

NoDrift3R: Raymap-Guided Coupling for Drift-Robust Unposed Feed-Forward 3D Reconstruction

Xiangyu Sun^{1*}, Liu Liu^{2✉}, Seungkwon Yang³, Jingbing Han³, Seungtae Nam³, Zhizhong Su^{2}, and Eunbyung Park^{3✉}

¹ Sungkyunkwan University, South Korea

² Horizon Robotics, China

³ Yonsei University, South Korea

Abstract. Pose-Free Feed-forward 3D Gaussian Splatting (3DGS) has recently emerged as a powerful paradigm for fast scene reconstruction. However, its performance degrades significantly in long image sequences due to cumulative camera pose estimation drift, which propagates errors into geometric modeling and severely limits rendering fidelity. In this work, we revisit the long-sequence bottleneck and identify pose drift as the primary factor restricting reconstruction quality. Furthermore, while SfM-based pseudo ground-truth poses introduce sensor noise, purely rendering-based supervision often leads to optimization instability and local minima due to the entangled optimization of geometry and pose. To address the challenges, we propose a synergistic pose-free framework that explicitly couples geometry and appearance via a Raymap-Guided Coupling Module (RGC). Concretely, we anchor Gaussian centers to raymap-induced geometry and jointly optimize RGB reconstruction, raymap consistency, and camera regularization under a unified objective, yielding a bidirectional feedback loop: stronger geometry improves rendering, and appearance supervision in turn refines geometry and pose. To further stabilize learning across wide temporal ranges, we introduce a Dual-Frequency Viewpoint Scheduling strategy that combines easy-to-hard interval expansion with replay of short-interval pairs. Extensive experiments across in-domain and cross-domain datasets show consistent gains in both rendering and pose estimation, with notably improved robustness on long sequences. Ablation studies validate our central insight: explicitly designed geometry-appearance synergy is the key to scalable and drift-robust pose-free feed-forward 3D reconstruction.

Keywords: Pose Drift · Dual-Frequency schedule · Gaussian Splatting

1 Introduction

Feed-forward 3D Gaussian Splatting (3DGS) has recently emerged as a powerful paradigm for rapid 3D reconstruction and novel view synthesis by predicting

* Intern at Horizon Robotics

✉ Corresponding authors.

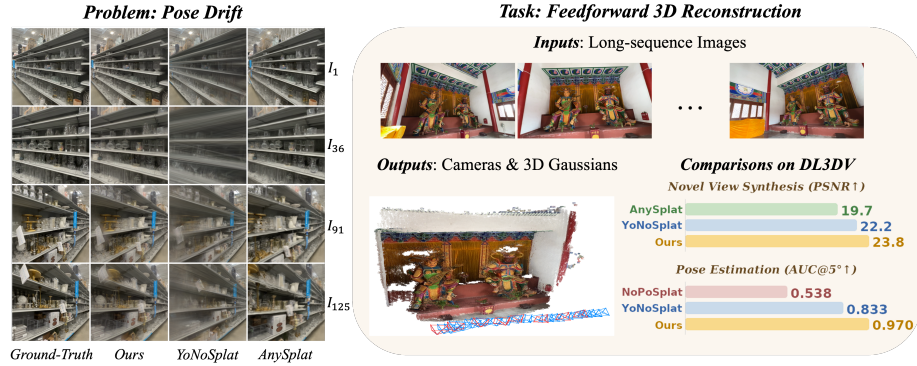


Fig. 1: Overview of our synergistic pose-free framework for feed-forward 3D reconstruction. Our method effectively suppresses the pose drift problem, especially in long-sequence settings. Left: representative failure cases of existing methods under pose drift. Right: our pipeline and outputs (camera poses and 3D Gaussians).

scene representations directly from multi-view images. Compared to conventional per-scene optimization methods [20, 26], feed-forward approaches offer significantly faster reconstruction times, opening the door to applications that require fast 3D reconstruction in various practical scenarios. Early feed-forward pipelines typically assume known camera poses and focus on predicting Gaussian representations conditioned on calibrated multi-view observations [1, 3, 16, 27, 41]. More recently, pose-free feed-forward models have been proposed to jointly infer camera geometry and scene representation directly from unposed image sequences [14, 18, 33, 42, 43], thereby expanding the applicability of feed-forward 3DGS to settings where camera poses are not available in inference.

Despite encouraging progress, pose-free feed-forward 3DGS remains challenging, particularly for long sequences or wide-baseline viewpoints. A central difficulty arises from the strong coupling between predicted camera parameters and scene representation: errors in pose estimation can directly distort the reconstructed representation, while inaccuracies in scene structure can further degrade pose predictions. This feedback loop often leads to accumulated pose drift, where small misalignments gradually amplify over time and deteriorate reconstruction quality. As a result, achieving stable and globally consistent reconstruction from long, unposed image sequences remains an open challenge.

Furthermore, we observe that training instability is a pervasive challenge inherent to all pose-free paradigms, transcending the choice of model initialization. Even when fine-tuning atop powerful 3D foundation models—such as Uni3R [33] with VGGT [36] or YoNoSplat [42] with Pi3 [38]—optimization remains volatile. While YoNoSplat attempts to stabilize this via a teacher-forcing mix-training strategy, and E-RayZer [49] introduces a curriculum learning approach for training from scratch, these methods still struggle to balance stability with fine-grained precision.

Specifically, we identify a critical limitation in E-RayZer’s linear visual-overlap extension: it tends to degrade performance at small-interval samples, which are essential for maintaining local geometric consistency. To bridge this gap, we design a Dual-Frequency Viewpoint Scheduling. By scheduling samples from "easy" (high overlap) to "hard" (low overlap) while systematically re-exposing the model to small-interval cases, our curriculum stabilizes the optimization process across all stages. This approach ensures robust performance across arbitrary frame intervals, effectively suppressing drift and maintaining high-fidelity reconstruction even in long, complex sequences.

In this work, we propose a synergistic joint optimization framework (Fig. 2-(c)) that tightly couples camera geometry and appearance generation through a Raymap-Guided Coupling (RGC) module. Existing approaches generally follow two design paradigms. The first (Fig. 2-(a)) predicts pixel-aligned Gaussian representations directly from images without explicitly leveraging estimated camera poses [14, 33]. While this strategy can produce reasonable renderings, it lacks a mechanism to tightly connect pose estimation with the resulting 3D representation. The second paradigm (Fig. 2-(b)) conditions Gaussian prediction on estimated camera poses, allowing geometry and appearance to interact to some extent [42]. However, this interaction remains limited because camera poses are represented by low-dimensional parameters (e.g., rotation and translation), creating an information bottleneck that cannot convey the dense geometric cues required for accurate Gaussian placement. In contrast, our RGC module establishes a tighter interaction between camera pose estimation and Gaussian generation by predicting dense raymaps that directly determine the spatial distribution of Gaussians. Because raymaps encode per-pixel ray directions, both appearance supervision and geometric constraints jointly influence every per-pixel Gaussian representation during training. This design forms a bidirectional optimization loop in which rendering supervision refines geometry while geometric predictions regularize appearance synthesis. As a result, our framework enables more stable pose-free optimization and improved reconstruction quality across long sequences and wide view intervals.

In sum, our contributions are threefold:

- Identification of the Long-Sequence Bottleneck: We provide a comprehensive analysis of feed-forward 3D Gaussian Splatting and identify that cumulative pose estimation drift is the primary bottleneck limiting rendering quality in long sequences. We further demonstrate that while SfM [31]-based pseudo-GT poses introduce sensor noise, pure rendering-based supervision (e.g., NoPoSplat [43], Uni3R [33]) often leads to optimization instability and local optima due to the simultaneous optimization of geometry and pose.
- Dual-Frequency Viewpoint Scheduling: We propose a novel training schedule driven by visual overlap and a replay mechanism. By systematically re-exposing the model to high-overlap samples while progressively introducing low-overlap challenges, our schedule stabilizes the optimization of coupled geometry and pose. This ensures robust, drift-free reconstruction across arbitrary intervals and wide-range viewpoints.

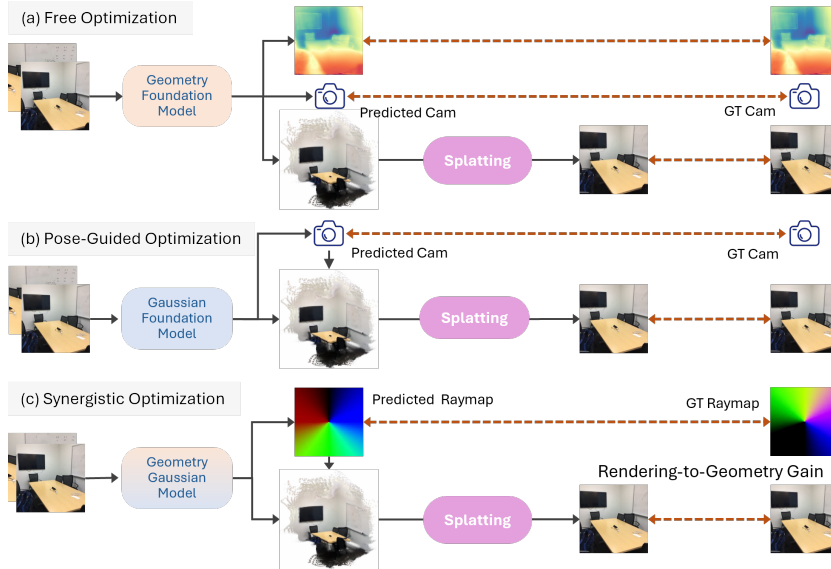


Fig. 2: Our synergistic framework for "Rendering-to-Geometry Gain".

- A Synergistic Framework for "Rendering-to-Geometry Gain": We propose a novel pose-free framework that integrates 3DGS rendering supervision with explicit raymap constraints. Unlike previous methods that optimize geometry and pose in isolation, our approach establishes a positive feedback loop—which we term "Rendering-to-Geometry Gain"—where appearance consistency serves as a powerful geometric prior. This synergy effectively resolves parameter ambiguity and suppresses drift in extended sequences, allowing our model to achieve state-of-the-art (SOTA) performance in both rendering fidelity and camera pose accuracy.

2 Related Work

Pose-dependent Feed-forward 3D Reconstruction. Traditional 3D reconstruction and neural rendering have been significantly advanced by approaches such as NeRF [26] and 3D Gaussian Splatting (3DGS) [20]. However, these methods typically rely on computationally intensive per-scene optimization over dense image sets, which limits their scalability and practicality in real-world applications. To overcome this limitation, feed-forward 3D reconstruction models have been proposed, aiming to predict 3D representations in a single forward pass by learning generalizable priors from large-scale datasets. Early feed-forward approaches primarily focused on multi-view stereo paradigms, leveraging carefully designed architectures that incorporate geometric reasoning mechanisms such as cost volumes and epipolar geometry constraints [1–4, 27, 41]. More recently, architectures based on Vision Transformers [7] that do not rely on such geometric

constraints have emerged as a unified framework for multi-view geometry modeling [11, 13, 15–17, 35, 45, 51]. To address long-sequence reasoning, tttLRM [35] adopts Test-Time Training [34, 48] layers, while Long-LRM [51] incorporates Mamba2 [5] blocks for efficient sequence modeling. Meanwhile, iLRM [16] and MVP [17] carefully design the use of image or viewpoint embeddings to enhance consistency and scalability.

Pose-free Feed-forward 3D Reconstruction. While posed models deliver high fidelity, a fundamental limitation of these approaches is their strong dependence on accurate camera pose, which is usually obtained via Structure from Motion (SfM) methods like COLMAP [31]. In unconstrained real-world scenarios, obtaining precise camera parameters is challenging or infeasible, which substantially restricts their practical applicability. While early methods [8, 9, 22, 25, 40] address this by jointly optimizing poses alongside scene representations, pose-free feed-forward models further bypass per-scene optimization entirely. They directly infer 3D scene representations or camera parameters from sparse inputs in a single pass, operating either on uncalibrated images or with known intrinsics. Recent frameworks bypass traditional SfM pipelines by anchoring 3D primitives directly within a canonical coordinate space [10, 12, 14, 43, 49], by consolidating local per-view geometries into a shared global coordinate frame [6, 18, 42, 47], or by fusing multi-view features into a view-agnostic latent representation for direct global prediction [33]. Despite their efficiency, the inherent coupling between predicted camera parameters and scene geometry frequently leads to accumulated pose drift. In contrast, our model incorporates a raymap loss that allows for the joint optimization of geometry and appearance, effectively suppressing drift accumulation across extended trajectories.

Multi-view Visual Geometry Foundation Models. SfM [31] and Multi-View Stereo remain widely used to obtain reliable visual geometry, but their multi-stage pipelines are notoriously tedious and computationally cumbersome. Recently, the landscape of multi-view visual geometry has been reshaped by large-scale foundation models. DUS3R [37] and MAST3R [21] reformulate the 3D reconstruction task as a pairwise dense pointmap regression problem, bypassing the need for prior camera calibration. VGGT [36] employs an alternating frame-wise and global self-attention mechanism to jointly predict camera parameters, keypoint tracking, depth, and pointmaps from multiple views in a single pass, avoiding the pairwise limitations. π^3 [38] improves VGGT by eliminating the inductive bias of anchoring scenes to a fixed reference frame by leveraging a fully permutation-equivariant architecture to predict affine-invariant outputs. Concurrently, frameworks introduce flexible architectures capable of seamlessly ingesting available geometric priors to enhance global alignment; for instance, MapAnything [19] can optionally condition on known camera intrinsics, poses, and sparse depth, while Depth Anything v3 [23] similarly supports the integration of camera intrinsics and poses. Because these models bypass traditional optimization and are considered to produce high-quality 3D features, they frequently serve as the foundational backbones or pseudo-label teachers for recent pose-free, feed-forward 3D reconstruction models.

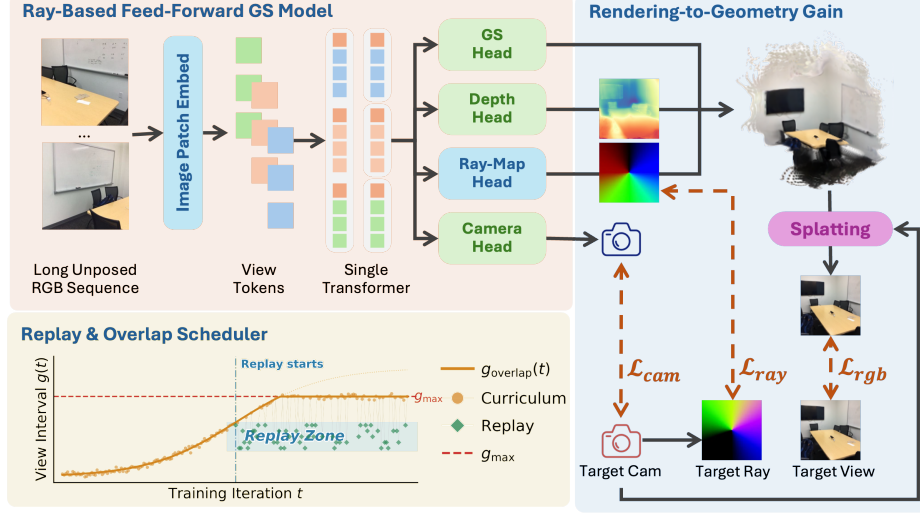


Fig. 3: Overview of the Synergistic Framework. (a) Ray-Based Feed-Forward GS Model. We employ a single transformer (Vallina DINOv2 model), followed by Gaussian Head, Depth Head, Ray-Map Head, and Camera Head. (b) A Raymap-Guided Coupling Module for "Rendering-to-Geometry Gain". (c) Our Replay & Overlap Scheduler leads to robust performance across arbitrary intervals.

3 Method

In this section, we first propose our Raymap-Guided Coupling Module (Sec. 3.1). Then we elaborate on the Synergistic Framework (Sec. 3.2). Subsequently, we propose the Dual-Frequency Viewpoint Scheduling to train the model under both easy- and hard-overlap circumstances (Sec. 3.3).

3.1 Raymap-Guided Coupling Module

Given an input image sequence $\{I_i\}_{i=1}^N$, we first apply a lightweight image patch-embedding to each view and obtain per-view tokens. These view tokens are then processed by a single multi-view transformer to perform inter-view interaction and produce latent tokens for each view. (e.g., Depth Anything v3 [23]).

From these per-view latent tokens, a Dual-DPT [30] head jointly predicts two geometric signals per view: a ray map $R \in \mathbb{R}^{HW \times 6}$ and a depth map $D \in \mathbb{R}^{HW}$. For each pixel, the first three channels of R encode the ray direction (denoted by $r \in \mathbb{R}^{HW \times 3}$) and the last three channels encode the ray origin, $o \in \mathbb{R}^{HW \times 3}$. The depth map provides the scalar distance along each ray. We then lift pixels to 3D and obtain Gaussian centers using both signals:

$$p_j = o_j + D_j \cdot r_j, \quad (1)$$

where $r_j \in \mathbb{R}^3$ and $o_j \in \mathbb{R}^3$ are the direction and origin at the pixel location j , and $p_j \in \mathbb{R}^3$ represents the corresponding Gaussian center position. This

raymap-guided lifting explicitly ties Gaussian positions to predicted geometry and serves as the geometric anchor for subsequent rendering and optimization.

After obtaining D and R , we feed the corresponding latent features (i.e., the per-view latent tokens output by the single transformer) to the Gaussian head to predict Gaussian attributes. Following common 3DGS parameterizations, each Gaussian is defined by its position and a full set of attributes. For each pixel-aligned primitive, our Gaussian head predicts $\{\alpha_j, c_j, s_j, q_j\}$, where $\alpha_j \in [0, 1]$ is opacity, $\mathbf{c}_j \in \mathbb{R}^3$ is color, $\mathbf{s}_j \in \mathbb{R}^3$ is anisotropic scale, and $\mathbf{q}_j \in \mathbb{R}^4$ is a unit rotation quaternion. Importantly, we used the lifted 3D position as the Gaussian center, $\mu_j = p_j$. To keep parameters in valid ranges, we apply

$$\alpha_j = \sigma(f_j^\alpha), \quad s_j = \exp(f_j^s), \quad q_j = \text{normalize}(f_j^q), \quad (2)$$

where f denotes the output from the Gaussian heads for each attribute, $\sigma(\cdot)$ is the sigmoid function. This design yields a complete Gaussian representation aligned with our raymap-derived geometry.

3.2 Learning a Synergistic 3D Representation

A key aspect of our method is the synergistic optimization between appearance and geometry. Unlike prior pipelines (e.g., YoNoSplat [42] and AnySplat [14]) that treat rendering supervision and geometric constraints as loosely coupled signals, our framework ties them together through raymap-derived Gaussian positions. This coupling allows improvements in one branch to directly benefit the other, creating a self-reinforcing training dynamic that is especially important for long sequences and pose-free settings.

Our training objective couples rendering, camera, and raymap supervision in a unified formulation. Given an image sequence $\{I_i\}_{i=1}^N$, we use an RGB reconstruction loss that combines MSE loss and perceptual LPIPS loss:

$$\mathcal{L}_{\text{rgb}} = \frac{1}{N} \sum_{i=1}^n \left(\lambda_{\text{mse}} \|I_i - \hat{I}_i\|_2^2 + \lambda_{\text{lpips}} \text{LPIPS}(I_i, \hat{I}_i) \right), \quad (3)$$

where $\hat{I}_i$ denotes the rendered images. For camera regularization, we supervise the camera parameters predicted by the camera head with Huber loss:

$$\mathcal{L}_{\text{cam}} = \frac{1}{N} \sum_{i=1}^N \text{Huber}(\hat{\theta}_i - \theta_i), \quad (4)$$

where $\hat{\theta}_i, \theta_i$ represent predicted and ground truth camera parameters, respectively. We further apply raymap loss to the per-frame predicted raymaps:

$$\mathcal{L}_{\text{ray}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{abs}}(\hat{R}_i, R_i). \quad (5)$$

The total objective is a weighted sum as follows:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{cam}} \mathcal{L}_{\text{cam}} + \lambda_{\text{ray}} \mathcal{L}_{\text{ray}}. \quad (6)$$

Crucially, these losses are synergistic rather than independent. Our Gaussian positions are derived from depth and raymap; consequently, the RGB loss $\mathcal{L}_{\text{rgb}}$ backpropagates through the rendering process into the raymaps, allowing appearance cues to refine geometry. Meanwhile, the raymap loss $\mathcal{L}_{\text{ray}}$ directly constrains ray directions, and since raymaps define the Gaussian distribution (positions), improved geometry leads to better rendering quality. This mutual reinforcement between raymap supervision and RGB supervision under a joint optimization objective plays an important role in achieving rendering-to-geometry gain, despite the simplicity of the losses.

Insights. This bidirectional coupling from our Raymap-Guided Coupling (RGC) module is the key advantage of our framework: appearance supervision sharpens geometry, and raymap supervision regularizes the spatial distribution of Gaussians, which in turn improves appearance. As a result, we obtain stable optimization and stronger generalization across arbitrary view intervals without introducing complex auxiliary losses. The simplicity of our loss design is therefore a feature, not a limitation—because the synergy is baked into the representation and the training flow itself. More detailed empirical evidence is provided in the ablation studies (Table 6).

3.3 Dual-Frequency Viewpoint Scheduling

We adopt a curriculum-based training strategy that gradually increases the difficulty of view pairs from high overlap to low overlap configurations, a commonly used paradigm in pose-free and self-supervised learning [49]. Unlike existing approaches, we develop two mechanisms tailored to our framework: (i) a bounded overlap-aware curriculum, and (ii) a replay mechanism that repeatedly injects small-interval samples to provide stochastic local geometric regularization.

Sparsity Upper-Bound Clipping. First, we leverage DINOv2 [28] to obtain pairwise overlap scores [49] by calculating cosine similarity as follows:

$$o(i, j) = \cos(\text{DINOv2}(I_i), \text{DINOv2}(I_j)). \quad (7)$$

Then, we calculate the overlap-aware view interval (the number of frames between input view images) at training iteration t , $g_{\text{overlap}}(t)$, is defined as,

$$g_{\text{overlap}}(t) = \min(g_{\text{curr}}(t), g_{\text{max}}), \quad (8)$$

where $g_{\text{curr}}(t)$ denotes the largest view intervals whose overlap score exceeds a target threshold at training iteration t . The target threshold is gradually annealed according to a schedule, decreasing from 1.0 to 0.75. In addition, we clip the maximum interval $g_{\text{max}} = 15$ between input image pairs to avoid overly sparse supervision. This constraint keeps the scheduler within a bounded temporal range while exposing the model to progressively larger view intervals.

Stochastic Replay Regularization. To prevent the performance degradation on small-interval evaluation, we add a replay branch that samples a large-overlap (small-interval) intervals with probability $p_{\text{small}} = 0.5$ during the last half of the training phase. Specifically, we draw $g_{\text{small}} \sim \mathcal{U}\{6, \dots, 10\}$, and select it with probability p_{small} . More formally, we select the view interval at training iteration t , $g(t)$, as follows:

$$g(t) = \begin{cases} g_{\text{small}}, & \text{with prob. } p_{\text{small}}, \\ g_{\text{overlap}}(t), & \text{with prob. } 1 - p_{\text{small}}. \end{cases} \quad (9)$$

This replay curriculum preserves the benefits of an easy-to-hard schedule while ensuring the model repeatedly sees small-interval scenes, leading to robust performance across arbitrary intervals.

4 Experiments

4.1 Experiments Setup

Implementation Details Our framework is implemented in PyTorch [29]. The model employs a standard ViT [7] backbone (vanilla DINOv2 [28]) without structural modifications to the transformer blocks. To evaluate the scalability and effectiveness of our framework, we adopt two variants of the Depth Anything V3 [23] architecture as our backbone:

- **Giant Model:** For primary comparisons with state-of-the-art (SOTA) methods, we utilize the Giant variant. Built upon a backbone with a hidden dimension of 1536 and 40 transformer blocks, this high-capacity model provides precise initial geometry and raymap priors, maximizing reconstruction fidelity.
- **Large Model:** For all ablation studies, we employ the Large variant to balance computational efficiency with rigorous evaluation. This version uses a backbone with a hidden dimension of 1024 and 24 transformer blocks.

By conducting ablations on the Large model, we demonstrate the Dual-Frequency Viewpoint Scheduling and Rendering-to-Geometry Gain yield consistent performance improvements regardless of the backbone scale. The Giant Model is trained on 8 H100 GPUs for 150k steps with a batch size of 1 for each, while the Large Model is trained on 8 RTX5090 GPUs for 150k steps with a batch size of 1 for each. Both are initialized from Depth Anything v3 pretrained weights.

Metrics For quantitative results on the novel view synthesis, we evaluate with the commonly used metrics PSNR, SSIM [39], and LPIPS [46]. For pose estimation, we report the area under the cumulative angular pose error curve (AUC) thresholded at 5° , 10° , and 20° . For a fair comparison, all experiments are conducted on 224×224 resolutions.

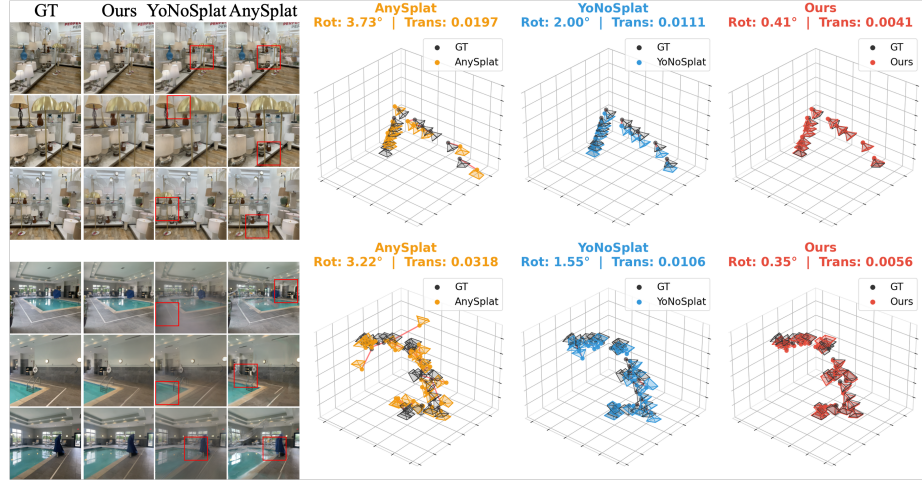


Fig. 4: Pose visualization and compared with representative methods. Left: the rendering visualization of the target views. Right: we visualize the predicted context cameras, which shows our model can handle the pose drift problem.

Datasets We conduct experiments on the DL3DV dataset [24], which consists of large-scale multi-view scenes spanning diverse indoor and outdoor environments. Each scene is processed using COLMAP [31, 32] to obtain accurate camera poses, containing approximately 250–350 frames per scene. We further train the model on the RealEstate10K (RE10K) dataset [50]. For both in-domain and cross-domain evaluation on DL3DV, RE10K, and Scannet++ [44] dataset, we follow the same settings as YoNoSplat [42].

Baselines We compared against SOTA representative sparse-view generalizable methods on novel view synthesis: 1) Pose-dependent: MVSplat [3], DepthSplat [41]; 2) Pose-free: NoPoSplat [43], AnySplat [14], and YoNoSplat [42]. For relative pose estimation, we follow the evaluation method of Depth Anything v3 [23] and then compare against SOTA methods: VGGT [36], π^3 [38], Depth Anything v3 and YoNoSplat [42].

4.2 Pose Estimation and Novel-View Synthesis

Novel View Synthesis Table 1 compares our method on the DL3DV dataset with representative feed-forward baselines for scene-level novel view synthesis, including pose-dependent methods (MVSplat and DepthSplat) and pose-free methods (NoPoSplat, AnySplat, and YoNoSplat). Our method consistently outperforms these baselines and improves over YoNoSplat by 1.6 dB PSNR in the 24-view setting. In Table 4, the results show our method trained and evaluated on RE10K still surpasses the current SOTA method YoNoSplat.

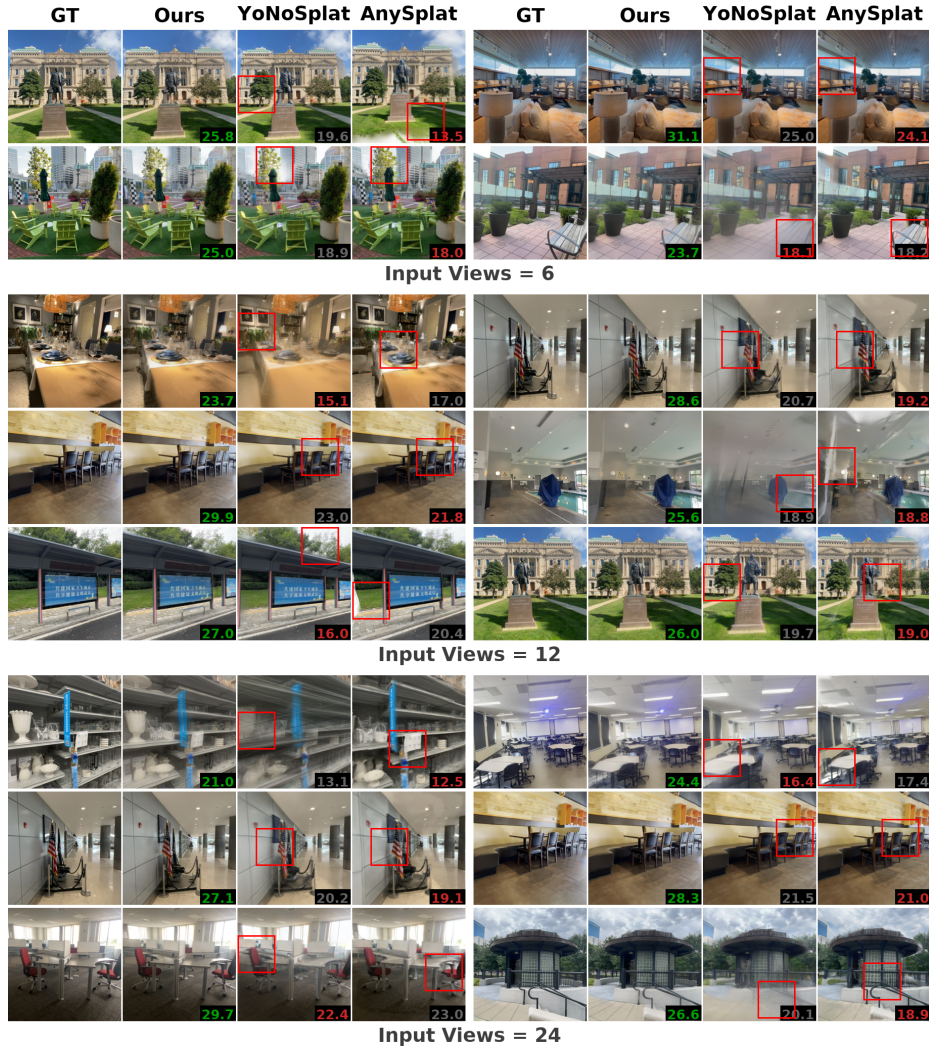


Fig. 5: Qualitative comparison of novel view synthesis on the DL3DV test set. Here we report PSNR in the right corner in the pose-free, calibration-free setting. Our model produces higher-quality results compared to others.

Figure 7 provides a qualitative visualization of DL3DV. Compared with YoNoSplat and AnySplat, our method produces sharper structures and cleaner textures under both short and long trajectories. In particular, the two baselines show more severe drift-induced degradation on long sequences, including blur, ghosting, and structural inconsistency, while our results remain visually stable with substantially reduced pose-drift artifacts.

Table 1: Novel view synthesis comparison under various input settings on the DL3DV dataset. We report results with 6, 12, and 24 input views. Our model consistently achieves the best performance, where p, k denote using ground-truth poses, intrinsics.

Method	Venue	p	k	6v			12v			24v		
				PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
MVSplat [3]	ECCV'24	✓	✓	22.659	0.760	0.173	21.289	0.709	0.224	19.975	0.662	0.269
DepthSplat [41]	CVPR'25	✓	✓	23.418	0.797	0.136	21.911	0.753	0.179	20.088	0.690	0.240
NoPoSplat [43]	ICLR'25	✓		22.766	0.743	0.179	19.380	0.563	0.318	17.860	0.495	0.397
AnySplat [14]	SA'25 (TOG)			19.027	0.554	0.235	18.940	0.549	0.262	19.703	0.596	0.249
YonoSplat [42]	ICLR'26			24.531	0.804	0.142	22.933	0.746	0.187	22.174	0.720	0.209
Erayzer $_{256 \times 256}$ [49]	CVPR'26			24.814	0.791	0.184	20.454	0.639	0.317	18.750	0.571	0.406
Ours	-			24.922	0.826	0.127	24.250	0.797	0.141	24.242	0.794	0.142

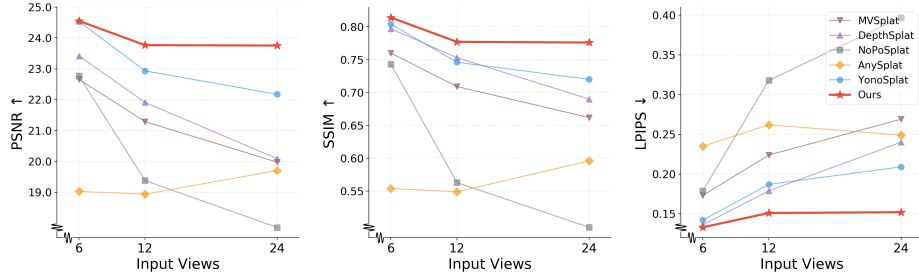


Fig. 6: As the number of input views increases (6→12→24), YoNoSplat degrades markedly, while our method remains stable across all metrics, demonstrating stronger robustness to pose drift in long sequences.

Table 2: Pose estimation comparison. Our model achieves the best performance on the DL3DV dataset with an input resolution of 224×224 under various input settings.

Model	6v			12v			24v		
	AUC@5° $\uparrow$	AUC@10° $\uparrow$	AUC@20° $\uparrow$	AUC@5° $\uparrow$	AUC@10° $\uparrow$	AUC@20° $\uparrow$	AUC@5° $\uparrow$	AUC@10° $\uparrow$	AUC@20° $\uparrow$
VGTT $_{518 \times 280}$ [36]	0.700	0.848	0.924	-	-	-	-	-	-
$\pi^3_{518 \times 280}$ [38]	0.795	0.897	0.949	-	-	-	-	-	-
NoPoSplat $_{256 \times 256}$ [43]	0.538	0.735	0.853	-	-	-	-	-	-
AnySplat $_{448 \times 448}$ [14]	0.596	0.776	0.884	0.517	0.732	0.864	0.476	0.708	0.851
YonoSplat $_{224 \times 224}$ [42]	0.833	0.917	0.958	0.804	0.902	0.951	0.778	0.885	0.942
Erayzer $_{256 \times 256}$ [49]	0.846	0.972	0.983	0.421	0.699	0.767	0.389	0.654	0.707
Ours $_{224 \times 224}$	0.967	0.983	0.992	0.961	0.979	0.990	0.949	0.972	0.985

Camera Pose Estimation Table 2 reports a clear and consistent advantage of our method in camera pose estimation across all view settings. We attribute this gain to our raymap-guided representation, which injects explicit geometric constraints into training and effectively *disambiguates the coupled optimization of geometry and pose*, leading to state-of-the-art pose accuracy. Notably, the improvement is more pronounced as the number of input views increases (12v/24v), where trajectories are longer, and drift accumulation is more severe; this trend verifies that our method better *controls cumulative pose drift* and maintains globally stable camera estimation in long-sequence scenarios.

Table 3: Zero-shot pose estimation on the RE10k dataset (6 input views). We trained the model on DL3DV and tested on RE10k dataset.

Method	AUC@5°↑	AUC@10°↑	AUC@20°↑
MASt3R _{518×288}	0.609	0.776	0.878
VGGT _{518×280}	0.566	0.753	0.867
π^3 _{518×280}	0.705	0.841	0.916
NoPoSplat _{256×256}	0.443	0.627	0.755
YoNoSplat _{224×224}	0.740	0.895	0.924
Ours _{224×224}	0.755	0.911	0.923

Table 4: NVS comparison on the RE10k dataset (6 input views). We trained the model on the RE10k dataset and tested on it.

Method	p	k	PSNR↑	SSIM↑	LPIPS↓
DepthSplat	✓	✓	24.156	0.846	0.145
NoPoSplat	✓		22.175	0.750	0.207
YoNoSplat	✓	✓	25.037	0.848	0.134
YoNoSplat	✓		25.395	0.857	0.131
YoNoSplat			24.571	0.823	0.144
Ours			25.736	0.876	0.128

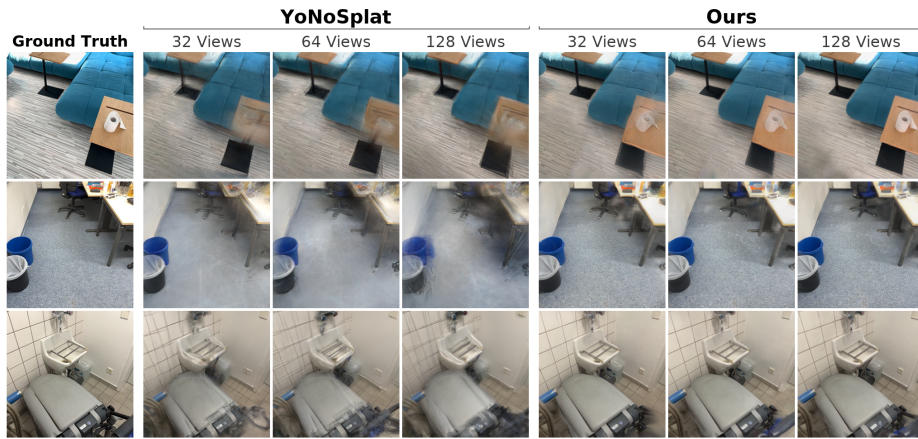


Fig. 7: Our model generalizes better than YoNoSplat on the ScanNet++ dataset and demonstrates coherent fusion of Gaussians across longer sequences. We attribute this gain to our synergistic framework, which tightly couples geometry and appearance.

Cross-Dataset Generalization To evaluate cross-dataset generalization, we train the model on DL3DV and directly evaluate on ScanNet++ without any fine-tuning. We compare against AnySplat and YoNoSplat on rendering performance. Notably, our method remains favorable even despite AnySplat’s training-domain advantage. As shown in Table 5, our approach consistently achieves state-of-the-art performance across all input-view settings.

We further evaluate zero-shot pose generalization in Table 3 by training on DL3DV and testing on RE10K. Without target-domain fine-tuning, our method achieves the best on 5° and 10°, demonstrating cross-dataset transferability of the proposed raymap-guided representation for pose estimation.

4.3 Ablation Studies

Ablations on Raymap-Guided Coupling Module(RGC) In Tab. 6, the first row (w/o raymap-guided) replaces raymap-guided geometry with geome-

Table 5: Generalization to ScanNet++. Trained on DL3DV and tested on ScanNet++, our model outperforms YonoSplat [42]. Input images are sampled from full sequences with a fixed target view. Our performance improves more on long-sequences because of accurate pose estimation.

Method	32v			64v			128v		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AnySplat	14.054	0.494	0.468	15.982	0.551	0.412	16.988	0.583	0.386
YoNoSplat	16.886	0.600	0.432	17.368	0.608	0.413	17.641	0.617	0.405
Ours	17.569	0.620	0.419	18.935	0.668	0.348	19.714	0.695	0.317

Table 6: Loss ablation across different input views on the large model.

Model	6v				12v				24v			
	PSNR↑	SSIM↑	AUC@5°↑	AUC@10°↑	PSNR↑	SSIM↑	AUC@5°↑	AUC@10°↑	PSNR↑	SSIM↑	AUC@5°↑	AUC@10°↑
w/o raymap-guided	<u>23.062</u>	<u>0.752</u>	0.821	0.907	22.375	0.717	0.753	0.867	<u>21.379</u>	<u>0.599</u>	0.690	0.821
w/o raymap loss	21.869	0.708	<u>0.869</u>	<u>0.927</u>	20.933	0.655	<u>0.814</u>	<u>0.902</u>	20.609	0.635	<u>0.782</u>	<u>0.875</u>
w/o rgb loss	12.608	0.321	0.851	0.922	12.375	0.325	0.801	0.891	12.378	0.328	0.754	0.851
Ours	23.302	0.766	0.874	0.935	<u>21.909</u>	<u>0.696</u>	0.829	0.907	21.626	0.675	0.787	0.883

try derived from camera-head pose prediction. Although this variant can still produce competitive rendering in some settings, its pose accuracy drops consistently versus the full model, showing that raymap-guided geometry is crucial for accurate pose estimation and global geometric consistency.

The second and third rows maintain the geometry guided by the raymap. Removing raymap loss (w/o raymap loss) weakens rendering quality, indicating that stronger geometry constraints from raymaps improve appearance (geometry helps appearance). Conversely, removing rendering loss (w/o rendering loss) degrades pose accuracy, showing that appearance supervision is also necessary to stabilize and refine geometry/pose (appearance helps geometry). Together, these trends verify the bidirectional coupling in our framework: geometry supervision improves rendering, and rendering supervision improves geometry. This mutual reinforcement is the *central insight* of our method and the main reason why our full model is consistently robust across view settings.

Dual-Frequency Viewpoint Scheduling Sampler Table 7 provides three findings. (1) Our sampler consistently outperforms the original sampler (similar to YoNoSplat), indicating sampling strategy is a primary limiting factor. (2) Without replay, increasing $g_{\max}$ yields a clear trade-off: performance degrades on the short range (0–50) but improves on the long range (0–150). This suggests that a larger interval expansion improves long-range coverage at the cost of short-range consistency. (3) Adding replay at $g_{\max} = \infty$ preserves strong long-range performance while recovering short-range performance. Overall, replay mitigates the short/long-range trade-off and enables a better balance between short-range and long-range fidelity.

Table 7: Ablation on *replay* and $g_{\max}$ settings.

Model	6 view, range 0–50				6 view, range 0–150			
	PSNR↑	SSIM↑	AUC@5°↑	AUC@10°↑	PSNR↑	SSIM↑	AUC@5°↑	AUC@10°↑
original sampler	22.696	0.747	0.782	0.884	18.815	0.571	0.591	0.744
$g_{\max} = 10$, w/o <i>replay</i>	23.302	0.766	0.874	0.935	19.542	0.579	0.687	0.803
$g_{\max} = 20$, w/o <i>replay</i>	22.892	0.752	0.872	0.935	19.690	0.592	0.709	0.826
$g_{\max} = \infty$, w/o <i>replay</i>	22.744	0.742	0.862	0.930	19.712	0.588	0.722	0.832
$g_{\max} = \infty$, <i>replay</i> = 50%	23.006	0.745	0.864	0.929	19.745	0.592	0.721	0.832

5 Conclusion

In this paper, we introduced a synergistic framework with a Raymap-Guided Coupling Module (RGC), which tightly couples geometric and appearance supervision. To stabilize training across wide temporal intervals, we further proposed a Dual-Frequency Viewpoint Scheduling strategy that combines easy-to-hard interval expansion with replay of short-interval samples. Extensive results and ablations show that the proposed bidirectional geometry–appearance coupling is the key to achieving robust, scalable pose-free 3D reconstruction with stronger rendering and pose performance on long-sequence generalization.

Limitations and Discussions. In the appendix, we use the same hyperparameters across all datasets for multi-dataset training. Although this generalizes well, an adaptive mechanism that can automatically adjust the sampling strategy across datasets would be a future direction, as it could improve performance and reduce tuning costs. Furthermore, our pipeline is pose-free at inference, but still relies on pose and raymap supervision during training. Extending it to a unified framework that can handle posed and unposed training data with less supervision is also another important direction for future work.

Acknowledgements

This research was supported by the Ministry of Science and ICT (MSIT) of Korea, under the National Research Foundation (NRF) grant (RS-2024-00337548); supported by a grant of the Korea-US Collaborative Research Fund (KUCRF), funded by the Ministry of Science and ICT and Ministry of Health & Welfare, Republic of Korea (grant number: RS-2024-00468417). This work was also supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2024-00457882, AI Research Hub Project) and the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25441838, Development of a human foundation model for human-centric universal artificial intelligence and training of personnel). This research was also supported by the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea

(Ministry of Science and ICT) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.RS-2026-25517417, Development of Compression, Reconstruction, and Rendering Technologies for Free-viewpoint Media).

References

1. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 19457–19467 (2024) [2](#), [4](#)
2. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. arXiv preprint arXiv:2103.15595 (2021) [4](#)
3. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European conference on computer vision. pp. 370–386. Springer (2024) [2](#), [4](#), [10](#), [12](#)
4. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. Advances in Neural Information Processing Systems **37**, 107064–107086 (2024) [4](#)
5. Dao, T., Gu, A.: Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. arXiv preprint arXiv:2405.21060 (2024) [5](#)
6. Deng, Y., Peng, S., Zhang, J., Heal, K., Sun, T., Flynn, J., Marschner, S., Chai, L.: Selfi: Self improving reconstruction engine via 3d geometric feature alignment. arXiv preprint arXiv:2512.08930 (2025) [5](#)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) [4](#), [9](#)
8. Fan, Z., Cong, W., Wen, K., Wang, K., Zhang, J., Ding, X., Xu, D., Ivanovic, B., Pavone, M., Pavlakos, G., et al.: Instantsplat: Sparse-view gaussian splatting in seconds. arXiv preprint arXiv:2403.20309 (2024) [5](#)
9. Fu, Y., Liu, S., Kulkarni, A., Kautz, J., Efros, A.A., Wang, X.: Colmap-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20796–20805 (2024) [5](#)
10. Hong, S., Jung, J., Shin, H., Han, J., Yang, J., Luo, C., Kim, S.: Pf3plat: Pose-free feed-forward 3d gaussian splatting. arXiv preprint arXiv:2410.22128 (2024) [5](#)
11. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023) [5](#)
12. Huang, R., Mikolajczyk, K.: No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27947–27957 (2025) [5](#)
13. Imtiaz, T., Chai, L., Heal, K., Luo, X., Park, J., Dy, J., Flynn, J.: Lvt: Large-scale scene reconstruction via local view transformers. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025) [5](#)
14. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., et al.: Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. ACM Transactions on Graphics (TOG) **44**(6), 1–16 (2025) [2](#), [3](#), [5](#), [7](#), [10](#), [12](#)

15. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: Lvsm: A large view synthesis model with minimal 3d inductive bias. arXiv preprint arXiv:2410.17242 (2024) [5](#)
16. Kang, G., Nam, S., Yang, S., Sun, X., Khamis, S., Mohamed, A., Park, E.: ilrm: An iterative large 3d reconstruction model. arXiv preprint arXiv:2507.23277 (2025) [2](#), [5](#)
17. Kang, G., Yang, S., Nam, S., Lee, Y., Kim, J., Park, E.: Multi-view pyramid transformer: Look coarser to see broader. arXiv preprint arXiv:2512.07806 (2025) [5](#)
18. Kang, G., Yoo, J., Park, J., Nam, S., Im, H., Shin, S., Kim, S., Park, E.: Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22012–22022 (2025) [2](#), [5](#)
19. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025) [5](#)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) [2](#), [4](#)
21. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024) [5](#)
22. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5741–5751 (2021) [5](#)
23. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) [5](#), [6](#), [9](#), [10](#)
24. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: D13dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22160–22169 (2024) [10](#)
25. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: Gnerf: Gan-based neural radiance field without posed camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6351–6361 (2021) [5](#)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) [2](#), [4](#)
27. Nam, S., Sun, X., Kang, G., Lee, Y., Oh, S., Park, E.: Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26683–26693 (2025) [2](#), [4](#)
28. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) [8](#), [9](#)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems **32** (2019) [9](#)

30. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021) [6](#)
31. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016) [3](#), [5](#), [10](#)
32. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European conference on computer vision. pp. 501–518. Springer (2016) [10](#)
33. Sun, X., Jiang, H., Liu, L., Nam, S., Kang, G., Wang, X., Sui, W., Su, Z., Liu, W., Wang, X., et al.: Uni3r: Unified 3d reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. arXiv preprint arXiv:2508.03643 (2025) [2](#), [3](#), [5](#)
34. Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., et al.: Learning to (learn at test time): Rnns with expressive hidden states. arXiv preprint arXiv:2407.04620 (2024) [5](#)
35. Wang, C., Tan, H., Yifan, W., Chen, Z., Liu, Y., Sunkavalli, K., Bi, S., Liu, L., Hu, Y.: ttllrm: Test-time training for long context and autoregressive 3d reconstruction. arXiv preprint arXiv:2602.20160 (2026) [5](#)
36. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025) [2](#), [5](#), [10](#), [12](#)
37. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024) [5](#)
38. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: pi3: Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025) [2](#), [5](#), [10](#), [12](#)
39. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) [9](#)
40. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF—: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021) [5](#)
41. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depth-splat: Connecting gaussian splatting and depth. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16453–16463 (2025) [2](#), [4](#), [10](#), [12](#)
42. Ye, B., Chen, B., Xu, H., Barath, D., Pollefeys, M.: Yonosplat: You only need one model for feedforward 3d gaussian splatting. arXiv preprint arXiv:2511.07321 (2025) [2](#), [3](#), [5](#), [7](#), [10](#), [12](#), [14](#)
43. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. arXiv preprint arXiv:2410.24207 (2024) [2](#), [3](#), [5](#), [10](#), [12](#)
44. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023) [10](#)
45. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 1–19. Springer (2024) [5](#)

46. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 586–595 (2018) [9](#)
47. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21936–21947 (2025) [5](#)
48. Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W.T., Tan, H.: Test-time training done right. arXiv preprint arXiv:2505.23884 (2025) [5](#)
49. Zhao, Q., Tan, H., Wang, Q., Bi, S., Zhang, K., Sunkavalli, K., Tulsiani, S., Jiang, H.: E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. arXiv preprint arXiv:2512.10950 (2025) [2](#), [5](#), [8](#), [12](#)
50. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018) [10](#)
51. Ziwen, C., Tan, H., Zhang, K., Bi, S., Luan, F., Hong, Y., Fuxin, L., Xu, Z.: Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4349–4359 (2025) [5](#)