StructSplat: Generalizable 3D Gaussian Splatting from Uncalibrated Sparse Views

Jia-Chen Zhao[1,2], Beiqi Chen[1], Xinyang Chen[1]✉, Guangcong Wang[2,3]✉, and Liqiang Nie[1]

[1] Harbin Institute of Technology (Shenzhen)
[2] Great Bay University
[3] Guangzhou CloudButterfly Technology Co., Ltd.
jc-zhao@outlook.com, {chenxinyang95, wanggc3, nieliqiang}@gmail.com
Project page: https://structsplat.github.io
Code available in: https://github.com/J-C-Zhao/StructSplat

Fig. 1: Qualitative comparison for novel-view synthesis from uncalibrated sparse views. Our method demonstrates fewer artifacts, less blur, and less shifting.

Abstract. We present **StructSplat**, a feed-forward and generalizable 3D Gaussian reconstruction framework that operates directly on uncalibrated images without requiring camera parameters. Existing methods either rely on per-scene optimization or assume known camera poses, and often entangle geometry and appearance within a unified backbone, limiting reconstruction fidelity and generalization. Our key idea is to adopt a **structured representation** that organizes geometry, semantic, and texture cues with explicit roles in the reconstruction process. Specifically, we introduce a pixel-aligned feature injection mechanism to enable accurate texture modeling from 2D observations, incorporate semantic-aware priors to improve global consistency, and design a camera alignment strategy to prevent information leakage and improve generalization. Experiments show that our method significantly outperforms prior approaches on challenging benchmarks. On DL3DV, our method achieves 28.045 PSNR, surpassing AnySplat (22.377) by +5.67 dB. In cross-dataset evaluation, our method achieves +1.94 dB over AnySplat on ACID and +1.72 dB on RealEstate10K.

Keywords: Generalizable; 3D Gaussian Splatting; Feed-Forward

✉ Corresponding authors.

1 Introduction

3D Gaussian Splatting (3DGS) [10,24,55] and Neural Radiance Fields (NeRF) [1, 2,31,44] have significantly advanced 3D representations and enabled high-quality novel view synthesis. To relax the dependence on accurate camera intrinsics and extrinsics, some studies [3, 11, 49, 50, 52, 54] explore pose-free or intrinsic-free reconstruction settings. However, they rely on *per-scene optimization*, requiring iterative fitting for each individual scene. This paradigm is computationally expensive and time-consuming, and more importantly, it lacks generalization ability, as the learned representation cannot be directly transferred to unseen scenes.

Recent efforts [5, 8, 15, 20, 30, 51, 56, 61] have moved beyond per-scene optimization toward feed-forward, generalizable novel view synthesis. One line of work [6,8,21,30,51] constructs 3D cost volumes via plane-sweep or homography-based warping, projecting source-view features across depth candidates for aggregation and reconstruction. Another line [4,5,12,17,41] leverages epipolar constraints and attention to implicitly encode 3D geometry without explicit cost volumes. More recently, large-model-based methods [15,20,26,43,56,61] learn strong priors from large-scale data to reconstruct 3D scenes with minimal geometric constraints. Despite promising generalization, these approaches still *require accurate camera intrinsics and extrinsics* obtained via SfM pipelines [7, 32, 37, 46], limiting their applicability in real-world settings.

To address these problems, some generalizable models focus on pose-free or intrinsic-free novel view synthesis [27, 40, 53, 58]. For example, NoPoSplat [53] addresses scale ambiguity via camera intrinsic token embedding but does not use poses. Splatt3R [40] leverages MASt3R's [25] pretrained ViT [9] encoder and cross-attention transformers to encode uncalibrated images and predict Gaussian parameters. FLARE [58] adopts a cascaded paradigm where camera poses are first estimated and then used to guide geometry and appearance learning. Depth Anything 3 [27] performs feed-forward reconstruction by introducing a DPT head after the geometry backbone for pixel-aligned Gaussian prediction with a pose-adaptive design.

Despite recent progress, feed-forward reconstruction from uncalibrated images remains challenging. *First*, most existing methods lack semantic-aware modeling, limiting their ability to leverage high-level priors for robust generalization. *Second*, geometry and appearance are often entangled within a unified backbone, which is suboptimal for recovering high-frequency textures and faithful colors, as compact 3D features are inherently biased toward spatial smoothness and geometric consistency, making them insufficient for modeling fine-grained, view-dependent appearance details. *Third*, some approaches jointly estimate camera parameters using both source and target views, and then perform feed-forward rendering in a shared coordinate system, which may introduce target-view information through cross-view attention during rendering, leading to information leakage and degraded generalization.

In this paper, we propose **StructSplat**, a feed-forward and generalizable 3D Gaussian reconstruction framework that operates directly on uncalibrated

images without requiring camera intrinsics or extrinsics. Our approach adopts a **structured representation** that organizes geometry, semantic, and texture information with explicit roles in the reconstruction process. Specifically, we introduce a geometry encoder to provide reliable structural guidance, a semantic encoder to incorporate high-level priors for global consistency, and a lightweight texture encoder to capture fine-grained appearance details. The texture encoder further leverages a pixel-aligned feature injection mechanism, establishing a direct pathway from 2D observations to Gaussian attributes, enabling accurate recovery of high-frequency textures and consistent colors. Furthermore, we design a **camera alignment strategy** that estimates camera parameters with and without target views and aligns them into a unified coordinate system, enforcing a clean separation between source and target information during both training and inference, thereby preventing information leakage and improving generalization.

Overall, our main contributions are summarized as follows:

- We propose **StructSplat**, a feed-forward and generalizable 3D Gaussian reconstruction framework that operates on uncalibrated images without requiring camera parameters.
- We introduce a structured representation that organizes geometry, semantic, and texture cues with explicit functional roles for improved reconstruction quality.
- We design a camera alignment strategy to prevent information leakage and ensure robust generalization under uncalibrated settings.
- Extensive experiments demonstrate that our method significantly outperforms prior approaches on challenging benchmarks.

2 Related Work

Generalizable Novel View Synthesis. To overcome the costly per-scene optimization required by NeRF [1,2,31,44] and 3DGS [10,24,55], recent studies have shifted toward feed-forward, generalizable models [5, 8, 15, 20, 30, 51, 56, 61]. By learning priors from large-scale datasets, these methods enable efficient inference and can be broadly categorized by their multi-view aggregation strategies. *Cost volume-based methods* [6, 8, 21, 30] construct 3D cost volumes via plane-sweep or homography warping, projecting source features across discretized depths for aggregation. DepthSplat [51] further incorporates monocular priors to improve robustness in textureless regions. *Geometry-aware transformer methods* [4, 5, 12, 17, 41] implicitly model 3D relationships via epipolar-constrained attention, restricting interactions along 1D epipolar lines instead of constructing explicit cost volumes. *Large-model-based methods* [15, 20, 43, 56, 61] learn strong priors directly from large-scale data, reducing reliance on explicit geometric modeling. For instance, GS-LRM [56] extends LRM [15] to generate 3D Gaussians, while LVSM [20] performs novel view synthesis from sparse inputs using transformer architectures. Despite their efficiency, most methods still rely on pre-computed camera poses from Structure-from-Motion (SfM) pipelines [7, 32, 37, 46], which

are computationally expensive and brittle in texture-poor or low-overlap scenarios [11]. This reliance limits the end-to-end efficiency of feed-forward models and restricts their applicability in unconstrained real-world settings.

Pose-free or Intrinsic-free Novel View Synthesis. To simplify the 3D reconstruction pipeline and remove the dependence on pre-computed camera parameters, recent studies jointly estimate scene geometry and camera poses. *Per-scene pose-free optimization methods* [3, 11, 49, 50, 52, 54] optimize camera parameters together with the scene representation. For example, FrozenRecon [50] aligns monocular depth using a trainable module, while NoPe-NeRF [3] leverages undistorted depth priors to constrain relative poses. CF-3DGS [11] further adopts an incremental strategy to estimate poses and grow a global Gaussian representation. However, these methods rely on costly per-scene optimization, limiting scalability and practical deployment. *Generalizable feed-forward pose-free methods* [13,14,23,38,42,47,53,58] predict 3D representations from unposed inputs in a single forward pass. SelfSplat [23] integrates self-supervised depth and pose estimation into 3DGS, while PF3plat [13] adopts a two-stage design that first predicts coarse geometry and poses before refinement. SpatialSplat [38] and Uni3R [42] adapt feed-forward 3D frameworks for Noval-View Segmentation. Despite improved efficiency, these methods typically assume known camera intrinsics, preventing a fully end-to-end 2D-to-3D mapping. *Fully intrinsic-free and pose-free methods* [18, 19, 27, 40] further remove both intrinsic and extrinsic dependencies. For instance, AnySplat [19] employs a VGGT backbone [45] to directly predict camera parameters and Gaussian primitives from unconstrained views, while RayZer [18] encodes camera information into latent representations. Depth Anything 3 [27] introduces a 3D foundation model with a Gaussian decoding head. Despite these advances, existing feed-forward approaches largely under-explore rich 2D semantics and high-frequency texture details, which are crucial for improving appearance fidelity and reconstruction quality.

3 Method

3.1 Problem Formulation

Given a set of uncalibrated source images $\mathcal{I}^{\mathrm{s}} = \{I_i^{\mathrm{s}} \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{N}$, our goal is to infer a 3D scene representation in the form of a set of Gaussians $\mathcal{G} = \{(\mu_j, \alpha_j, c_j, s_j, r_j)\}_{j=1}^{NMHW}$ in a feed-forward manner, which is given by $\mathcal{G} = f_\theta(\mathcal{I}^{\mathrm{s}})$, where f_θ is a neural network parameterized by θ. Notably, our formulation does not require camera intrinsics or extrinsics. Each Gaussian is parameterized by its center μ_j, opacity α_j, color c_j, scale s_j, and rotation r_j, and we predict Gaussians per source pixel. Given the reconstructed Gaussians $\mathcal{G}$, novel views are synthesized via differentiable splatting and rasterization [24], which is given by $\widehat{\mathcal{I}}^{\mathrm{t}} = \mathrm{Render}(\mathcal{G}, \mathbf{p}^{\mathrm{t}})$, where $\widehat{\mathcal{I}}^{\mathrm{t}}$ denotes the rendered image at target viewpoint $\mathbf{p}^{\mathrm{t}}$.

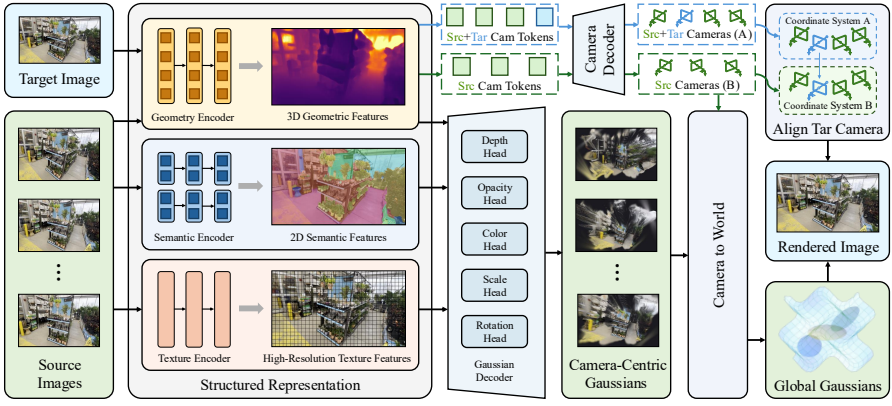

Fig. 2: Overview of our proposed StructSplat. Given uncalibrated source images, we perform feed-forward 3D reconstruction without camera parameters by adopting a structured representation that organizes texture, semantic, and geometric cues. Encoders extract multi-level features, which drive two decoding pathways: a Gaussian decoder predicts camera-centric Gaussians, while a camera decoder estimates camera parameters followed by a camera alignment module for unified registration. The aligned Gaussians are then transformed to a global representation and rendered via differentiable splatting.

3.2 Overview of StructSplat

The overview of StructSplat is illustrated in Fig. 2. Given a set of uncalibrated source images, our goal is to perform feed-forward 3D reconstruction without relying on camera parameters. To this end, we adopt a **structured representation** that organizes multi-level visual cues for effective geometry and appearance modeling. Specifically, the input images are processed by a set of encoders to extract complementary features, including high-resolution texture features, 2D semantic features, and 3D geometric features. These features are then used to drive two decoding pathways. The Gaussian decoder, equipped with multiple specialized heads, densely predicts the attributes of camera-centric Gaussians. Meanwhile, a camera decoder estimates camera parameters to resolve spatial inconsistencies across views, followed by a camera alignment module that registers all views into a unified coordinate system. Finally, guided by the predicted cameras, a camera-to-world transformation converts camera-centric Gaussians into a global representation, which is rendered via differentiable splatting to synthesize novel views.

3.3 Structured Generalizable 3D Representation

A straightforward way to build a generalizable 3DGS model is to attach a Gaussian head to a strong 3D geometry backbone (e.g., MASt3R [25] or VGGT [45]). For example, Splatt3R [40] leverages MASt3R to encode uncalibrated images

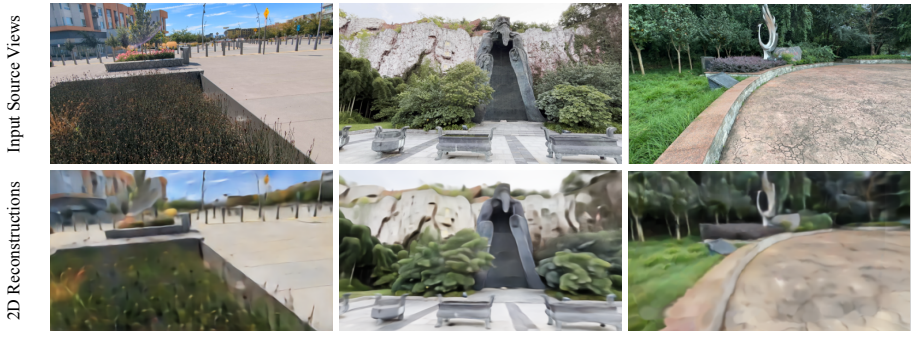

Fig. 3: Geometry-oriented features fail to capture appearance details. We attach a color head to VGGT [45], a strong pre-trained geometry encoder, to directly perform 2D reconstruction, but observe poor color fidelity and texture details even on training views. This reveals that geometry-oriented features fail to capture appearance information, motivating our structured representation with a dedicated appearance pathway.

and predict Gaussian parameters, while Depth Anything 3 [27] introduces a foundation geometry model with an additional fine-tuned DPT head [35] to infer pixel-aligned 3D Gaussians. However, as illustrated in Fig. 3, directly relying on geometry features is insufficient for appearance modeling. When attaching a learnable color head to a strong pre-trained geometry encoder (e.g., VGGT) for 2D image reconstruction, we observe poor color fidelity and texture details, even on training views. This indicates a mismatch between geometry-oriented features and appearance reconstruction, since compact 3D representations are inherently biased toward spatial smoothness and geometric consistency, limiting their ability to capture fine-grained, view-dependent appearance details.

Motivated by this observation, we propose a structured representation that organizes multi-level visual cues. By integrating high-resolution 2D texture features and semantic priors with 3D geometry, our design preserves fine-grained appearance details while maintaining reliable structural consistency.

Geometry Encoder. We employ a VGGT-based encoder to capture global geometric structure and provide reliable 3D priors for reconstruction. Its attention-based architecture enables effective reasoning over multi-view relationships and facilitates the aggregation of long-range contextual information across views. The encoder produces tokenized 3D features that encode scene geometry in a view-consistent manner, which serve as the foundation for subsequent depth and camera estimation. This design ensures robust geometric understanding and provides stable structural guidance for Gaussian reconstruction. We decode the 3D token features into a pixel-aligned depth map, which is then transformed into the world coordinate system to present the positions μ of the 3D Gaussians.

Semantic Encoder. To complement the geometry encoder, we introduce a semantic encoder to provide high-level contextual priors for reconstruction. Un-

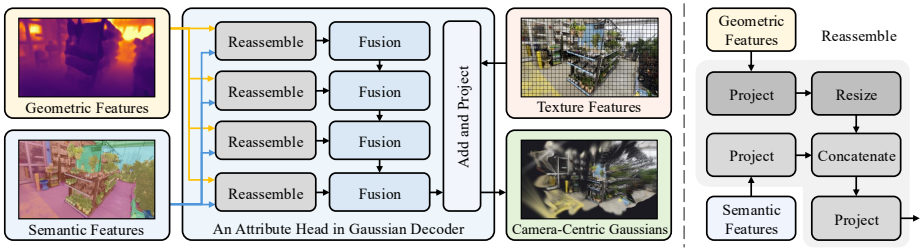

Fig. 4: Detailed architecture of the proposed Gaussian decoder head. By modifying the standard DPT [35] heads, our decoder effectively integrates a structured representation to predict camera-centric Gaussian attributes. As detailed on the right, the redesigned *Reassembling* block spatially aligns and concatenates geometric and semantic features, using projection layers to maintain channel dimensions. After processing through cascaded *Reassembling* and *Fusion* blocks, high-resolution texture features are injected at a late stage to explicitly preserve fine visual details before yielding the final output.

like geometry features that focus on structure, semantic representations capture object-level organization and global scene understanding, which are essential for resolving ambiguities under sparse or uncalibrated views. Specifically, we extract semantic features from each input image using a pre-trained vision foundation model (e.g., DINOv3 [39]). These features encode view-invariant correspondences and structural regularities across images. We integrate the semantic features into the reconstruction pipeline to guide Gaussian prediction, encouraging globally consistent structure and coherent appearance across views. This design improves robustness to challenging scenarios such as large viewpoint changes, leading to better generalization on unseen scenes.

Texture Encoder. Complementary to the semantic and geometry encoders, we introduce a texture encoder to capture high-frequency appearance details directly from input images. Unlike semantic features that encode high-level abstractions, appearance is inherently pixel-aligned and requires preserving local spatial fidelity. Specifically, the texture encoder employs lightweight convolutional layers to extract high-resolution features from each source view. These features are maintained in the image space and retain fine-grained color and texture information. We then inject the resulting features into the 3D feature stream through a pixel-aligned feature injection mechanism, establishing a direct correspondence between 2D observations and Gaussian attributes. By providing explicit appearance cues to the depth and Gaussian heads, this design enables accurate modeling of fine textures and consistent colors.

Gaussian Prediction. The Gaussian prediction stage consists of a Gaussian decoder and a subsequent coordinate transformation. Initially, the Gaussian decoder predicts the attributes of Gaussian primitives in the local camera coordinate system of each source view, leveraging the structured representations. Sub-

sequently, these camera-centric Gaussians are transformed into a unified world coordinate system based on the estimated camera parameters.

As illustrated in Fig. 4, we modify the standard DPT [35] heads to effectively integrate multi-scale semantic features and high-resolution texture cues. Specifically, the *Reassembling* block is redesigned to align and concatenate geometric and semantic features, which are then projected to preserve the original DPT channel dimensions. The detailed structure of the *Reassembling* block is shown on the right side of Fig. 4. After passing through cascaded *Reassembling* and *Fusion* blocks, high-resolution texture features are injected just before the final prediction layer to produce the Gaussian attributes. This late-stage integration explicitly preserves fine visual details and reduces texture degradation.

Finally, to construct a coherent 3D scene, the camera-centric predictions must be globally aligned. The decoder outputs the spatial position (parameterized by depth along the camera's z-axis) and the rotation quaternion defined in the local camera coordinate system for each Gaussian primitive. Consequently, a rigorous coordinate transformation step is applied to map these camera-space attributes into a unified world coordinate framework. This transformation utilizes the intrinsic and extrinsic parameters derived from the camera decoder, ensuring strict spatial consistency across all source views.

3.4 Camera Alignment for Leakage-Free Training

A key challenge arises during training: the target and source camera extrinsics must be established in a unified coordinate system. Simply incorporating target images into the source set to unify camera parameters and then separating their features for Gaussian prediction may introduce information leakage, which compromises generalization. As shown in Fig. 5, due to the cross-attention mechanism in the geometry extractor, target image information inevitably leaks into the source features. This leakage grants the Gaussian decoder implicit access to the target view's ground truth, compromising training supervision and evaluation fairness. To address this problem, we propose a parallel-stream camera pose alignment strategy that effectively isolates source and target features, enabling robust training and a fair evaluation within the existing framework.

Our approach processes two image sets in parallel through the geometry feature extractor: a mixed set containing both source and target views, and a source-only set. The camera head then decodes the camera parameters for each set. Initially, these two sets of parameters reside in disparate world coordinate systems. We therefore align the target cameras from the mixed set to the coordinate system of the source-only set, using the source cameras as anchors. This process involves simultaneous optimization of rotation and translation.

Rotation Alignment. To minimize computational overhead while preserving precision, we directly align the rotation quaternions predicted by the camera head. Let $q^A_{\mathrm{s},i}$ and $q^B_{\mathrm{s},i}$ denote the rotation quaternions of the i-th source view predicted from the mixed set A and the source-only set B, respectively. We seek a unit quaternion Δq that aligns $q^A_{\mathrm{s},i}$ with $q^B_{\mathrm{s},i}$ for all source views i. This leads

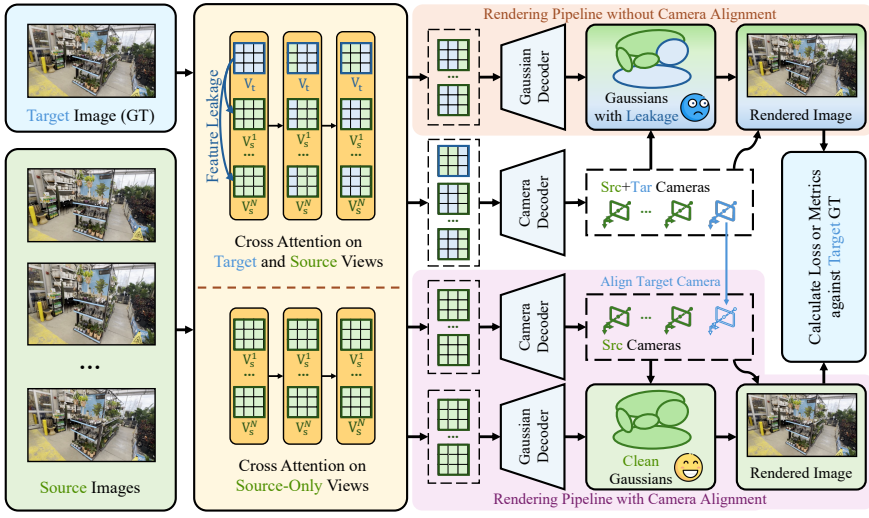

Fig. 5: Rendering pipeline with or without camera alignment. Due to the cross-attention mechanism in the geometry extractor, target image information inevitably leaks into the source features. In contrast, our camera alignment strategy allows the Gaussian decoder to utilize pure features extracted from source-only views, thereby preventing feature leakage.

to the following constrained optimization problem:

$$\Delta q^* = \arg\max_{\|\Delta q\|=1} \sum_{i=1}^{n} \mathrm{Re}\big(\Delta q\, q^A_{\mathrm{s},i}\, (q^B_{\mathrm{s},i})^{-1}\big)\,. \tag{1}$$

where $\mathrm{Re}(\cdot)$ extracts the real part of a quaternion and n is the number of source views. We solve this problem using a Lagrange multiplier formulation.

Translation Alignment. For the translation vectors $\tau^A_{\mathrm{s},i}$ and $\tau^B_{\mathrm{s},i}$, we seek a scale $\lambda \in \mathbb{R}$ and an offset $\Delta\tau \in \mathbb{R}^3$ that align $\lambda\,\tau^A_{\mathrm{s},i} + \Delta\tau$ with $\tau^B_{\mathrm{s},i}$ for all source views i. This can be formulated as a least-squares problem:

$$(\lambda^*, \Delta\tau^*) = \arg\min_{\lambda,\Delta\tau} \sum_{i=1}^{n} \left\|\lambda\,\tau^A_{\mathrm{s},i} + \Delta\tau - \tau^B_{\mathrm{s},i}\right\|_2^2. \tag{2}$$

Finally, we transform the target-view poses from the mixed set into the Gaussian representation coordinate system (i.e., the source-only set) as $q^B_{\mathrm{t},j} = \Delta q^* q^A_{\mathrm{t},j}$ and $\tau^B_{\mathrm{t},j} = \lambda^* \tau^A_{\mathrm{t},j} + \Delta\tau^*$, where $(q^A_{\mathrm{t},j}, \tau^A_{\mathrm{t},j})$ and $(q^B_{\mathrm{t},j}, \tau^B_{\mathrm{t},j})$ denote the camera poses of the j-th target view before and after alignment, respectively. As illustrated in Fig. 2, this strategy ensures that the modules responsible for predicting Gaussian attributes never access features from the target views. Meanwhile, it aligns the target views into a consistent world coordinate system, which is crucial for model generalization and fair evaluation.

Table 1: Comparison with representational reconstruction methods. The best and second-best results of methods without any camera parameters are highlighted in **bold** and underlined, respectively. Note that we evaluate all metrics without any post optimization in this table. ("Weak" denotes only relying on intrinsics and "No" denotes methods without any camera parameters.)

Method	Camera Condition	DL3DV [28]			ACID [29]			RealEstate10K [60]		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PF3plat [13]	Weak	22.579	0.731	0.186	23.401	0.729	0.279	22.766	0.784	0.190
SelfSplat [23]	Weak	18.017	0.511	0.366	23.528	0.729	0.279	22.542	0.781	0.229
NoPoSplat [53]	Weak	25.592	0.838	0.104	25.521	0.750	0.185	24.724	0.818	0.145
FLARE* [58]	Weak	23.441	0.757	0.126	24.365	0.702	0.200	23.611	0.777	0.155
Splatt3R [40]	No	15.936	0.391	0.408	11.000	0.307	0.582	12.092	0.364	0.540
RayZer [18]	No	19.802	0.527	0.286	-	-	-	-	-	-
AnySplat [19]	No	22.377	0.716	0.150	22.433	0.651	0.237	20.521	0.686	0.212
Depth Anything 3 [27]	No	20.715	0.615	0.226	20.482	0.588	0.346	18.769	0.613	0.312
StructSplat (Ours)	No	**28.045**	**0.888**	**0.091**	**24.372**	**0.712**	**0.219**	**22.240**	**0.729**	**0.201**

* FLARE [58] natively utilizes camera intrinsics as conditional input for *novel-view-synthesis* subtasks to align with the experimental settings of NoPoSplat [53].

4 Experiments

In this section, we first describe the experimental setup, including datasets and baseline methods. We then present the main comparisons with state-of-the-arts. Finally, we conduct ablation studies and further analyses to assess the contribution of each component in our framework.

4.1 Experiment Setting

Datasets and Evaluation. We evaluate our method on three large-scale datasets: DL3DV [28], RealEstate10K [60], and ACID [29]. For DL3DV, we follow prior works [13,51] and use the official benchmark split as the test set. For RealEstate10K and ACID, we use the model trained on DL3DV for cross-dataset evaluation. To ensure a fair evaluation, we carefully remove overlapping and anomalous samples from the training set. Since baseline methods render images at different resolutions, we uniformly resize and center-crop all outputs and ground-truth images to 256×256. We report PSNR, SSIM [48], and LPIPS [57] as evaluation metrics.
Baseline Methods. We comprehensively benchmark our approach against a variety of state-of-the-art baselines. Based on their input requirements, these baselines can be categorized into two main groups: 1) pose-free but intrinsics-dependent methods, which operate without camera poses but still rely on known camera intrinsics, including PF3plat [13], SelfSplat [23], NoPoSplat [53], and FLARE [58]; 2) entirely camera-parameter-free methods, comprising Splatt3R [40], RayZer [18], AnySplat [19], and Depth Anything 3 [27]. Notably, to align with the experimental settings of NoPoSplat [53], FLARE [58] natively utilizes camera intrinsics as conditional input for novel-view-synthesis subtasks.
Implementation. To optimize training efficiency and alleviate GPU memory overhead, we incorporate bfloat16 (BF16) mixed-precision training [22] along-

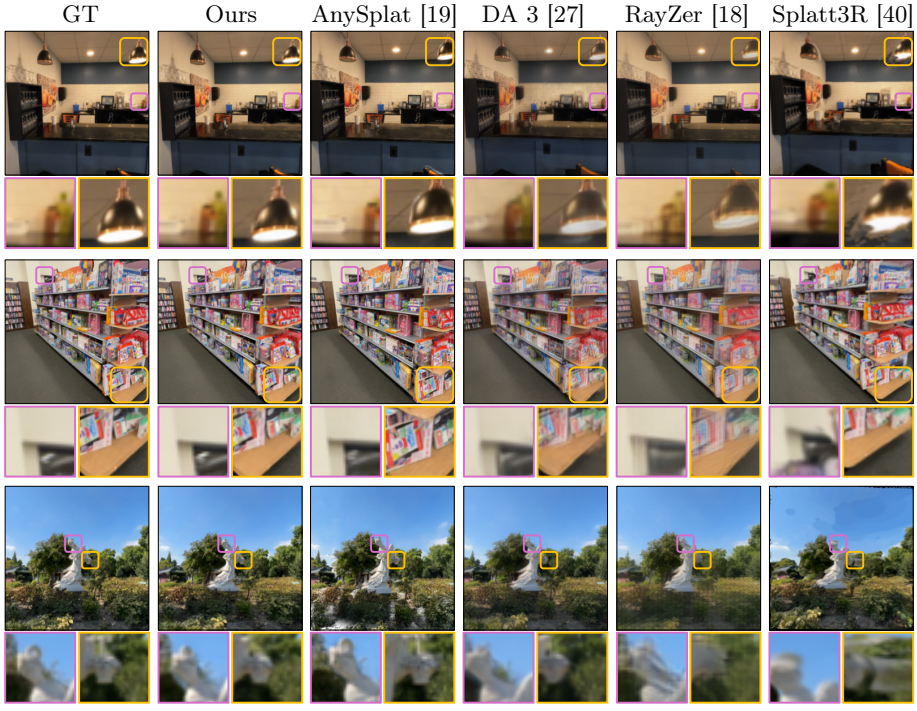

Fig. 6: Visual comparison of novel view synthesis on DL3DV [28] dataset.

side the DeepSpeed optimization library [33, 34, 36]. During the training phase, we employ the Warmup-Stable-Decay (WSD) learning rate scheduler [16], which ensures stable convergence and provides flexibility for continuous model retraining. To fully leverage the available data and enforce multi-view consistency, we jointly render both the source and target views to compute the training loss. For final evaluation, all quantitative metrics are computed exclusively on the target views.

4.2 Comparisons with State-of-the-Art Methods

Quantitative Comparison. The quantitative comparison with state-of-the-art methods is reported in Table 1. Overall, our method consistently achieves the best performance across all datasets and metrics among approaches that do not rely on camera parameters. On DL3DV, our method significantly outperforms prior works, achieving a PSNR of 28.045, which surpasses the best competing method AnySplat (22.377) by a large margin of +5.67 dB. Similar gains are observed in SSIM (0.888 vs. 0.716) and LPIPS (0.091 vs. 0.150), demonstrating substantial improvements in both structural fidelity and perceptual quality.

On ACID, we use the models trained on DL3DV as a cross-dataset evaluation. Our method achieves the best performance among camera-free approaches,

Table 2: Comparison for multi-view reconstruction on DL3DV [28]**.** The best and second-best results of methods without any camera parameters are highlighted in **bold** and underlined, respectively.

Method	2 Source Views			4 Source Views			6 Source Views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RayZer [18]	19.802	0.527	0.286	27.888	0.868	0.140	28.851	0.889	0.128
AnySplat [19]	22.377	0.716	0.150	24.499	0.769	0.112	26.529	0.841	0.085
Depth Anything 3 [27]	20.715	0.615	0.226	20.486	0.601	0.232	20.566	0.602	0.232
StructSplat (Ours)	**28.045**	**0.888**	**0.091**	**30.415**	**0.931**	**0.061**	**31.137**	**0.942**	**0.057**

with a PSNR of 24.372, outperforming AnySplat (22.433) by +1.939 dB. Consistent improvements are also observed in SSIM and LPIPS, indicating better structural fidelity and perceptual quality. Given the challenging nature of ACID, which involves complex camera motions and diverse scene layouts, these results demonstrate the robustness of our method in handling uncalibrated and geometrically challenging scenarios.

On RealEstate10K, which contains more complex real-world scenes with diverse layouts and watermark artifacts, we use the models trained on DL3DV as a cross-dataset evaluation. Our method continues to achieve the best performance among methods without camera parameters. Specifically, it achieves a PSNR of 22.240, outperforming AnySplat (20.521) by +1.719 dB, and also obtains higher SSIM (0.729 vs. 0.686) and lower LPIPS (0.201 vs. 0.212). Notably, despite the increased difficulty of this dataset, our method remains competitive with approaches that rely on camera intrinsics, such as PF3plat (22.766 PSNR), highlighting the robustness of our camera-free design.

Importantly, even compared with methods that rely on camera intrinsics ("Weak" setting), our approach achieves competitive or superior results. For instance, on DL3DV, our method (28.045 PSNR) clearly surpasses NoPoSplat (25.592), highlighting the effectiveness of our fully camera-free design.

Qualitative Comparison. Fig. 6 presents qualitative results on the DL3DV dataset [28]. As shown in the zoomed-in regions, our approach recovers fine-grained details and produces sharper edges, whereas baseline methods tend to generate blurry or distorted results. Our method also preserves more consistent textures and structural boundaries across views, leading to more faithful scene reconstruction. Additionally, visual comparisons on the RealEstate10K [60] and ACID [29] datasets are provided in Fig. 7. Despite the increased scene complexity and viewpoint variations in these datasets, our method reconstructs coherent 3D structures and produces photorealistic views that align closely with the Ground Truth.

Furthermore, we investigate the multi-view reconstruction capability of our method on DL3DV [28] using 2, 4 and 6 source views. As shown in Table 2, our method consistently outperforms other methods across different numbers of input views, demonstrating its capability for high-quality 3D reconstruction even with multiple source views.

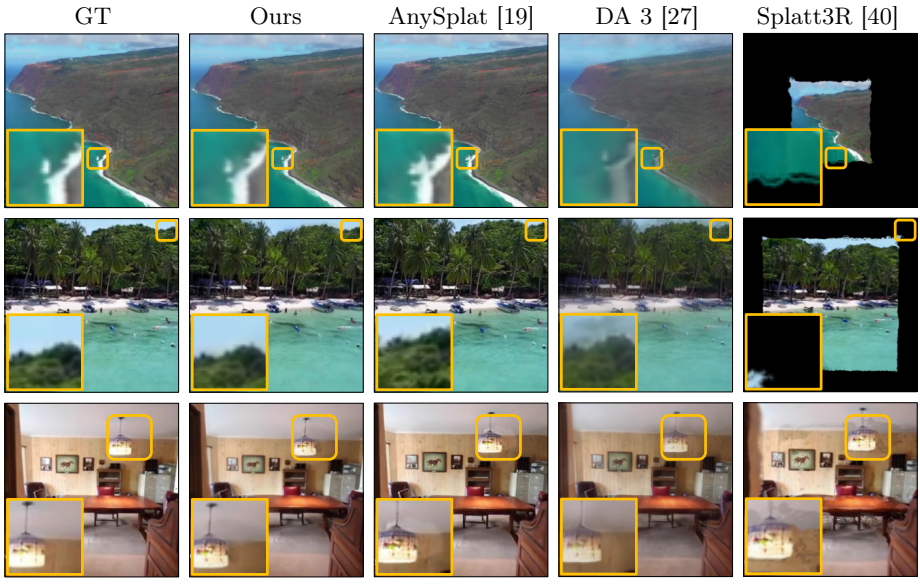

Fig. 7: Visual comparison of novel view synthesis on RealEstate10K [60] **and ACID** [29] **datasets.**

Table 3: Comparison for ablation of Semantic Features and Texture Features on DL3DV [28]**.** The best results are highlighted in **bold**.

Experiment Setting	PSNR↑	SSIM↑	LPIPS↓
Geometric Features	20.610	0.622	0.343
+ Semantic Features	26.236	0.848	0.151
+ Texture Features	**28.045**	**0.888**	**0.091**

Table 4: Impact of pose post-optimization on DL3DV. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	PSNR↑	SSIM↑	LPIPS↓
NoPoSplat [53]	<u>27.243</u>	<u>0.877</u>	<u>0.093</u>
FLARE [58]	26.916	0.865	0.099
Ours	**29.287**	**0.911**	**0.085**

4.3 Ablation Studies and Further Analyses

Impact of Semantic Encoder. To evaluate the role of semantic priors, we first augment a baseline model that uses only *Geometric Features* with *Semantic Features*. As shown in Table 3, introducing semantic features leads to a substantial performance gain, improving PSNR from 20.610 to 26.236. This significant improvement indicates that high-level semantic context plays a critical role in resolving structural ambiguities and enhancing global consistency in complex scenes.

Impact of Texture Encoder. We further analyze the contribution of pixel-aligned appearance modeling by adding *Texture Features* on top of geometric and semantic features. This results in additional performance improvements, achieving the best results across all metrics (PSNR 28.045). These gains demon-

Table 5: Comparison for different Camera Alignment strategies on DL3DV [28]**.** The best results are highlighted in **bold**.

Method	ATE_{rot} ↓	ATE_{pos} ↓
AnySplat [19]	0.0866	0.1003
StructSplat (Ours)	**0.0016**	**0.0093**

Table 6: Rendering comparison for ablation of our Camera Alignment strategy on DL3DV [28]**.** The best results are highlighted in **bold**.

Train Setting	PSNR↑	SSIM↑	LPIPS↓
w/o CA	27.338	0.879	0.097
StructSplat (Ours)	**28.045**	**0.888**	**0.091**

strate that texture features are essential for recovering fine-grained details and accurate colors, complementing the structural guidance provided by geometry and semantics.

Impact of Pose Post-optimization. We further evaluate the effect of camera pose refinement [53] by optimizing the target extrinsics for a few iterations. As shown in Table 4, our method continues to outperform NoPoSplat [53] and FLARE [58], indicating that the advantage of our structured representation persists even when additional pose refinement is applied.

Impact of Camera Alignment. To ensure robust feed-forward reconstruction, we incorporate a camera alignment strategy to enforce a consistent coordinate system across views. AnySplat [19] also employs a simpler strategy that aligns the translation scale during evaluation. We compare the Absolute Trajectory Error (ATE) [59] of target camera poses obtained using these two alignment strategies in Table 5. Our StructSplat achieves lower errors in both rotation and translation. As shown in Table 6, removing this component leads to noticeable performance degradation, demonstrating its critical role in maintaining reconstruction fidelity. The alignment mechanism effectively prevents information leakage between source and target views, which is crucial for learning a more generalizable representation. As a result, the model achieves stronger generalization ability to unseen scenes, particularly in cases with complex geometry.

Real-world Applications. We further evaluate our method on real-world, in-the-wild image collections without camera calibration, including Internet videos and handheld videos captured in casual settings. Despite the absence of camera parameters, our method reconstructs coherent 3D structures and produces consistent novel views under challenging conditions such as complex scene layouts.

5 Conclusion

In this paper, we presented StructSplat, a feed-forward and generalizable 3D Gaussian reconstruction framework that operates on uncalibrated images without requiring camera parameters. We addressed key limitations of existing methods by introducing a structured representation that models geometry, appearance, and semantics in their native spaces, together with a pixel-aligned feature injection mechanism and a camera alignment strategy to prevent information leakage. Extensive experiments demonstrate that our method achieves superior

performance over prior approaches. Despite these improvements, challenges remain under extremely sparse views or severe occlusions, and modeling complex view-dependent effects and lighting remains difficult. Future work will focus on more robust geometry priors and improved appearance modeling, as well as extending the framework to dynamic scenes and large-scale environments.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62506063 and No. 62306085), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515110178), the Start-up Funding (Grant No. YJKY230070), the Shenzhen College Stability Support Plan (Grant No. GXWD20231130151329002) and the Shenzhen Science and Technology Program (Grant No. KQTD20240729102207002). Computational resources were provided by the Songshan Lake High Performance Computing Center (SSL-HPC) at Great Bay University. This work was also supported by the Guangdong Research Team for Communication and Sensing Integrated with Intelligent Computing (Project No. 2024KCXTD047).

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV. pp. 19697–19705 (2023)
3. Bian, W., Wang, Z., Li, K., Bian, J.W.: NoPe-NeRF: Optimising neural radiance field with no pose prior. In: CVPR. pp. 4160–4169 (2023)
4. Bourigault, E., Bourigault, P.: MVDiff: Scalable and flexible multi-view diffusion for 3d object reconstruction from single-view. In: CVPR. pp. 7579–7586 (2024)
5. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: PixelSplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: CVPR. pp. 19457–19467 (2024)
6. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: MVSNerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV. pp. 14124–14133 (2021)
7. Chen, S., Li, X., Wang, Z., Prisacariu, V.A.: DFNet: Enhance absolute pose regression with direct feature matching. In: ECCV. pp. 1–17 (2022)
8. 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: ECCV. pp. 370–386 (2024)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Fang, G., Wang, B.: Mini-splatting: Representing scenes with a constrained number of gaussians. In: ECCV. pp. 165–181. Springer (2024)

11. Fu, Y., Wang, X., Liu, S., Kulkarni, A., Kautz, J., Efros, A.A.: COLMAP-free 3d gaussian splatting. In: CVPR. pp. 20796–20805 (2024)
12. He, Y., Yan, R., Fragkiadaki, K., Yu, S.I.: Epipolar transformers. In: CVPR. pp. 7776–7785 (2020)
13. Hong, S., Jung, J., Shin, H., Han, J., Yang, J., Luo, C., Kim, S.: PF3plat: Pose-free feed-forward 3d gaussian splatting for novel view synthesis. In: ICML (2025)
14. Hong, S., Jung, J., Shin, H., Yang, J., Kim, S., Luo, C.: Unifying correspondence, pose and nerf for generalized pose-free novel view synthesis. In: CVPR. pp. 20196–20206 (2024)
15. 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. In: ICLR (2024)
16. Hu, S., Tu, Y., Han, X., et al.: Minicpm: Unveiling the potential of small language models with scalable training strategies. arXiv preprint arXiv:2404.06395 (2024)
17. Jia, H., Zhu, L., Zhao, N.: H3R: Hybrid multi-view correspondence for generalizable 3d reconstruction. In: ICCV. pp. 7655–7665 (2025)
18. Jiang, H., Tan, H., Wang, P., et al.: Rayzer: A self-supervised large view synthesis model. In: ICCV. pp. 4918–4929 (2025)
19. 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. arXiv preprint arXiv:2505.23716 (2025)
20. 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. In: ICLR. vol. 2025, pp. 60001–60021 (2025)
21. Johari, M.M., Lepoittevin, Y., Fleuret, F.: Geonerf: Generalizing nerf with geometry priors. In: CVPR. pp. 18365–18375 (2022)
22. Kalamkar, D., Mudigere, D., Mellempudi, N., Das, D., Banerjee, K., Avancha, S., Vooturi, D.T., Jammalamadaka, N., Huang, J., Yuen, H., et al.: A study of bfloat16 for deep learning training. arXiv preprint arXiv:1905.12322 (2019)
23. 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: CVPR. pp. 22012–22022 (2025)
24. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
25. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: ECCV. pp. 71–91 (2024)
26. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. ICLR (2024)
27. 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)
28. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: CVPR. pp. 22160–22169 (2024)
29. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: ICCV. pp. 14458–14467 (2021)
30. Liu, T., Wang, G., Hu, S., Shen, L., Ye, X., Zang, Y., Cao, Z., Li, W., Liu, Z.: MVSGaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo. In: ECCV. pp. 37–53 (2024)

31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
32. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: ECCV. pp. 58–77 (2024)
33. Rajbhandari, S., Rasley, J., Ruwase, O., He, Y.: Zero: Memory optimizations toward training trillion parameter models. In: Int. Conf. High Perform. Comput. Netw. Storage Anal. pp. 1–16 (2020)
34. Rajbhandari, S., Ruwase, O., Rasley, J., Smith, S., He, Y.: Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In: Int. Conf. High Perform. Comput. Netw. Storage Anal. pp. 1–14 (2021)
35. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV. pp. 12179–12188 (2021)
36. Ren, J., Rajbhandari, S., Aminabadi, R.Y., Ruwase, O., Yang, S., Zhang, M., Li, D., He, Y.: Zero-offload: Democratizing billion-scale model training. In: USENIX Annu. Tech. Conf. pp. 551–564 (2021)
37. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016)
38. Sheng, Y., Deng, J., Zhang, X., Zhang, Y., Hua, B., Zhang, Y., Ji, J.: Spatialsplat: Efficient semantic 3d from sparse unposed images. In: ICCV. pp. 26404–26414 (2025)
39. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
40. Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. arXiv preprint arXiv:2408.13912 (2024)
41. Suhail, M., Esteves, C., Sigal, L., Makadia, A.: Generalizable patch-based neural rendering. In: ECCV. pp. 156–174 (2022)
42. Sun, X., Jiang, H., Liu, L., et al.: Uni3r: Unified 3d reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. In: CVPR. pp. 33280–33290 (2026)
43. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: ECCV. pp. 1–18 (2024)
44. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV. pp. 9065–9076 (2023)
45. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR. pp. 5294–5306 (2025)
46. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: VGGSfM: Visual geometry grounded deep structure from motion. In: CVPR. pp. 21686–21697 (2024)
47. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. ICLR (2024)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
49. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2022)
50. Xu, G., Yin, W., Chen, H., Shen, C., Cheng, K., Zhao, F.: FrozenRecon: Pose-free 3d scene reconstruction with frozen depth models. In: ICCV. pp. 9276–9286 (2023)
51. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: DepthSplat: Connecting gaussian splatting and depth. In: CVPR. pp. 16453–16463 (2025)

52. Yan, Q., Wang, Q., Zhao, K., Chen, J., Li, B., Chu, X., Deng, F.: CF-NeRF: Camera parameter free neural radiance fields with incremental learning. In: AAAI. vol. 38, pp. 6440–6448 (2024)
53. 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)
54. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: iNeRF: Inverting neural radiance fields for pose estimation. In: IROS. pp. 1323–1330 (2021)
55. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: CVPR. pp. 19447–19456 (2024)
56. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: GS-LRM: Large reconstruction model for 3d gaussian splatting. In: ECCV. pp. 1–19 (2024)
57. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
58. 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: CVPR. pp. 21936–21947 (2025)
59. Zhang, Z., Scaramuzza, D.: A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: IROS. pp. 7244–7251 (2018)
60. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
61. 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: ICCV. pp. 4349–4359 (2025)

A Implementation and Architecture Details

Gaussian Activations. As summarized in Table 7, the output activations of different attribute heads in the Gaussian decoder are selected according to the statistical distributions of their targets. For the color branch, which is supervised by ground-truth values in $[0, 1]$, we adopt a linear activation to prevent gradient saturation. Consequently, no explicit sigmoid is required, as the regression loss naturally enforces valid-range predictions.

Table 7: Output activations and channels of different attribute heads in Gaussian decoder.

Attribute	Activation	Channels
Depth	$y = 2\text{Softplus}(x; \beta = 2\ln 2)$	1
Opacity	$y = \text{Sigmoid}(x - 2)$	1
Color	$y = 0.25x$	3
Scale	$y = \text{Softplus}(x; \beta = \ln 2)$	3
Rotation	$y = \frac{x}{\|x\|_2}$	4

Training Loss. The joint training of our Gaussian representation is guided by a multi-term loss function, where each component emphasizes a distinct aspect of rendering quality. The overall objective is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{Gaussian}}(\mathcal{I}, \widehat{\mathcal{I}}) = \sum_{i=1}^{N_{\text{src}}+N_{\text{tar}}} \Big[& w_1 \,\text{MSE}(I_i, \widehat{I}_i) \\ & + w_2 \left(1 - \text{SSIM}(I_i, \widehat{I}_i)\right) \\ & + w_3 \,\text{LPIPS}(I_i, \widehat{I}_i)\Big]. \end{aligned} \tag{3}$$

This objective combines the Mean Squared Error (MSE) for pixel-level fidelity, the Structural Similarity Index Measure (SSIM) [48] to preserve structural consistency, and the Learned Perceptual Image Patch Similarity (LPIPS) [57] to improve perceptual realism.

During training, the ground-truth set $\mathcal{I} = \{I_i\}_{i=1}^{N_{\text{src}}+N_{\text{tar}}}$ and the rendered set $\widehat{\mathcal{I}} = \{\widehat{I}_i\}_{i=1}^{N_{\text{src}}+N_{\text{tar}}}$ include both N_{src} source views and N_{tar} target views. For evaluation, all quantitative metrics are computed exclusively on the target views.

Solution for Camera Alignment. We formulate the **Camera Alignment** as two optimization problems:

$$\Delta q^* = \arg\max_{\|\Delta q\|=1} \sum_{i=1}^{n} \text{Re}\big(\Delta q \, q_{\text{s},i}^A \, (q_{\text{s},i}^B)^{-1}\big)\,, \tag{4}$$

$$(\lambda^*, \Delta\tau^*) = \arg\min_{\lambda, \Delta\tau} \sum_{i=1}^{n} \left\|\lambda \, \tau_{\text{s},i}^A + \Delta\tau - \tau_{\text{s},i}^B\right\|_2^2. \tag{5}$$

Table 8: Comparison for larger-scale reconstruction on DL3DV [28]**.** The best and second-best results of methods without any camera parameters are highlighted in **bold** and underlined, respectively.

Method	4 Source Views			6 Source Views			12 Source Views			18 Source Views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AnySplat [19]	16.852	0.446	0.310	18.125	0.496	0.273	17.887	0.487	0.300	18.920	0.539	0.269
Depth Anything 3 [27]	16.780	0.443	0.391	17.126	0.455	0.370	14.771	0.364	0.494	14.290	0.368	0.554
StructSplat (Ours)	**20.171**	**0.692**	**0.231**	**22.358**	**0.749**	**0.187**	**22.173**	**0.731**	**0.205**	**22.396**	**0.752**	**0.196**

For rotation, the solution derived via the Lagrange multiplier method is

$$\Delta q^* = \frac{\sum_{i=1}^{n} q_{\mathrm{s},i}^{B} \left(q_{\mathrm{s},i}^{A}\right)^{-1}}{\left\| \sum_{i=1}^{n} q_{\mathrm{s},i}^{B} \left(q_{\mathrm{s},i}^{A}\right)^{-1} \right\|}. \tag{6}$$

For translation, setting the gradient to zero yields

$$\lambda^* = \frac{\sum_{i=1}^{n} \tau_{\mathrm{s},i}^{A} \cdot \tau_{\mathrm{s},i}^{B} - \frac{1}{n} \sum_{i=1}^{n} \tau_{\mathrm{s},i}^{A} \cdot \sum_{i=1}^{n} \tau_{\mathrm{s},i}^{B}}{\sum_{i=1}^{n} \left\| \tau_{\mathrm{s},i}^{A} \right\|_2^2 - \frac{1}{n} \left\| \sum_{i=1}^{n} \tau_{\mathrm{s},i}^{A} \right\|^2}, \tag{7}$$

$$\Delta\tau^* = \frac{\sum_{i=1}^{n} \tau_{\mathrm{s},i}^{B} - \lambda^* \sum_{i=1}^{n} \tau_{\mathrm{s},i}^{A}}{n}. \tag{8}$$

Computation Details. Our StructSplat is trained using a single H100 GPU for two-views setting. The total training time is approximately 2.5 days, with a peak memory usage of 73 GB. In terms of model complexity and efficiency, our model contains 1.535 B parameters and requires 164.657 ms per inference pass.

Table 9: Comparison for high resolutions on DL3DV [28]**.** The best and second-best results of methods without any camera parameters are highlighted in **bold** and underlined, respectively.

Method	448×448 Resolution			504×504 Resolution		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AnySplat [19]	20.870	0.640	0.218	-	-	-
Depth Anything 3 [27]	20.169	0.598	0.272	19.745	0.575	0.295
StructSplat (Ours)	**27.073**	**0.859**	**0.126**	**26.886**	**0.853**	**0.137**

B Additional Experiments

Larger-Scale Representation. To evaluate the scalability of our approach, we further reduce the sampling rate of the source views, thereby covering a larger scene with fewer input views. The novel-view-synthesis results for this challenging setup are reported in Table 8. Across all three configurations (4, 6, 12, and

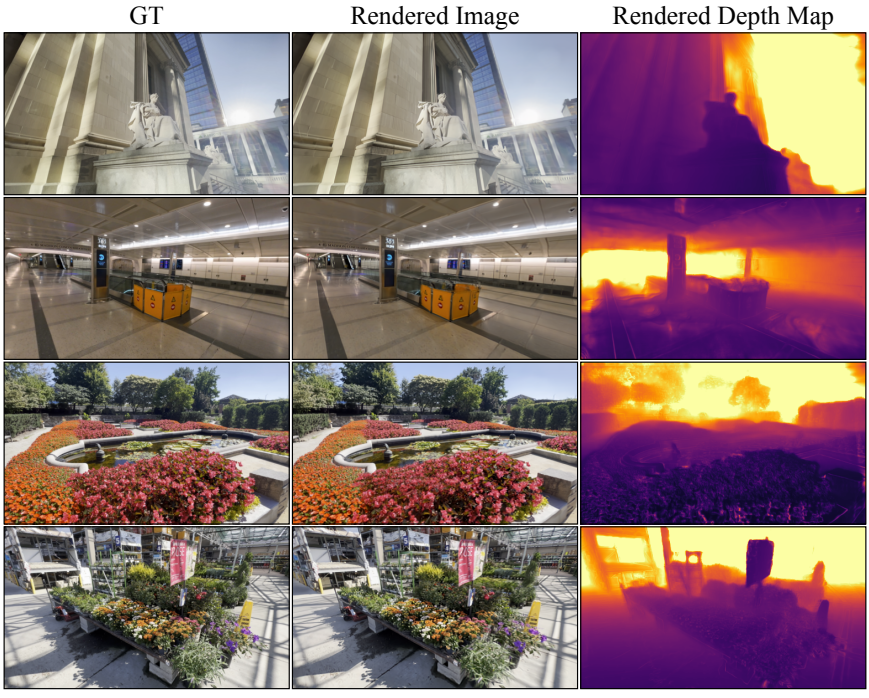

Fig. 8: Qualitative visualization of rendered novel views and depth maps. From left to right: Ground-truth images, RGB images rendered by our method, and the corresponding rendered depth maps. Although our model is trained exclusively with 2D photometric and perceptual supervision from RGB images without any ground-truth depth data or explicit depth consistency losses, it successfully captures the underlying 3D geometry, yielding highly accurate and structurally consistent depth maps across diverse scenes.

18 source views), our method consistently outperforms the baselines. Specifically, StructSplat achieves PSNR scores of 20.171 dB, 22.358 dB, 22.173 dB and 22.396 dB, with improvements of +3.319 dB, +4.233 dB, +4.286 dB and +3.476 dB over the second-best method, AnySplat [19]. Furthermore, our approach achieves the best performance in terms of both SSIM and LPIPS. These results highlight StructSplat's strong robustness and its ability to reconstruct larger-scale scenes from sparse observations.

High Resolution Evaluation. We conduct the quantitative comparison at the default resolutions of AnySplat [19] (at 448×448) and Depth Anything 3 [27] (at 504×504). As shown in Tab. 9, StructSplat achieves the best performance across both high-resolution settings.

Visualization of Rendered Depth. In Figure 8, we visualize the depth maps rendered from the 3D Gaussians predicted by our method. Notably, our model achieves precise depth estimation without relying on any ground-truth depth data or explicit depth consistency losses during training. Driven solely by RGB

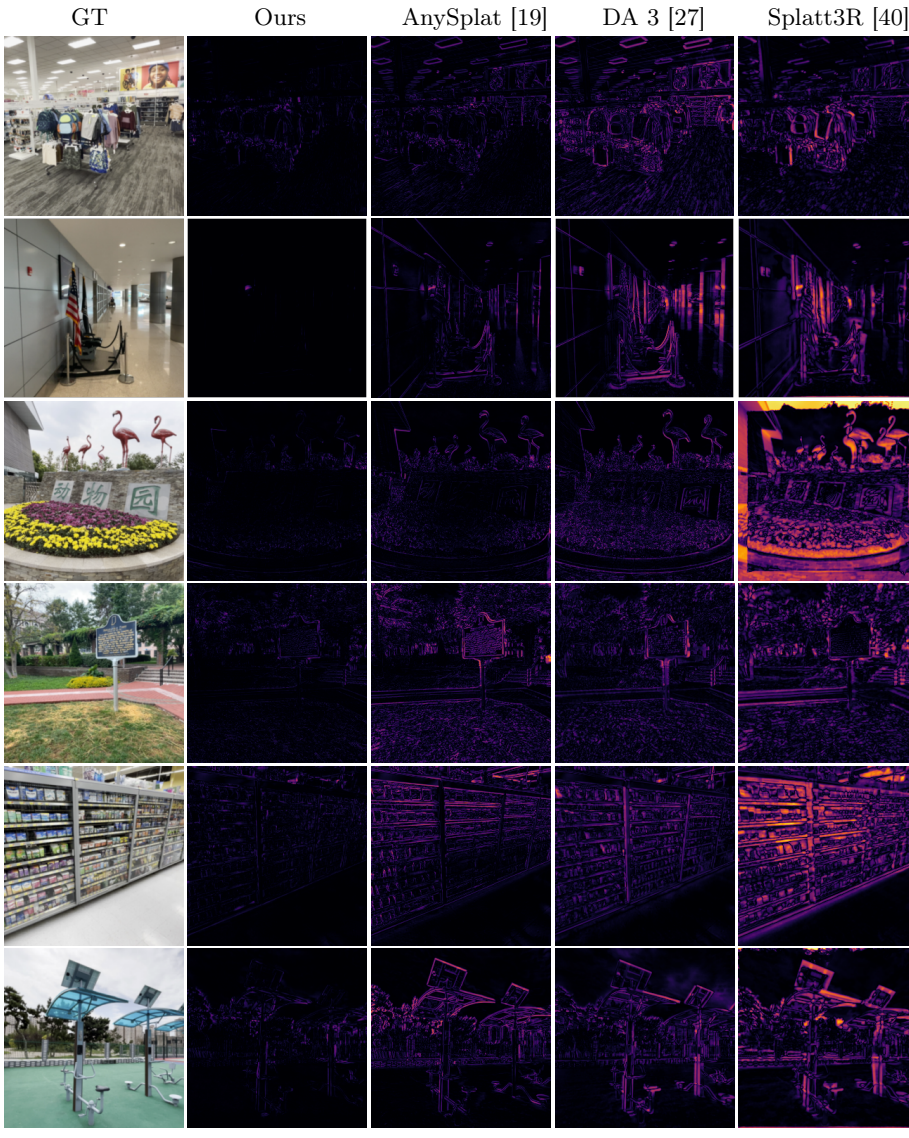

Fig. 9: Qualitative comparison of L1 error maps. We visualize the per-pixel L1 error between rendered images and the ground truth (GT). Darker regions indicate lower reconstruction errors. Compared to baseline methods (AnySplat [19], Depth Anything 3 [27], and Splatt3R [40]), our StructSplat consistently exhibits significantly lower error across all scenes, demonstrating its superior capability in preserving texture details and maintaining high-fidelity reconstruction.

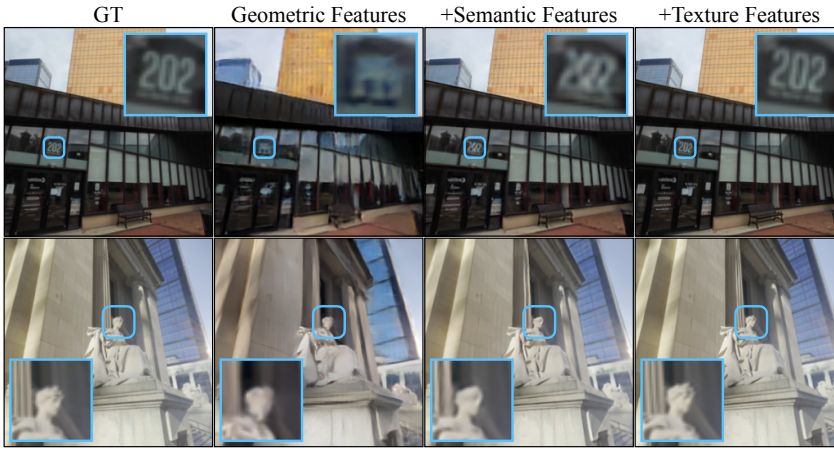

Fig. 10: Qualitative comparison of ablation. 1) Using only geometric features causes severe structural collapse and large color artifacts. 2) Adding semantic features restores coherent structures and object layouts, while 3) further introducing texture features enables sharper images with accurate high-frequency details.

supervision, our proposed Structured Representation intrinsically captures the underlying 3D geometry, producing accurate and structurally consistent depth maps.

Visualization of L1 Error Map. Since visual differences between the rendered RGB images of different methods can sometimes be subtle, we additionally visualize the corresponding L1 error maps to explicitly highlight the performance gaps, as shown in Figure 9. The error maps clearly indicate that the novel views synthesized by StructSplat exhibit substantially lower deviations from the ground truth. This reduction in error demonstrates our method's ability to recover fine texture details and preserve high-fidelity structural consistency across the scene.

Discussion on the Key Components. Quantitative results in Tab. 3 verify the effectiveness of our key components. We further conduct qualitative visual comparisons in Fig 10. 1) Using only *geometric features* causes severe structural collapse and large color artifacts. 2) Adding *semantic features* restores coherent structures and object layouts, while 3) further introducing *texture features* enables sharper images with accurate high-frequency details.

Regarding our choice and design of encoders: 1) we choose the VGGT [45] backbone as the geometry encoder, and the Depth Anything 3 [27] backbone is also compatible with our framework. 2) VGGT [45] fine-tunes DINOv2 for single-view geometric pre-extraction, resulting in a loss of appearance details in the geometry-oriented features. By introducing a separate frozen DINOv3 [39] encoder, we provide uncorrupted semantic priors. 3) We design a texture encoder to decouple Gaussian attributes (depth, opacity, color, scale, and rotation) to reduce interference among them.