated geometry that contains errors?*

In this paper, we propose **RaViS** (Fig. 1), a novel framework built on a simple principle: *attend to the geometry, do not rasterize it*. Rather than rasterizing the estimated geometry, where even tiny errors surface as artifacts, RaViS synthesizes the target view implicitly with a transformer, using the geometry to guide how attention combines information across views based on the 3D relationships between patches. This design lets the model learn from data how far to trust the geometry, making RaViS robust to geometric errors.

To synthesize a target view from multiple input images, the model needs to relate image patches across viewpoints. To capture these cross-view relationships, we introduce the Point-Anchored Geometric Embedding (PAGE), which uses the estimated geometry to encode where patches lie relative to one another in 3D and the camera orientations from which they are observed. These relationships guide attention when combining visual information across views. Unlike prior embeddings that encode only camera poses, PAGE also anchors each patch at its estimated 3D point, exposing the scene structure itself. The model learns to combine this geometric guidance with visual features to synthesize the target view.

We validate RaViS on RealEstate10K (Zhou et al., 2018) and DL3DV (Ling et al., 2024). With GFM-estimated poses for all methods, it outperforms the strongest baseline by at least 1.4 dB PSNR on RealEstate10K and by up to 1.8 dB on DL3DV with 2 to 16 context views, and it transfers best to three unseen datasets (Barron et al., 2022; Wang et al., 2023; Meuleman et al., 2023) without any fine-tuning. Our ablations show that anchoring the embedding at the 3D point of each patch, rather than at the camera, gives a clear PSNR gain (Sec. 4.3).

In summary, our contributions are as follows:

- We propose RaViS, a novel NVS framework that attends to GFM geometry instead of rasterizing it, so that geometric errors are not imprinted on the output as artifacts.
- We introduce PAGE, a point-anchored embedding that conditions every attention layer on the relative 3D position and camera rotation between patches.
- RaViS achieves state-of-the-art performance using GFM-estimated camera poses and even outperforms similarly sized baselines that have access to ground-truth poses.

2 Related Work

View Synthesis with 3D Primitives. NeRF (Mildenhall et al., 2020) and 3DGS (Kerbl et al., 2023) reach photorealistic quality by optimizing a scene representation, which requires densely captured views, camera poses from structure-from-motion (Schönberger & Frahm, 2016), and a lengthy per-scene optimization. Feed-forward methods amortize this cost by regressing the representation in a single forward pass (Yu et al., 2021; Hong et al., 2024; Zhang et al., 2024; Charatan et al., 2024; Chen et al., 2024), but they still assume accurate camera poses. Geometry foundation models (GFMs) such as DUSt3R (Wang et al., 2024) and VGGT (Wang et al., 2025; 2026b) remove this assumption by jointly predicting poses and dense geometry from uncalibrated images, and recent methods predict 3D Gaussians with a head attached to such a backbone (Smart et al., 2024; Ye et al., 2025; Jiang

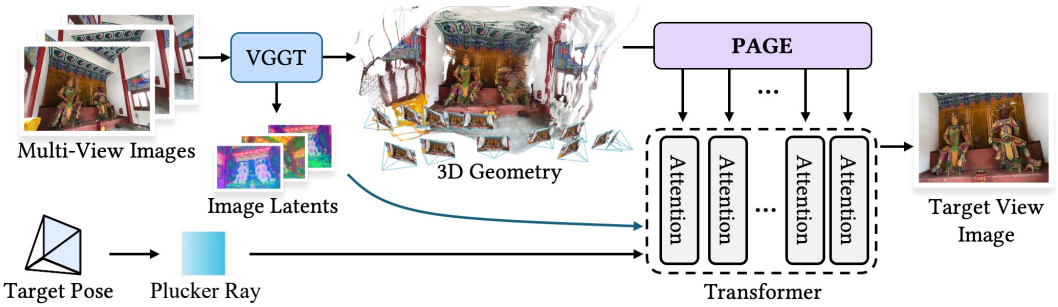

Figure 2: **Overview of RaViS.** A frozen GFM turns the context images into scene geometry and image latents. The latents and the Plücker rays of the target pose form the tokens of a decoder-only transformer, while the 3D points are never rasterized into the image and enter only through PAGE, which conditions every attention layer on the relative 3D position and rotation between patches.

et al., 2025b; Liu et al., 2026; Lin et al., 2026). All of them render an explicit 3D representation, so even small errors in the predicted poses and geometry translate directly into rendering artifacts. RaViS builds on the same GFM geometry but never rasterizes it into image content.

Implicit View Synthesis. Another line of work synthesizes the target view directly with a transformer, without any explicit 3D representation (Suhail et al., 2022; Sajjadi et al., 2022). LVSM (Jin et al., 2025) shows that a plain transformer with minimal 3D inductive bias, conditioned only on Plücker ray embeddings, scales well and outperforms Gaussian-based feed-forward methods. Follow-up works relax its pose supervision (Jiang et al., 2025a; Wang et al., 2026a) or reuse pre-trained 3D features in latent form (Szymanowicz et al., 2026). Closest to our work, PVSM (Wu et al., 2026b) also exploits GFM geometry, rasterizing the unprojected context point cloud into the target camera and feeding the resulting point-cloud image to the synthesis model. Since the geometry still reaches the model in rendered form, misaligned or missing points appear directly in its input, and the 3D structure is reduced to a single 2D projection. RaViS instead never uses the geometry as image content and exposes the 3D geometry of every patch to every attention layer.

Camera Positional Embedding. In multi-view transformers, the "position" of a token is determined by its camera. The common practice of concatenating Plücker ray maps to the input (Jin et al., 2025) is an absolute embedding and thus depends on an arbitrarily chosen world frame. Relative alternatives act inside attention in the spirit of RoPE (Su et al., 2024), injecting the relative $SE(3)$ camera pose (Safin et al., 2023; Kong et al., 2024; Miyato et al., 2024) or, in PRoPE (Li et al., 2025), the relative projective transform including intrinsics. Later works refine how cameras and rays are encoded (Kenney & Suzuki, 2026; Xie et al., 2026; Zhang et al., 2026). These embeddings depend only on the cameras and pixel rays, not on the depth at which each patch actually lies, so they cannot express the 3D geometry of the scene. RayRoPE (Wu et al., 2026a) does place each patch at a point along its ray, but it predicts this depth in every attention layer wherever it is unknown, such as for target views, and re-encodes keys and values for each query camera. PAGE instead anchors each token at the 3D point of its patch, using a parameter-free per-token matrix that is computed once from the GFM geometry and plugs into standard attention like PRoPE.

3 Method

Given multi-view context images and a target camera, RaViS synthesizes the target view (Fig. 2). A frozen GFM first estimates the scene geometry from the context images and extracts their visual features. Every context and target image is then divided into patches, each of which becomes one token of our synthesis transformer. Context tokens carry the GFM features and the Plücker ray embeddings of their camera, whereas target tokens carry only the ray embeddings of the target camera. The transformer combines information across all tokens by self-attention and decodes the output target tokens into RGB patches. The estimated geometry enters only this attention, through PAGE, which encodes the geometric relationship between every pair of tokens, so the geometry is never rasterized into the image. We first describe geometry inference (Sec. 3.1), then introduce PAGE (Sec. 3.2), and finally detail the synthesis network and its training (Sec. 3.3).

3.1 GEOMETRY INFERENCE

We adopt VGGT-Ω (Wang et al., 2026b) as our geometry foundation model. Given V multi-view images $\{I_n \in \mathbb{R}^{3\times H\times W}\}_{n=1}^{V}$, the model predicts, in a single feed-forward pass, a per-view depth map $D_n \in \mathbb{R}^{H\times W}$, camera intrinsics K_n, and an extrinsic $T_n \in SE(3)$:

$$\{D_n, K_n, T_n\}_{n=1}^{V} = \mathcal{F}_{\text{geo}}(\{I_n\}_{n=1}^{V}), \tag{1}$$

where $T_n = \left[\begin{smallmatrix} R_n & \boldsymbol{t}_n \\ \mathbf{0} & 1 \end{smallmatrix}\right]$ stacks the world-to-camera rotation $R_n \in SO(3)$ and translation $\boldsymbol{t}_n \in \mathbb{R}^3$. Following VGGT (Wang et al., 2025), the model expresses all of its predictions in the coordinate frame of the first image I_1, so that $R_1 = \mathbf{I}$ and $\boldsymbol{t}_1 = \mathbf{0}$. Since which image comes first is arbitrary, the global frame in which the geometry lives is arbitrary as well.

Together, depth and camera parameters lift every pixel into the global frame. A pixel $\boldsymbol{u}$ of view n with homogeneous coordinate $\bar{\boldsymbol{u}}$ is unprojected to a 3D point

$$\boldsymbol{p}_n(\boldsymbol{u}) = R_n^{\top}\left(D_n(\boldsymbol{u})\, K_n^{-1}\bar{\boldsymbol{u}} - \boldsymbol{t}_n\right), \tag{2}$$

yielding a dense point map $\boldsymbol{p}_n$ per view in the global frame.

Because a GFM predicts this geometry without explicit 3D constraints, its poses and depth may contain small errors, which a rasterization pipeline would carry directly into the synthesized image as artifacts. We therefore use the geometry *implicitly*, as a guide for attention.

A naive alternative is to feed the estimated 3D points to the transformer as absolute coordinates. However, these coordinates depend on the arbitrary global frame, so a mere rotation or translation of the same scene changes every input, and thus the output. This motivates a *relative* representation of geometry that is invariant to such global transformations.

3.2 POINT-ANCHORED GEOMETRIC EMBEDDING (PAGE)

Prior relative positional embeddings encode the relative *camera* pose between tokens (Kong et al., 2024; Miyato et al., 2024; Li et al., 2025). PAGE extends them so that each token carries the estimated *3D point* of its patch, not just the pose of its camera.

Relative attention. Throughout this section, i and j index *tokens* (image patches), in contrast to the view index n used in Sec. 3.1. Token i lies in view $n(i)$ and inherits its camera, so we abbreviate that view's parameters by the token index, writing $R_i := R_{n(i)}$, $\boldsymbol{t}_i := \boldsymbol{t}_{n(i)}$, $K_i := K_{n(i)}$, and $D_i := D_{n(i)}$. Following recent relative embedding schemes (Miyato et al., 2024; Li et al., 2025), we attach to each token a per-token embedding matrix E_i that carries its geometry. We transform the query q_i of every token by $E_i^{\top}$, its key k_i and value v_i by E_i^{-1}, and the attention output o_i by E_i, so that two tokens interact only through their relative geometry:

$$o_i = E_i\,\text{Attention}\big(E_i^{\top} q_i,\ \{E_j^{-1}k_j\},\ \{E_j^{-1}v_j\}\big) \tag{3}$$

$$= E_i \sum_j \underset{j}{\text{softmax}}\big((E_i^{\top} q_i)^{\top} E_j^{-1}k_j\big)\, E_j^{-1}v_j \tag{4}$$

$$= \sum_j \underset{j}{\text{softmax}}\big(q_i^{\top} \underbrace{E_i E_j^{-1}}_{\text{relative}} k_j\big)\, \underbrace{E_i E_j^{-1}}_{\text{relative}}\, v_j. \tag{5}$$

Tokens i and j thus interact only through the *relative* term $E_iE_j^{-1}$, which stays unchanged when a common transformation is applied to the embeddings of all tokens. Up to this point our formulation coincides with prior work. The difference is that previous methods instantiate E_i with a *camera*-only embedding, whereas we encode *both camera and geometry*.

Geometry component. We define E_i as a combination of a per-patch geometry component G_i and a 2D rotary embedding. G_i is a per-patch analogue of a camera's world-to-camera projection. It defines a local coordinate frame that is *anchored at the patch's own 3D point* $\boldsymbol{p}_i$ and oriented by the camera rotation R_i, and additionally carries the intrinsics K_i. Replacing the usual camera center with the patch point $\boldsymbol{p}_i$ is what lets the relative term $G_iG_j^{-1}$ capture not only the camera pose but also the geometry. Formally,

$$G_i = \tilde{K}_i \begin{bmatrix} R_i & -R_i\,\boldsymbol{p}_i \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{6}$$

where the *anchor* $\boldsymbol{p}_i \in \mathbb{R}^3$ is the patch center $\boldsymbol{u}_i$ unprojected by Eq. 2 with the mean depth $\bar{D}_i = \frac{1}{|\mathcal{P}_i|}\sum_{\boldsymbol{u}\in\mathcal{P}_i} D_i(\boldsymbol{u})$ over the pixels $\mathcal{P}_i$ of patch i that carry a valid depth, $\boldsymbol{p}_i = R_i^\top\left(\bar{D}_i\, K_i^{-1}\bar{\boldsymbol{u}}_i - \boldsymbol{t}_i\right)$. The block $\left[\begin{smallmatrix} R_i & -R_i\boldsymbol{p}_i \\ \mathbf{0} & 1 \end{smallmatrix}\right]$ maps a world point $\boldsymbol{X}$ to $R_i(\boldsymbol{X} - \boldsymbol{p}_i)$, moving the origin to $\boldsymbol{p}_i$ and rotating by R_i, while $\tilde{K}_i$ lifts K_i to an invertible 4×4 matrix, $\tilde{K} = \left[\begin{smallmatrix} K & \mathbf{0} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$.

The relative information between two patches i and j is given by $G_i G_j^{-1}$, which expands as

$$G_i G_j^{-1} = \tilde{K}_i \begin{bmatrix} R_i & -R_i\,\boldsymbol{p}_i \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} R_j^\top & \boldsymbol{p}_j \\ \mathbf{0} & 1 \end{bmatrix} \tilde{K}_j^{-1} \tag{7}$$

$$= \tilde{K}_i \begin{bmatrix} R_i R_j^\top & R_i(\boldsymbol{p}_j - \boldsymbol{p}_i) \\ \mathbf{0} & 1 \end{bmatrix} \tilde{K}_j^{-1}. \tag{8}$$

As illustrated in Fig. 3, the relative term consists of two blocks. The upper-left block $R_i R_j^\top$ is the *relative rotation* between the two views' cameras, while the upper-right block $R_i(\boldsymbol{p}_j - \boldsymbol{p}_i)$ is the *relative position* of patch j's point with respect to patch i's.

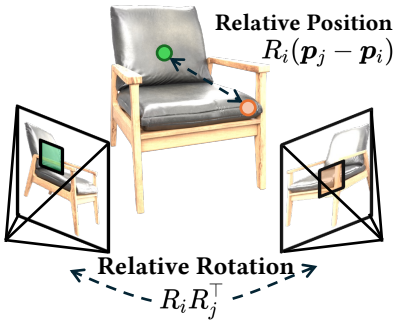

Figure 3: **Illustration of PAGE.** Each patch is anchored at its own 3D point, so two patches interact through the relative rotation of their cameras and the relative position of their points.

$SE(3)$-invariance. The global frame of the GFM geometry is arbitrary. Choosing a different reference view moves all cameras and points by a common rigid transform $g \in SE(3)$ that maps a world point as $\boldsymbol{X} \mapsto Q\boldsymbol{X} + \boldsymbol{s}$, with rotation $Q \in SO(3)$ and translation $\boldsymbol{s} \in \mathbb{R}^3$. Under this change, $R_i \mapsto R_i Q^\top$ and $\boldsymbol{p}_i \mapsto Q\boldsymbol{p}_i + \boldsymbol{s}$, so every component becomes $G_i\, g^{-1}$, and the common factor cancels in the relative term, $(G_i g^{-1})(G_j g^{-1})^{-1} = G_i G_j^{-1}$. Intuitively, the difference $\boldsymbol{p}_j - \boldsymbol{p}_i$ rotates with the global frame while R_i counter-rotates by the same amount, so $R_i(\boldsymbol{p}_j - \boldsymbol{p}_i)$, the position of patch j *as seen from* patch i, does not depend on how the coordinates are defined. We additionally normalize the scene by its median patch depth, which removes the global scale ambiguity of the GFM geometry as well. PAGE therefore retains the exact global-frame invariance of camera-level embeddings and gives every pair of patches its own relative geometry, whereas camera-level embeddings use the same camera term for all patch pairs of two views.

Embedding construction. Since G_i is only 4×4, we follow PRoPE (Li et al., 2025) and build E_i block-diagonally for each attention head of dimension d, devoting half of its channels to copies of G_i and the other half to a 2D rotary embedding Φ_i of the patch's image coordinates:

$$E_i = \underbrace{G_i \oplus \cdots \oplus G_i}_{\text{geometry}} \;\oplus\; \underbrace{\Phi_i \oplus \cdots \oplus \Phi_i}_{\text{2D patch position}} \in \mathbb{R}^{d\times d}, \tag{9}$$

where $\oplus$ is the direct sum, $A \oplus B = \left[\begin{smallmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{smallmatrix}\right]$. Φ_i applies $SO(2)$ rotations whose angles are set by the pixel coordinates (x_i, y_i) of the patch center, so its relative term $\Phi_i\Phi_j^{-1}$ depends only on the 2D offset between two patches in the image plane. Because $\tilde{K}_i$ uses intrinsics normalized by the image size, context and target tokens on different patch grids are embedded consistently.

Anchors of target tokens. PAGE needs a 3D point for every token, but the depth of the target view is unknown. We therefore project the context point maps (Eq. 2) into the target camera to obtain an approximate target depth, from which the target anchors are computed in the same way as for context patches. Patches that receive no point, e.g., in disoccluded regions, fall back to the camera center. This projected depth only places anchors inside attention and never becomes image content, so its holes and errors act as soft cues rather than being imprinted on the output.

3.3 Synthesis Network and Training

Latent Input. Instead of raw RGB, which carries no explicit 3D cue, we feed the synthesis transformer with latent features that the frozen GFM already computes while estimating the geometry. These features carry geometric and semantic information and require no separate image encoder. For every context patch, a trainable DPT decoder (Ranftl et al., 2021) fuses the multi-scale tokens

that feed the GFM depth head into the *geometry feature* f_{geo}, and the *appearance feature* f_{rgb} comes from an early layer of the GFM image encoder, which still preserves local color and texture. The two features are concatenated, linearly projected, and summed with the Plücker ray embedding of the context camera to form the input token f_{input} of each context patch.

Target View Synthesis The target view is specified by its per-pixel Plücker rays $\boldsymbol{r}$, which are patchified into target tokens that carry no appearance. Following LVSM (Jin et al., 2025), a decoder-only transformer $\mathcal{T}$ processes the target tokens together with the tokens of all context views by full self-attention, with PAGE applied at every layer, and the output target tokens are linearly decoded into the image:

$$\hat{I} = \mathcal{T}(f_{\text{input}},\, \boldsymbol{r}\,;\, \mathcal{G})\,, \tag{10}$$

where $\mathcal{G}$ denotes the estimated scene geometry (camera parameters and patch anchors). Importantly, the anchors never enter the network as absolute 3D coordinates, only *relatively* through PAGE.

Training Loss With the GFM frozen, we train the transformer $\mathcal{T}$ and the DPT decoder to reconstruct the ground-truth target image I_{gt} with an RGB reconstruction term and a perceptual term,

$$\mathcal{L} = \|\hat{I} - I_{\text{gt}}\|_2^2 \;+\; \lambda\, \mathcal{L}_{\text{perc}}\Big(\hat{I},\, I_{\text{gt}}\Big)\,, \tag{11}$$

where $\mathcal{L}_{\text{perc}}$ is the perceptual loss (Johnson et al., 2016) and $\lambda = 0.5$ balances the two terms.

4 Experiment

We report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) on RealEstate10K (Sec. 4.1), on DL3DV with zero-shot transfer (Sec. 4.2), and in ablations (Sec. 4.3).

4.1 Two-view Input Setting

Dataset and Protocol We train on RealEstate10K (RE10K) (Zhou et al., 2018) and evaluate on the NoPoSplat (Ye et al., 2025) benchmark, whose test scenes provide two context and three target views at 256×256, grouped into small, medium, and large overlap. Our main protocol uses *GFM-estimated camera poses*, the condition under which a GFM-based pipeline is deployed in practice. All poses come from a VGGT-Ω (Wang et al., 2026b) pass over all views, and RaViS takes depth and latents from a second pass over the context views only, so that the target image never influences the context features. For reference, we also report the baselines with the dataset's *GT camera poses*.

Implementation Details The frozen GFM is the 1B-parameter VGGT-Ω, which receives images upsampled to 512×512 (16×16 patches), while targets are synthesized at 256×256 (8×8 patches), giving 32×32 tokens per view. We train the 24-layer, 768-wide synthesis transformer and the DPT decoder (total 205M parameters) for 100K iterations with GFM-estimated poses (Appendix A).

Baselines We compare against two families of feed-forward methods on the same scenes and metrics. *Gaussian-splatting* methods, WorldMirror (Liu et al., 2026) and Depth Anything 3 (Lin et al., 2026), rasterize predicted 3D Gaussians. Being GFMs themselves, they use their own predicted poses, and we run their official checkpoints, trained on other data. *Implicit* methods, LVSM (Jin et al., 2025), PVSM (Wu et al., 2026b), and LagerNVS (Szymanowicz et al., 2026), synthesize the target image with a transformer like ours, and we run their official RE10K checkpoints under both pose protocols, taking the depth for PVSM's point cloud from MapAnything (Keetha et al., 2026). LagerNVS (†) fine-tunes its 1B-parameter GFM encoder, giving it about $5\times$ more trainable parameters than RaViS, so we compare with it at a matched budget in Sec. 4.2.

Results Under GFM-estimated poses, RaViS achieves the best result in every overlap setting and on every metric of Table 1, exceeding every baseline, including the much larger LagerNVS, by at least 1.4 dB PSNR. LVSM and LagerNVS are sensitive to the pose source, losing up to 1.3 and 3.1 dB relative to their GT-pose results, whereas PVSM, whose depth is re-estimated for each pose source, changes by at most 0.3 dB. Although RaViS relies only on the less accurate GFM-estimated poses, it still outperforms every method of comparable size that is *given* GT poses, and only LagerNVS with GT poses and $5\times$ more trainable parameters is higher. The Gaussian-splatting methods stay 7.1 to 13.4 dB behind, which is consistent with rasterization errors limiting them, although they are also trained on different data. Fig. 4 shows the same pattern qualitatively, where the baselines blur

Model	Small Overlap			Medium Overlap			Large Overlap		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GT Camera Pose									
LVSM	24.41	0.815	0.169	27.44	0.869	0.124	30.62	0.916	0.085
PVSM	23.78	0.801	0.176	26.32	0.854	0.130	28.65	0.896	0.094
LagerNVS†	25.83	0.846	0.140	28.94	0.892	0.100	32.27	0.933	0.066
GFM-estimated Camera Pose									
Depth Anything 3	14.83	0.463	0.474	16.16	0.501	0.405	17.43	0.525	0.346
WorldMirror	18.34	0.652	0.297	20.42	0.709	0.241	22.53	0.754	0.196
LVSM	23.83	0.800	0.175	<u>26.69</u>	0.855	0.129	<u>29.34</u>	<u>0.898</u>	0.092
PVSM	23.77	<u>0.802</u>	0.175	26.26	<u>0.856</u>	0.129	28.34	0.892	0.095
LagerNVS†	<u>23.95</u>	0.785	<u>0.161</u>	26.32	0.830	<u>0.121</u>	29.21	0.879	<u>0.084</u>
RaViS (Ours)	**25.40**	**0.839**	**0.153**	**28.09**	**0.883**	**0.114**	**30.79**	**0.922**	**0.079**

Table 1: **Quantitative results on RealEstate10K.** Methods in gray are *given* ground-truth (GT) camera poses, unlike RaViS, and are left out of the ranking. †LagerNVS trains its pretrained GFM encoder along with its new layers, giving it about 5× more trainable parameters than RaViS. **Bold** and <u>underline</u> denote the best and second-best results among methods with GFM-estimated poses.

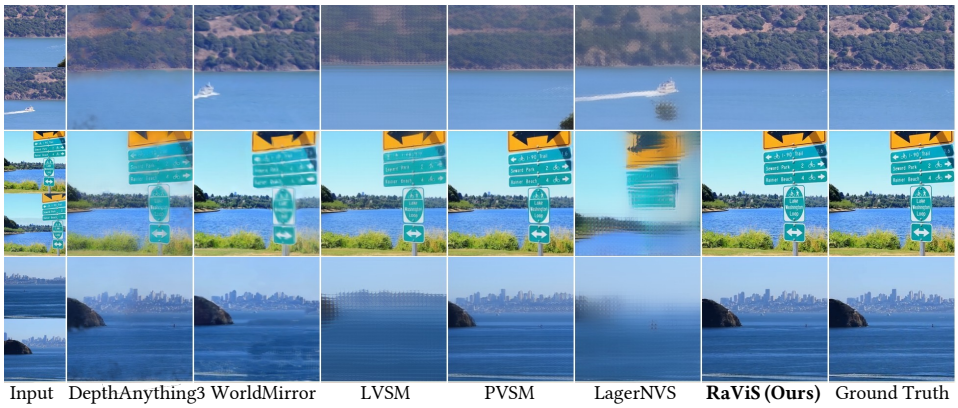

Figure 4: **Qualitative comparison on RealEstate10K under GFM-estimated camera poses.** All methods receive the same two context views (left). Depth Anything 3 and WorldMirror, which rasterize Gaussians, blur and duplicate scene structure where the predicted geometry is misaligned, as on the road signs in the second row. LVSM and LagerNVS wash out the skyline and the rock in the third row, and LagerNVS also the road signs. PVSM keeps the coarse layout but blurs fine textures such as the hillside in the first row. RaViS stays sharp and closest to the ground truth.

or lose structures that RaViS keeps sharp. These results support our premise that geometry which only guides attention, rather than being rasterized, lets the model tolerate geometric errors.

4.2 Multi-view Input Setting

Dataset and Protocol We next study a varying number of context views on DL3DV-10K (Ling et al., 2024), training on about 10K scenes and evaluating on the 140 held-out test scenes of Long-LRM (Ziwen et al., 2025) at a resolution of 256×448. We train from scratch for 50K iterations, drawing the number of context views from $\{2, 4, 8, 16\}$ as in DVSM (Sun et al., 2026), with all other settings as in Sec. 4.1 (Appendix A). At test time, every 8th frame is held out as a target and $N \in \{2, 4, 16\}$ context views are chosen from the rest by K-means clustering of the camera centers.

Baselines Since the public checkpoints of LVSM, PVSM, and LagerNVS differ in data, resolution, number of views, and compute, we re-train them in our codebase under a setting *identical* to ours (Table 2) with a comparable number of trainable parameters. All methods receive the same GFM-estimated poses and take any geometry from the same frozen VGGT-Ω. LagerNVS (†) freezes this encoder instead of fine-tuning it and trains only its later layers. To help it converge, we feed it Plücker rays, RGB patches, and early features of a separate DINOv3 (Siméoni et al., 2025).

Model	2 Context Views			4 Context Views			16 Context Views		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LVSM	14.58	0.382	0.599	16.73	0.457	0.489	<u>20.92</u>	<u>0.625</u>	<u>0.301</u>
PVSM	<u>15.97</u>	<u>0.427</u>	0.552	16.74	0.439	0.517	16.54	0.423	0.543
LagerNVS†	15.95	0.427	<u>0.538</u>	<u>17.52</u>	<u>0.476</u>	<u>0.451</u>	20.13	0.574	0.324
RaViS (Ours)	**16.97**	**0.483**	**0.465**	**19.15**	**0.567**	**0.350**	**22.68**	**0.702**	**0.214**

Table 2: **Novel view synthesis on DL3DV with a varying number of context views.** We re-train all baselines under a setting identical to ours (data, view sampling, resolution, schedule, loss, frozen GFM, and GFM-estimated poses). †LagerNVS keeps its pretrained GFM encoder frozen and trains only its new layers, so its number of trainable parameters is comparable to that of RaViS.

Model	MipNeRF360			Free			Hike		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LVSM	19.18	0.434	0.448	<u>19.40</u>	<u>0.533</u>	0.373	<u>18.24</u>	<u>0.347</u>	0.453
PVSM	19.19	<u>0.434</u>	0.445	18.64	0.502	0.392	17.18	0.277	0.539
LagerNVS†	<u>19.48</u>	0.426	<u>0.423</u>	19.17	0.507	<u>0.366</u>	18.01	0.310	<u>0.450</u>
RaViS (Ours)	**20.87**	**0.522**	**0.327**	**21.05**	**0.616**	**0.283**	**19.20**	**0.423**	**0.384**

Table 3: **Zero-shot generalization with 4 context views.** We evaluate the models of Table 2, trained on DL3DV only, on unseen datasets with GFM-estimated poses and without any fine-tuning. †As in Table 2, LagerNVS keeps its GFM encoder frozen and trains only its later layers. **Bold** and <u>underline</u> denote the best and second-best results.

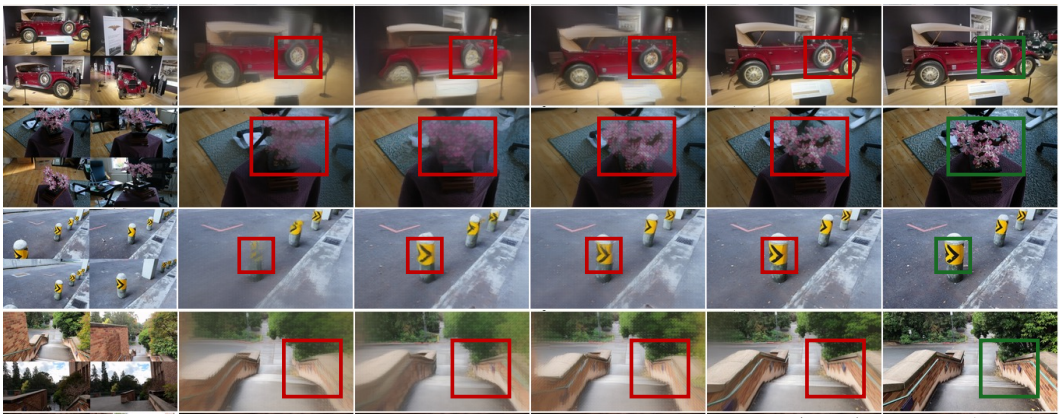

Figure 5: **Qualitative results with 4 context views** on DL3DV (first row) and zero-shot on MipNeRF360, Free, and Hike (other rows). The baselines blur, distort, or wash out fine structures such as the wheel spokes, the blossoms, the road bollard, and the stairs, while RaViS recovers them.

Results on DL3DV As shown in Table 2, RaViS outperforms the re-trained baselines on every metric for every number of context views. With 4 context views, RaViS improves over the best baseline by 1.6 dB PSNR and reduces LPIPS from 0.451 to 0.350. The margin grows to 1.8 dB with 16 views, and RaViS still leads by 1.0 dB with only 2 views, where much of the target is unobserved. With 4 or more views, it beats every baseline in PSNR on all 140 test scenes. PVSM is the one baseline that barely benefits from more context views (16.0, 16.7, and 16.5 dB for 2, 4, and 16 views), although it is the strongest baseline with two views. It is also the only method that consumes the geometry as a rasterized point-cloud image, whose PSNR against the target rises only from 10.9 to 13.7 dB from 2 to 16 views. When the context views are instead sampled as in training, so that every target lies inside the covered range, this PSNR even drops from 16.8 to 13.9 dB, suggesting that small depth disagreements between views accumulate in the rendering. RaViS attends to the same geometry instead of rasterizing it, so these errors remain soft cues, not visible artifacts.

Zero-shot Generalization We evaluate the DL3DV-trained models without fine-tuning on MipNeRF360 (Barron et al., 2022), Free (Wang et al., 2023), and Hike (Meuleman et al., 2023), holding out every 8th frame (every 10th for Hike) as targets, selecting 16 context views as above, and rendering each target from its 4 nearest context views. All images are resized and center-cropped to the

Geometry block G_i	**Rotation**	**Intrinsics**	**Anchor**	PSNR ↑	SSIM ↑	LPIPS ↓
None (2D patch position only)			–	23.01	0.732	0.231
Camera rotation	✓		–	23.65	0.760	0.215
Camera extrinsics (= GTA)	✓		camera	23.73	0.762	0.215
Camera projection (= PRoPE)	✓	✓	camera	<u>23.80</u>	<u>0.765</u>	<u>0.213</u>
PAGE (Ours)	✓	✓	3D point	**24.44**	**0.791**	**0.194**

Table 4: **Ablation on PAGE** on RealEstate10K. Each row adds one factor to G_i, keeping the 2D rotary part Φ_i, the inputs, and the training recipe fixed. Anchor is the origin of G_i's local frame. The camera-anchored rows correspond to GTA (Miyato et al., 2024) and PRoPE (Li et al., 2025).

Context token input	PSNR ↑	SSIM ↑	LPIPS ↓
RGB	22.08	0.713	0.260
f_{rgb}	<u>23.93</u>	<u>0.781</u>	<u>0.212</u>
f_{rgb} + f_{geo} (Ours)	**24.44**	**0.791**	**0.194**

Table 5: **Ablation on the latent input** on RealEstate10K, with PAGE and the training recipe fixed. f_{rgb} and f_{geo} are the appearance and geometry features of Sec. 3.3.

training resolution of 256×448, since PVSM does not transfer to other aspect ratios. As shown in Table 3, RaViS transfers best on all three datasets and on every metric, beating the strongest baseline by 1.39, 1.65, and 0.96 dB PSNR and every baseline on 19 of the 20 test scenes, suggesting that the advantage is not specific to DL3DV. Fig. 5 confirms this qualitatively, as RaViS recovers fine structures, such as the blossoms and the road bollard, that the baselines blur or distort.

4.3 Ablation

We ablate PAGE and the latent input in the two-view setting, training every variant from scratch for 10K iterations on a single GPU, so absolute numbers are lower than in Table 1.

PAGE Table 4 builds the geometry block G_i step by step. With Plücker rays alone, the model reaches 23.01 dB, and the camera rotation, translation, and intrinsics lift this to 23.80 dB, where the last two steps correspond to GTA (Miyato et al., 2024) and PRoPE (Li et al., 2025). Moving the anchor from the camera center to each patch's 3D point adds another 0.64 dB, nearly as much as all camera terms together, and PAGE beats the camera-anchored variant on 85% of the test scenes.

Latent Input Table 5 varies the context-token input with PAGE fixed. Raw RGB patches instead of GFM features lower PSNR by 2.4 dB and are worse on 99% of the test scenes. The appearance feature f_{rgb} alone recovers most of this gap, and the geometry feature f_{geo} adds a further 0.5 dB.

5 Conclusion

We asked how to synthesize realistic images from geometry that contains errors, and answered that such geometry should guide attention rather than be rasterized into pixels. RaViS realizes this principle through PAGE, which anchors the positional embedding of each token at the 3D point of its own patch instead of at its camera, so that any two patches interact through their relative rotation and their relative 3D displacement. Because the geometry is never rasterized into the output image, an error in it only shifts where the model looks instead of leaving a visible artifact. Using only GFM-estimated cameras, RaViS leads on every metric of five benchmarks, with anywhere from 2 to 16 context views and in zero-shot transfer to unseen datasets. In our ablations, anchoring each token at its own 3D point roughly doubles the PSNR gain of camera-level embeddings.

Limitations and future work. RaViS inherits whatever the frozen GFM predicts. It tolerates the small errors that GFMs make in practice, but a grossly wrong pose or depth would still misplace the anchors. Target anchors are also approximate, as they come from projecting the context point maps, and patches without a point fall back to the camera center. Future work could make PAGE aware of how reliable each anchor is, for instance with the depth confidence that GFMs such as VGGT provide, and could refine the target anchors during synthesis instead of fixing them in advance.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, to assist with translation, and to interpret results. We have not used generative AI tools to propose or refine hypotheses, to assist in writing proofs, to formulate mathematical claims, to provide critical ingredients for proving them, to help develop theoretical models or conceptual frameworks, to design or give feedback on the research methodology or experiments, or to support qualitative and thematic data analysis. Generating synthetic data sets and cleaning or reformatting datasets are not applicable to this work.

Additionally, we used generative AI tools to search for information, to identify and summarize relevant literature, to create or edit software code, to edit the paper for readability, and to format references. We have reviewed all AI-assisted work, and the authors verified the code and checked the experimental results produced with generative AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision (ECCV)*, 2024.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *International Conference on Learning Representations (ICLR)*, 2024.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A self-supervised large view synthesis model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2025b.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *International Conference on Learning Representations (ICLR)*, 2025.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*, 2026.

Shun Kenney and Teppei Suzuki. DPPE: Rethinking camera-based positional encoding for scaling multi-view transformers. *arXiv preprint arXiv:2606.31585*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J. Davison. Escher-Net: A generative model for scalable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. WorldMirror: Universal 3D world reconstruction with any-prior prompting. In *International Conference on Machine Learning (ICML)*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

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

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Aleksandr Safin, Daniel Duckworth, and Mehdi S. M. Sajjadi. RePAST: Relative pose attention scene representation transformer. *arXiv preprint arXiv:2304.00947*, 2023.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot Gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

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

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Cheng Sun, Jaesung Choe, Min-Hung Chen, Ryo Hachiuma, and Yu-Chiang Frank Wang. DVSM: Decoder-only view synthesis model done right. *arXiv preprint arXiv:2605.29891*, 2026.

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. LagerNVS: Latent geometry for fully neural real-time novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Haoru Wang, Kai Ye, Minghan Qin, Yangyan Li, Wenzheng Chen, and Baoquan Chen. The less you depend, the more you learn: Synthesizing novel views from sparse, unposed images with minimal 3D knowledge. In *International Conference on Learning Representations (ICLR)*, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-NeRF: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. RayRoPE: Projective ray positional encoding for multi-view attention. In *European Conference on Computer Vision (ECCV)*, 2026a.

Zirui Wu, Zeren Jiang, Martin R. Oswald, and Jie Song. From rays to projections: Better inputs for feed-forward view synthesis. *arXiv preprint arXiv:2601.05116*, 2026b.

Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. URoPE: Universal relative position embedding across geometric spaces. In *European Conference on Computer Vision (ECCV)*, 2026.

Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, Dahua Lin, Michael Zollhöfer, and Christian Richardt. VR-NeRF: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia 2023 Conference Papers*, 2023.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street Gaussians: Modeling dynamic urban scenes with Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3D Gaussian splats from sparse unposed images. In *International Conference on Learning Representations (ICLR)*, 2025.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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 (ECCV)*, 2024.

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 (CVPR)*, 2018.

Shuo Zhang, Xin Su, Wei Wang, Jun Liu, Xinrui Zeng, Yongsen Chen, Chenjie Wang, Guibo Zhu, Jinqiao Wang, Bin Luo, and Liangpei Zhang. G-ray: Ray-level relative geometric position encoding in multi-view vision transformers under camera heterogeneity. *arXiv preprint arXiv:2609.15018*, 2026.

Yuhang Zheng, Xiangyu Chen, Yupeng Zheng, Songen Gu, Runyi Yang, Bu Jin, Pengfei Li, Chengliang Zhong, Zengmao Wang, Lina Liu, Chao Yang, Dawei Wang, Zhen Chen, Xiaoxiao Long, and Meiqing Wang. GaussianGrasper: 3D language Gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 9(9):7827–7834, 2024.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4), 2018.

Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-LRM: Long-sequence large reconstruction model for wide-coverage Gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

A Training Details

This section collects the optimization and view-sampling details of the two settings in Sec. 4.

Two-view setting (RealEstate10K) We train with AdamW (Loshchilov & Hutter, 2019) (β_1=0.9, β_2=0.95, weight decay 0.05) for 100K iterations with a batch size of 256 scenes, a peak learning rate of 4×10^{-4} with 3K warm-up iterations followed by cosine decay, gradient clipping at 1.0, and bfloat16 mixed precision on four NVIDIA B200 GPUs. Following LVSM (Jin et al., 2025), each training sample is a window of 25 to 192 frames whose two end frames serve as context views, with two target views drawn from inside the window.

Multi-view setting (DL3DV) The model is trained from scratch for 50K iterations with a peak learning rate of 2×10^{-4}, and all remaining hyper-parameters follow the two-view setting. Context images are again upsampled by a factor of two before entering the GFM. At every iteration the number of context views N is drawn from $\{2, 4, 8, 16\}$, consecutive context frames are separated by a random stride of 1 to 16 frames, and $\min(N, 4)$ target views are sampled from the frame range covered by the context views. Because our decoder processes each target view together with all context tokens, the cost of a scene grows with N, so we adapt the number of scenes per batch to keep the number of processed views per iteration constant. At test time all context views are fed at once.

B Additional Qualitative Results

Figs. 6, 7, and 8 show zero-shot results of the DL3DV-trained models on MipNeRF360, Free, and Hike under the protocol of Table 3, in which each target is rendered from its 4 nearest context views at 256×448. The first column shows these context views. We show targets on which RaViS improves most over the strongest baseline in LPIPS, excluding targets on which PVSM fails entirely, with at most two widely separated targets per scene (three for Hike). The baselines blur thin and high-frequency structures such as the posts in Free and the foliage in Hike, while RaViS keeps them sharp.

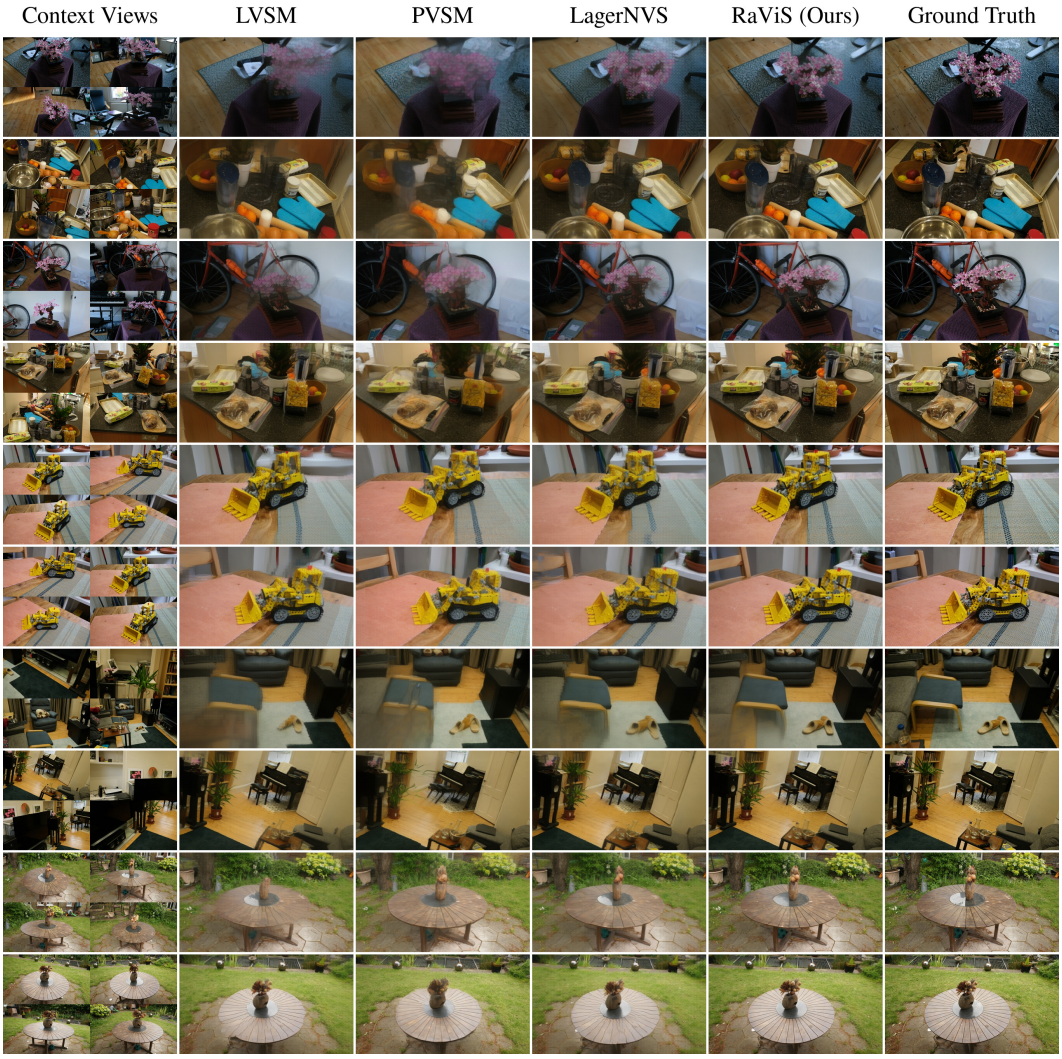

Figure 6: **Zero-shot qualitative results on MipNeRF360** with 4 context views.

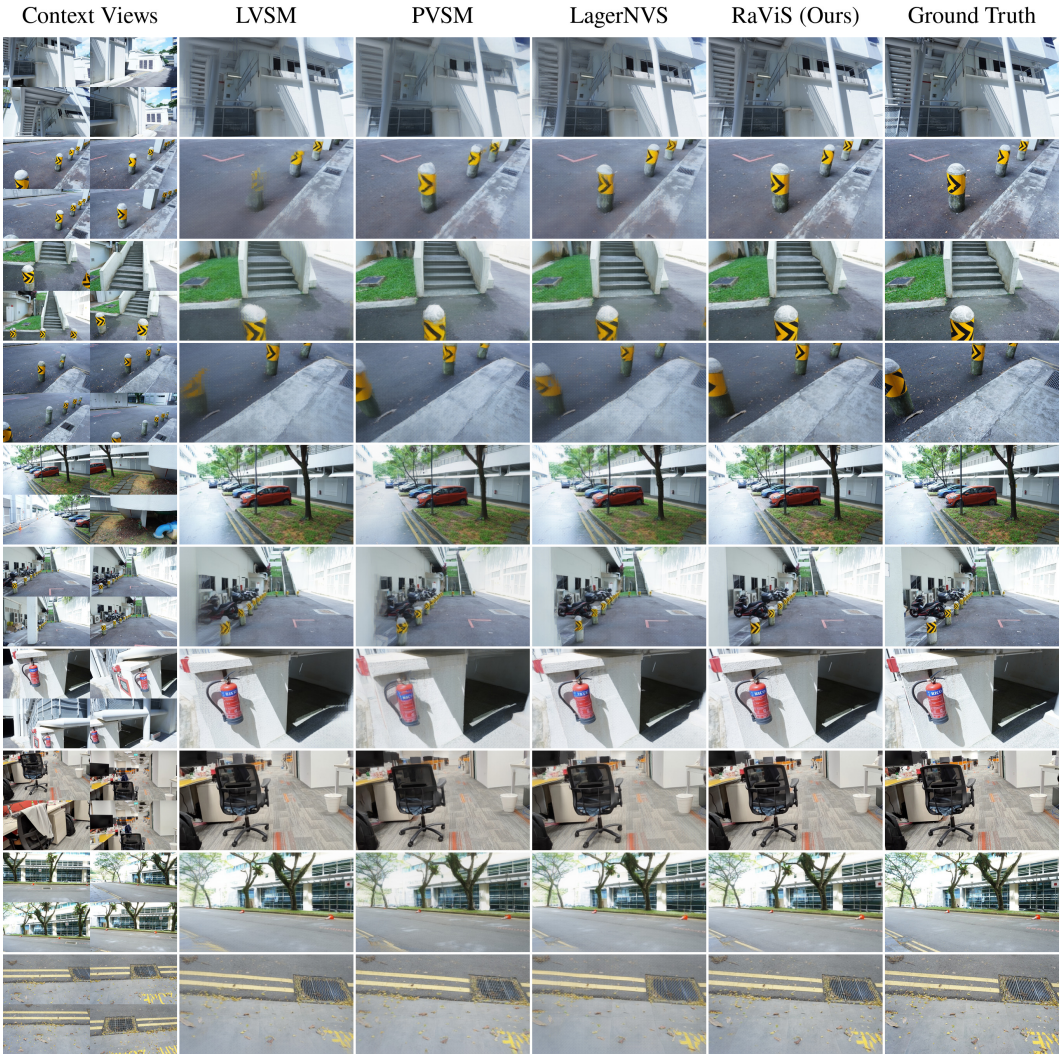

Figure 7: **Zero-shot qualitative results on Free** with 4 context views.

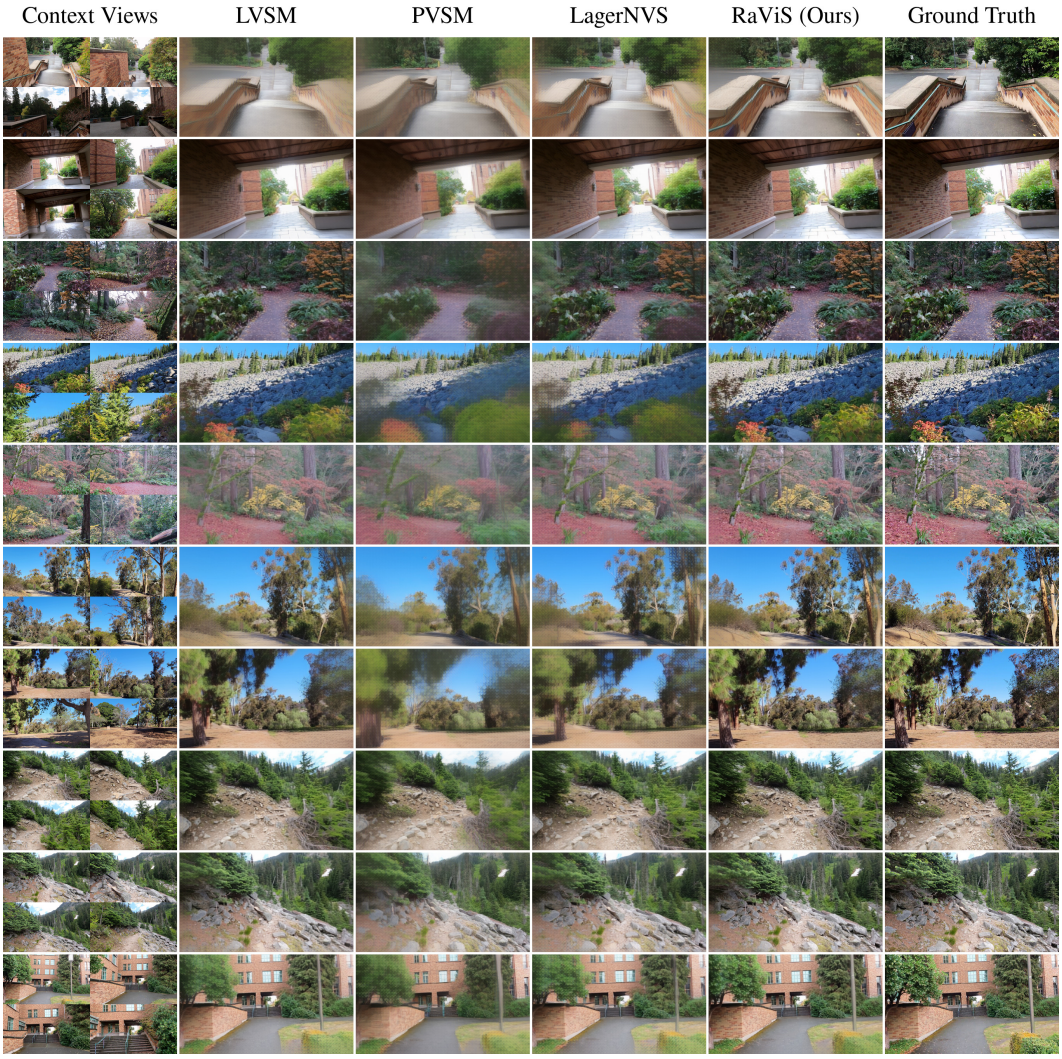

Figure 8: **Zero-shot qualitative results on Hike** with 4 context views.