SPLATANYTHING: GAUSSIAN SPACING NORMALIZATION FOR FEED-FORWARD 3DGS ACROSS ARBITRARY SCENES

Anonymous authors
Paper under double-blind review



Figure 1: Given any number of input images and optional camera poses, **SplatAnything** reconstructs a 3D Gaussian Splatting (3DGS) representation within seconds. It generalizes across arbitrary scene scales and camera configurations, while supporting challenging extrapolated viewpoints. Compared with prior feed-forward 3DGS methods, SplatAnything achieves substantially stronger cross-scene generalization and novel-view rendering quality.

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

ABSTRACT

We identify a key limitation of existing feed-forward 3DGS methods: Gaussian attributes are typically predicted in a *radial-distance-normalized* space that canonicalizes a global radial statistic of the reconstructed geometry, while the spacing between neighboring Gaussian centers can still vary substantially across scene scales and camera configurations. This mismatch induces systematic shifts in Gaussian-scale targets, leading to ambiguous supervision. To address this, we propose **SplatAnything** with *Gaussian Spacing Normalization (GSNorm)*, which canonicalizes a robust scene-level statistic of Gaussian spacing to provide a more consistent spatial reference for Gaussian prediction. To rigorously assess our method, we further establish a systematic benchmark spanning diverse datasets, input sparsity levels, and challenging extrapolated viewpoints. Extensive experiments demonstrate that SplatAnything achieves state-of-the-art performance, with particularly strong gains under substantial variations in scene scale and viewpoint.

1 INTRODUCTION

Feed-forward 3D Gaussian Splatting (3DGS) directly predicts a 3D Gaussian representation (Kerbl et al., 2023) from input images, avoiding the costly per-scene optimization of conventional 3DGS. This efficiency makes it attractive for large-scale 3D reconstruction and digital content creation. The ultimate goal is a single model that generalizes reliably across arbitrary scene scales, camera configurations, and target viewpoints, yet such broad generalization remains challenging.

A prevailing paradigm is to first predict per-pixel depth using a geometry foundation model (Wang et al., 2025; Lin et al., 2025; Team et al., 2026; Jiang et al., 2026; Yu et al., 2026), and then predict per-pixel Gaussian attributes based on the resulting geometry in its *radial-distance-normalized* coordinate space. *Radial Distance Normalization (RDNorm)* uses the mean radial distance of reconstructed 3D points to the reference-camera origin as a scene-level scale, and globally rescales the scene to canonicalize this statistic. However, RDNorm **does not** directly canonicalize the spacing between neighboring Gaussian centers, which we refer to as *Gaussian spacing*. Since adequate surface coverage requires the Gaussian scale to match the local Gaussian spacing, variations in this quantity directly induce shifts in the corresponding learning targets.

As illustrated in Fig. 2, visually similar regions can exhibit substantially different Gaussian spacings after RDNorm under different scene scales and camera configurations. Consequently, similar local observations may correspond to different Gaussian-scale targets across scenes, introducing ambiguity in scale supervision. A predictor biased toward smaller Gaussian scales may under-cover regions with large Gaussian spacing and produce holes, whereas one biased toward larger scales may cause excessive overlap in densely sampled regions and lead to over-smoothing, as shown in Fig. 7. This ambiguity in Gaussian-scale supervision poses a major bottleneck to robust feed-forward 3DGS generalization across varying scene scales and camera configurations.

To address this issue, we introduce **SplatAnything**, a feed-forward 3DGS framework equipped with *Gaussian Spacing Normalization (GSNorm)*. The key idea of GSNorm is to derive the scene-level normalization factor from a representative statistic of Gaussian spacing, rather than from a global radial statistic. Under the per-pixel Gaussian prediction paradigm, where each valid image pixel corresponds to a Gaussian center, perspective projection relates Gaussian spacing to scene depth and camera intrinsics. Based on this relation, GSNorm computes Gaussian spacing over all valid observations and uses its median as a robust scene-level normalization factor. The reconstructed geometry is then rescaled accordingly, canonicalizing the representative Gaussian spacing across scenes. This provides a more consistent spatial reference for Gaussian-scale prediction, alleviating ambiguity in supervision caused by variations in scene scale and camera configuration and thereby improving generalization across diverse scenes and viewpoints.

To rigorously assess SplatAnything and prior methods, we further introduce a cross-dataset benchmark for in-the-wild evaluation. Unlike conventional protocols (Wang et al., 2026a; Xu et al., 2025b;a; Ye et al., 2026; Ren et al., 2026), which often focus on training-related domains and interpolated target views, our benchmark covers diverse datasets, scene scales, image resolutions, and camera configurations, while explicitly varying input-view sparsity and emphasizing challenging

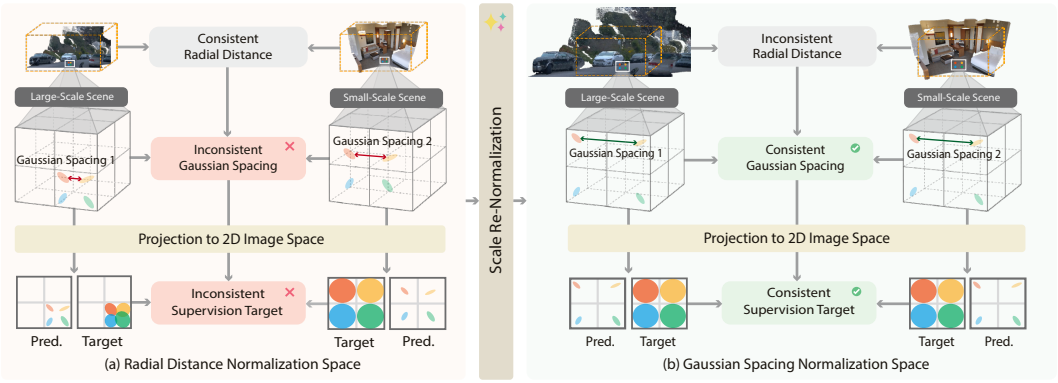

Figure 2: **Normalization space.** (a) *Radial Distance Normalization* canonicalizes a global radial statistic but leaves Gaussian spacing varying across scenes and camera configurations, resulting in inconsistent supervision targets. (b) *Gaussian Spacing Normalization* canonicalizes a representative statistic of Gaussian spacing, providing a more consistent spatial reference for Gaussian prediction.

extrapolated viewpoints. Extensive experiments show that SplatAnything achieves state-of-the-art performance, with particularly strong gains under large scene and viewpoint shifts.

In summary, our contributions are:

- We identify ambiguity in Gaussian-scale supervision across varying scene scales and camera configurations induced by RDNorm. To address this issue, we propose *Gaussian Spacing Normalization (GSNorm)*, which canonicalizes a robust scene-level statistic of Gaussian spacing to provide a more consistent spatial reference for Gaussian prediction.
- We establish a systematic cross-dataset evaluation benchmark for feed-forward 3DGS, spanning diverse scene categories, image resolutions, camera configurations, input-view sparsity levels, and challenging extrapolated viewpoints. This benchmark enables a controlled evaluation of cross-scene and cross-view generalization.

2 Related Work

Geometry foundation models. Recent feed-forward geometry foundation models (Wang et al., 2024; 2025; Lin et al., 2025; Liu et al., 2025; Jiang et al., 2026) directly recover depth, point maps, and camera poses from RGB observations, providing strong geometric priors across diverse scenes. These models typically employ radial distance normalization to keep reconstructed geometry within a numerically stable range. While effective for geometry prediction, such normalization does not ensure consistent *Gaussian spacing* across scenes, cameras, and image resolutions. This becomes critical when the normalized geometry is reused for per-pixel Gaussian prediction, since Gaussian scales are defined in the same 3D coordinate system.

Feed-forward 3D Gaussian Splatting. Feed-forward 3DGS methods directly infer Gaussian representations from image observations, often by first estimating per-pixel depth and then predicting Gaussian attributes. pixelSplat (Charatan et al., 2024) reconstructs 3D Gaussians from image pairs by learning a 3D probability distribution over Gaussian centers. DepthSplat (Xu et al., 2025b) combines monocular depth priors with multi-view matching, while ReSplat (Xu et al., 2025a) further refines predicted Gaussians through rendering-based optimization. However, limited geometric accuracy can become a bottleneck under large viewpoint extrapolation. More recent methods (Jiang et al., 2025; Lin et al., 2025; Team et al., 2026; Anonymous, 2027) leverage strong geometry foundation models to initialize Gaussian centers and predict the remaining attributes. Although this substantially improves reconstruction quality, these methods still inherit the radial-distance-normalized space of their geometry backbones, in which Gaussian spacing can vary substantially across scene scales and camera configurations, leading to ambiguity in Gaussian-scale supervision. Such supervision ambiguity makes it difficult to learn a single Gaussian predictor that generalizes reliably across diverse input settings. Other approaches (An et al., 2025; Li et al., 2026) do not rely on explicit geometric priors. While they can achieve strong rendering quality at interpolated viewpoints, the learned Gaussian geometry can become less reliable under large viewpoint extrapolation.

3 SPLATANYTHING

3.1 FORMULATION

Given N input images $\mathcal{I} = \{I_i\}_{i=1}^N$, together with optional camera intrinsics $\mathcal{K} = \{K_i\}_{i=1}^N$ and extrinsics $\mathcal{T} = \{(R_i, t_i)\}_{i=1}^N$, our goal is to predict a 3D Gaussian scene representation

$$\mathcal{G} = \{(\mu_j, \alpha_j, \Sigma_j, \mathrm{sh}_j)\}_{j=1}^M, \tag{1}$$

where μ_j, α_j, Σ_j, and sh_j denotethe Gaussian center, opacity, covariance, and SH coefficients.

3.2 PRELIMINARIES

Gaussian spacing. For per-pixel Gaussian prediction, each valid image pixel defines a Gaussian center at its back-projected 3D point. We refer to the spatial spacing between Gaussian centers induced by neighboring image pixels as *Gaussian spacing*. This quantity is particularly relevant to Gaussian-scale prediction, since adequate surface coverage requires the Gaussian scale to be commensurate with the local Gaussian spacing.

For the i-th view, let $D_i(\mathbf{u})$ denote the depth at pixel $\mathbf{u} = (u, v)$, and let $(f_{x,i}, f_{y,i})$ and $(c_{x,i}, c_{y,i})$ denote the focal lengths and principal point in pixels. Under the pinhole camera model, the back-projected point $\mathbf{P}_i^c(\mathbf{u})$ in the local camera frame is

$$\mathbf{P}_i^c(\mathbf{u}) = D_i(\mathbf{u}) \begin{bmatrix} (u - c_{x,i})/f_{x,i} \\ (v - c_{y,i})/f_{y,i} \\ 1 \end{bmatrix}. \tag{2}$$

Consider two horizontally adjacent pixels $\mathbf{u}$ and $\mathbf{u} + \mathbf{e}_x$. Under a locally constant-depth approximation, $D_i(\mathbf{u} + \mathbf{e}_x) \approx D_i(\mathbf{u})$, their horizontal Gaussian spacing is

$$d_{x,i}(\mathbf{u}) = \|\mathbf{P}_i^c(\mathbf{u} + \mathbf{e}_x) - \mathbf{P}_i^c(\mathbf{u})\|_2 \approx \frac{D_i(\mathbf{u})}{f_{x,i}}. \tag{3}$$

Similarly, the vertical spacing is $d_{y,i}(\mathbf{u}) \approx D_i(\mathbf{u})/f_{y,i}$. We summarize the two directions using their geometric mean,

$$d_i(\mathbf{u}) = \sqrt{d_{x,i}(\mathbf{u}) d_{y,i}(\mathbf{u})} \approx \frac{D_i(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}}, \tag{4}$$

which we use as a first-order scalar characterization of local Gaussian spacing.

3.3 NORMALIZATION SPACE FOR GAUSSIAN PREDICTION

Limitation of radial distance normalization. Feed-forward 3DGS methods built on geometry foundation models typically predict Gaussian attributes in the normalized coordinate space inherited from their geometry backbones. These models commonly normalize a reconstructed scene using the mean radial distance of 3D points to the reference-camera origin, which we refer to as *Radial Distance Normalization (RDNorm)*. Let $\mathbf{P}_i(\mathbf{u})$ denote the reconstructed 3D point at pixel $\mathbf{u}$ in view i, expressed in the reference-camera coordinate system before scene-level normalization, and let Ω_i denote the set of valid pixels in view i. The normalization factor s_r under RDNorm is computed as

$$s_r = \frac{1}{N_p} \sum_{i=1}^N \sum_{\mathbf{u} \in \Omega_i} \|\mathbf{P}_i(\mathbf{u})\|_2, \qquad N_p = \sum_{i=1}^N |\Omega_i|. \tag{5}$$

The geometry, depth, and camera translations are then rescaled as

$$\mathbf{P}_i^r = \frac{\mathbf{P}_i}{s_r}, \qquad D_i^r = \frac{D_i}{s_r}, \qquad \mathbf{t}_i^r = \frac{\mathbf{t}_i}{s_r}, \tag{6}$$

From Eq. 4, the Gaussian spacing after RDNorm is approximately

$$d_i^r(\mathbf{u}) \approx \frac{D_i^r(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}} = \frac{D_i(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}\, s_r}. \tag{7}$$

RDNorm canonicalizes a scene-level radial statistic of the reconstructed geometry, but does not directly canonicalize a scene-level statistic of Gaussian spacing.

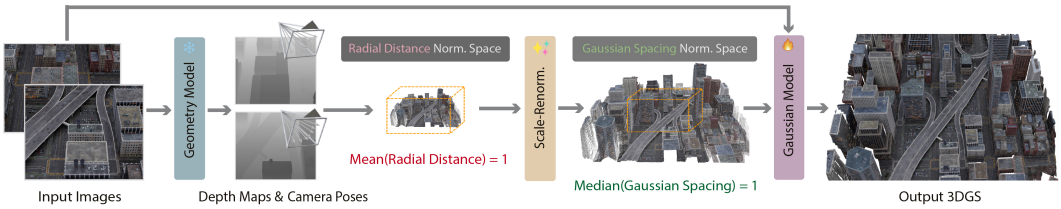

Figure 3: **Architecture of SplatAnything.** Given N input images and optional camera parameters, a frozen geometry foundation model first estimates multi-view geometry. We convert the geometry from its native radial-distance-normalized space to the proposed Gaussian-spacing-normalized space and predict per-pixel Gaussian attribute through a Gaussian backbone and a lightweight GShead.

Consequently, the spacing between neighboring Gaussian centers can still vary substantially across different scene scales and camera configurations, as shown in Fig. 4. This variation is particularly important for Gaussian prediction because the Gaussian scale required for adequate surface coverage should be commensurate with the local Gaussian spacing. Therefore, similar local observations can correspond to different Gaussian-scale targets after RDNorm, introducing ambiguity in scale supervision across scenes. This motivates us to define the normalization space directly from a representative statistic of Gaussian spacing.

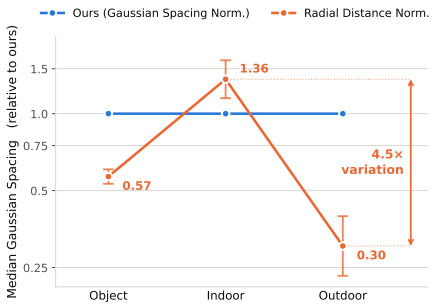

Figure 4: **Median Gaussian spacing across scene types.** RDNorm leaves up to $4.5\times$ variation in median Gaussian spacing across scene categories, whereas GSNorm canonicalizes it by construction.

Gaussian spacing normalization. We propose *Gaussian Spacing Normalization (GSNorm)*, which defines the coordinate scale for Gaussian prediction using Gaussian spacing rather than a global radial statistic. Because the reconstruction is rescaled by a single scene-level factor, we use the median characteristic Gaussian spacing as a robust representative scale. Specifically, for a generic scene-level scale $s > 0$, the characteristic Gaussian spacing becomes

$$d_i^s(\mathbf{u}) \approx \frac{D_i(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}\, s}. \tag{8}$$

GSNorm chooses s such that the scene-level median Gaussian spacing is canonicalized:

$$\underset{i,\mathbf{u}\in\Omega_i}{\text{median}}\left(d_i^s(\mathbf{u})\right) \approx \underset{i,\mathbf{u}\in\Omega_i}{\text{median}}\left(\frac{D_i(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}\, s}\right) = 1. \tag{9}$$

Since the median is equivariant to positive scaling, Eq. 9 directly yields the normalization factor s_g:

$$s_g = \underset{i,\mathbf{u}\in\Omega_i}{\text{median}}\left(\frac{D_i(\mathbf{u})}{\sqrt{f_{x,i} f_{y,i}}}\right). \tag{10}$$

Using s_g, we rescale the geometry, depth, and camera translations as

$$\mathbf{P}_i^g = \frac{\mathbf{P}_i}{s_g}, \qquad D_i^g = \frac{D_i}{s_g}, \qquad \mathbf{t}_i^g = \frac{\mathbf{t}_i}{s_g}. \tag{11}$$

GSNorm canonicalizes the scene-level median Gaussian spacing while preserving the relative variation of local Gaussian spacing. This aligns the representative Gaussian spacing scale across scenes and camera configurations without suppressing meaningful local geometric variation. Compared with RDNorm, GSNorm therefore provides a more consistent scale reference for Gaussian prediction that is directly related to Gaussian surface coverage. This alleviates ambiguity in Gaussian-scale supervision across different input settings, facilitating the learning of a feed-forward Gaussian model with robust cross-scene and cross-view generalization.

3.4 SPLATANYTHING ARCHITECTURE

The overall architecture is illustrated in Fig. 3. More details are provided in the appendix.

Geometry prediction and rescaling. Given N input images and optional camera parameters, we employ the frozen WorldMirror 2.0 (Team et al., 2026) geometry model to estimate multi-view depths, intrinsics, and camera poses. The predicted depths and camera translations are initially represented in the radial-distance-normalized coordinate space of the geometry model. We then use GSNorm (Eq. 11) to rescale these scale-dependent quantities into the proposed Gaussian-spacing-normalized space, obtaining $D^g = \{D_i^g\}_{i=1}^N$ and $T^g = \{(R_i, t_i^g)\}_{i=1}^N$. We unproject the rescaled depth maps using the corresponding camera parameters to obtain $\mathbf{P}^g = \{\mathbf{P}_i^g\}_{i=1}^N$, where each valid per-pixel 3D point serves as the center of a Gaussian primitive.

Gaussian attribute prediction. We initialize a *Gaussian Backbone* from the geometry backbone of WorldMirror 2.0 and fine-tune it for Gaussian prediction. A lightweight *GSHead*, trained from scratch, predicts per-pixel Gaussian attributes for all input views:

$$\{\mathcal{G}_i^g\}_{i=1}^N = \mathrm{GSHead}\left(\mathrm{GaussianBackbone}(\mathcal{I}, \mathit{Optional}(\mathcal{K}, \mathcal{T}))\right). \tag{12}$$

For each pixel, the network predicts opacity, rotation, scale, and spherical-harmonic appearance coefficients. Together with the corresponding center from $\mathbf{P}^p$, these quantities define the per-pixel Gaussian primitives in the Gaussian-spacing-normalized coordinate space.

3.5 TRAINING SPLATANYTHING

The training datasets and view-sampling strategies are detailed in the appendix.

Given a rendered target-view image $\hat{I}_t$ and its ground-truth image I_t, we supervise the model using a pixel-wise MSE loss and an LPIPS perceptual loss:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{3HW}\left\|\hat{I}_t - I_t\right\|_2^2, \qquad \mathcal{L}_{\mathrm{LPIPS}} = \mathrm{LPIPS}\left(\hat{I}_t, I_t\right). \tag{13}$$

The overall objective is

$$\mathcal{L} = \lambda_{\mathrm{MSE}}\mathcal{L}_{\mathrm{MSE}} + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}}, \tag{14}$$

where $\lambda_{\mathrm{MSE}} = \lambda_{\mathrm{LPIPS}} = 1$ in our implementation.

SplatAnything is trained for 20,000 iterations on 64 NVIDIA H20 GPUs, using 4 to 32 input views and dynamically varying the input resolution from 504 to 1024. We use a 500-step warm-up and learning rates of $5 \times 10^{-6}/2 \times 10^{-5}$ for the Gaussian Backbone and GSHead.

4 EXPERIMENTS

4.1 EVALUATION BENCHMARK

Existing benchmarks are typically concentrated on a small set of datasets and evaluate interpolated target views, making it difficult to assess cross-domain and viewpoint generalization. To provide a more comprehensive evaluation, we design a new benchmark with the following properties:

Cross-domain and cross-scale coverage. We evaluate on **seven datasets**: ScanNet (Dai et al., 2017), NRGBD (Azinović et al., 2022), DL3DV (Ling et al., 2024), Tanks and Temples (Knapitsch et al., 2017), Mip-NeRF 360 (Barron et al., 2022), ETH3D (Schops et al., 2017), and DTU (Jensen et al., 2014). These datasets span a wide range of physical scene scales, providing a substantially broader test of cross-domain and cross-scale generalization.

Controlled extrapolation and observation sparsity. We sample context views from a compact trajectory span and select target views using an occlusion-aware symmetric co-visibility criterion. The input–target overlap is controlled within **0.2–0.6**, with lower overlap corresponding to more challenging extrapolation. We further evaluate **1/4/8/16/32/64 input views** under a fixed context span to isolate the effect of observation density. More details are provided in the appendix.

Table 1: Quantitative comparison across seven benchmark datasets, averaged over **4–64 input views**. The top-3 results are highlighted as first , second , and third .

	Input Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.278	0.414	0.236	0.352	0.371	0.327	0.593	0.737	0.538	0.787	0.278	0.154	0.520	0.810
	YoNoSplat	0.265	0.236	0.285	0.240	0.375	0.219	0.424	0.338	0.405	0.586	0.321	0.033	0.484	0.729
	DA3-GS	0.286	0.305	0.233	0.217	0.337	0.228	0.414	0.380	0.371	0.565	0.228	0.087	0.365	0.582
	Mirror2-GS	0.193	0.222	0.235	0.237	0.246	0.188	0.306	0.351	0.283	0.528	0.205	0.070	0.347	0.617
	QuerySplat	0.236	0.242	0.261	0.216	0.343	0.291	0.425	0.393	0.412	0.652	0.290	0.031	0.424	0.676
	Ours	0.184	0.207	0.195	0.183	0.221	0.156	0.252	0.299	0.233	0.463	0.147	0.049	0.229	0.483
w/ Pose	DepthSplat	0.288	0.278	0.323	0.272	0.329	0.191	0.397	0.353	0.561	0.656	0.445	0.068	0.370	0.598
	TokenGS	0.369	0.272	0.333	0.277	0.375	0.211	0.515	0.346	0.502	0.716	0.400	0.042	0.593	0.828
	tttLRM	0.292	0.300	0.320	0.292	0.323	0.196	0.390	0.342	0.350	0.531	0.225	0.063	0.382	0.599
	ReSplat	0.272	0.240	0.284	0.232	0.304	0.159	0.396	0.275	0.349	0.525	0.219	0.027	0.441	0.664
	DA3-GS (Pose Cond.)	0.274	0.291	0.238	0.227	0.311	0.219	0.425	0.368	0.362	0.556	0.223	0.089	0.368	0.596
	Mirror2-GS (Pose Cond.)	0.195	0.223	0.231	0.226	0.239	0.180	0.304	0.346	0.272	0.506	0.201	0.062	0.315	0.532
	Ours (Pose Cond.)	0.185	0.206	0.186	0.162	0.215	0.148	0.251	0.290	0.219	0.440	0.133	0.037	0.184	0.364

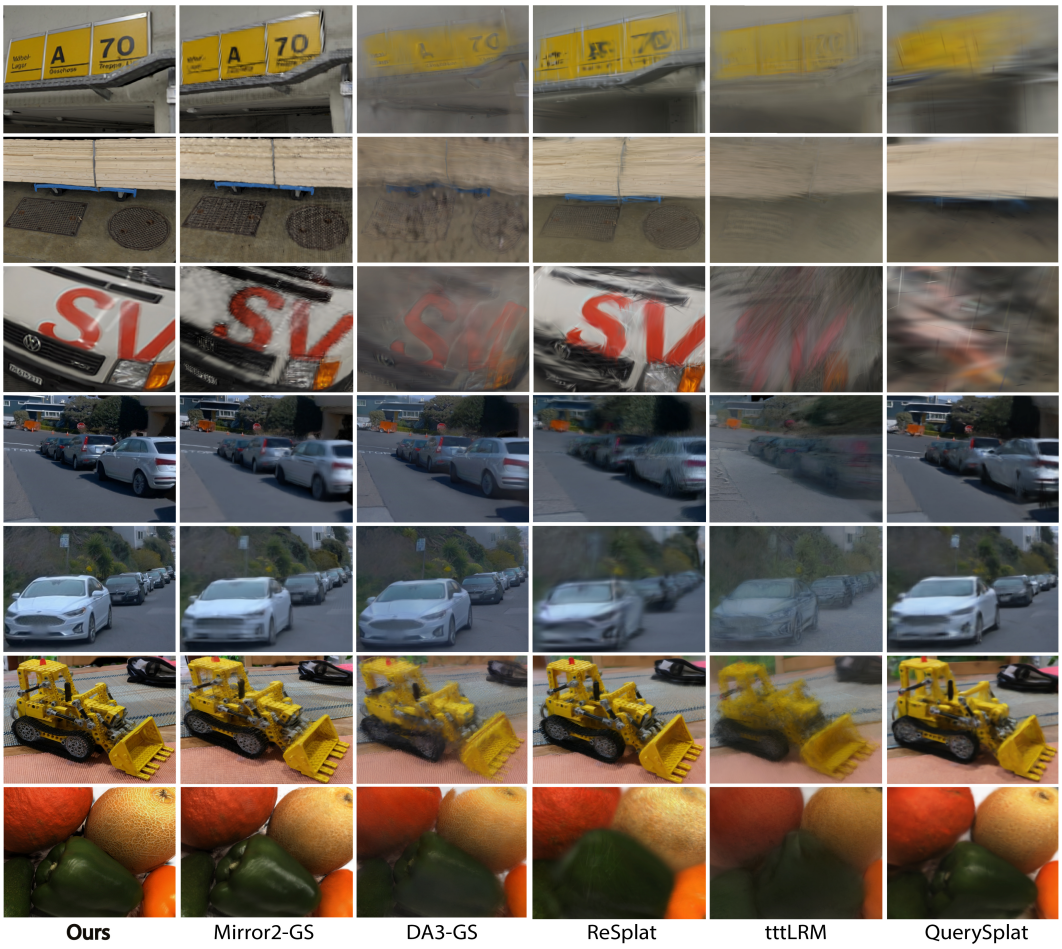

Figure 5: Qualitative comparisons across diverse scenes and viewpoints with multi-view input.

Multi-resolution. Our benchmark covers a broad resolution range, from approximately **480p to 1K**, enabling evaluation under both low- and high-resolution settings.

Compared methods. We evaluate two categories methods according to whether camera poses are provided. In the *w/o Pose* setting, methods reconstruct the scene directly from RGB observations, including AnySplat (Jiang et al., 2025), YoNoSplat (Ye et al., 2026), DA3-GS (Lin et al., 2025), Mirror2-GS (Team et al., 2026), and QuerySplat (Li et al., 2026).

In the *w/ Pose* setting, camera poses are additionally provided for model input. We compare against DepthSplat (Xu et al., 2025b), TokenGS (Ren et al., 2026), tttLRM (Wang et al., 2026a), ReSplat (Xu et al., 2025a), as well as pose-conditioned variants of DA3-GS (Lin et al., 2025) and Mirror2-GS (Team et al., 2026). Additionally, we compare against SHARP (Mescheder et al., 2026) and InfiniSplat (Wang et al., 2026b), which target for monocular view synthesis.

Table 2: Quantitative comparison using a **single input view**. The top-3 results are highlighted as first , second , and third .

Method	DL3DV		Tanks & Temples		ETH3D		DTU	
	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
SHARP	0.335	0.280	0.374	0.470	0.200	0.133	0.433	0.724
InfiniSplat	0.527	0.596	0.452	0.531	0.367	0.376	0.562	0.754
Ours	0.279	0.220	0.342	0.430	0.195	0.104	0.327	0.657

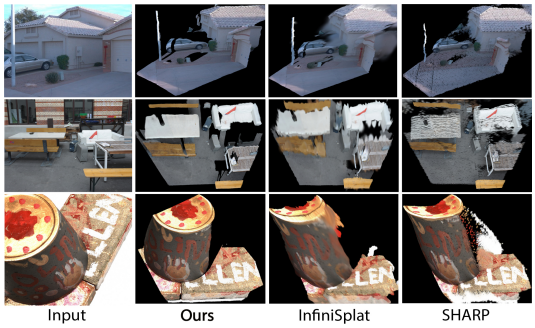

Figure 6: Qualitative comparisons across diverse scenes and viewpoints with monocular input.

4.2 Quantitative Comparisons

We report the average metrics over **4/8/16/32/64 input views** in Table 1, specific results under different input-view settings are provided in the appendix. Following SHARP (Mescheder et al., 2026), we primarily report *LPIPS* and *DISTS* in the main paper, as they better reflect rendering quality and higher-frequency details. We also report the results in the monocular setting in Table 2. Overall, SplatAnything achieves the strongest performance across most datasets and evaluation settings, demonstrating its superior generalization capabilities.

4.3 Qualitative Comparisons

Figs. 5 and 6 present qualitative comparisons across diverse scenes and extrapolated viewpoints under multi-view and monocular input settings, covering object-centric captures, indoor environments, and large-scale outdoor scenes. **SplatAnything** demonstrates a significant advantage in rendering quality: it generalizes robustly across a wide range of scene types and spatial scales, maintains geometric consistency under challenging large viewpoint extrapolation, and produces sharp, detailed renderings that faithfully recover fine local structures. More results are provided in the appendix.

4.4 Efficiency Analysis

We evaluate the inference efficiency of **SplatAnything** and other methods under different input-view settings. Detailed timing results are provided in the appendix.

4.5 Ablation Studies

Effect of Gaussian spacing normalization. We ablate the normalization space for Gaussian prediction in Table 3, comparing RDNorm, RDNorm with scale calibration, and GSNorm under the same architecture and training settings. For RDNorm with scale calibration, Gaussian prediction remains in the RDNorm space, while the predicted Gaussian scales are uniformly rescaled to match the scene's median Gaussian spacing. Both RDNorm variants remain consistently inferior to GSNorm. Qualitatively, RDNorm tends to produce holes in small object-centric scenes and over-smoothed results in large outdoor scenes (Fig. 7). Scale calibration alleviates the holes on object-centric scenes, but often turns them into over-smoothed reconstructions, while outdoor scenes remain blurry. These results indicate that post-prediction scale correction cannot resolve the ambiguity in scale supervision introduced by RDNorm. By canonicalizing the representative Gaussian spacing scale before Gaussian prediction, GSNorm provides a better-aligned prediction space and yields more complete and sharper reconstructions across diverse scenes. Additional results are provided in the appendix.

Training design choices We also analyze several architecture and training design choices, including fine-tuning the pretrained geometry backbone, fixing Gaussian centers, scaling the amount of training data, and using mixed sampling strategies for Gaussian supervision. Please refer to the appendix for more analysis and results.

Table 3: Normalization-space ablation across different numbers of input views.

Norm. Space	Num. Views=1		Num. Views=4		Num. Views=8		Num. Views=16		Num. Views=32		Num. Views=64	
	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
Radial Distance Norm.	0.308	0.394	0.223	0.278	0.223	0.272	0.218	0.268	0.225	0.277	0.224	0.250
Radial Distance Norm + Scale Calibration.	0.323	0.400	0.249	0.309	0.252	0.308	0.245	0.300	0.253	0.305	0.234	0.265
Gaussian Spacing Norm. (Ours)	**0.294**	**0.367**	**0.198**	**0.245**	**0.196**	**0.237**	**0.190**	**0.232**	**0.197**	**0.241**	**0.217**	**0.229**

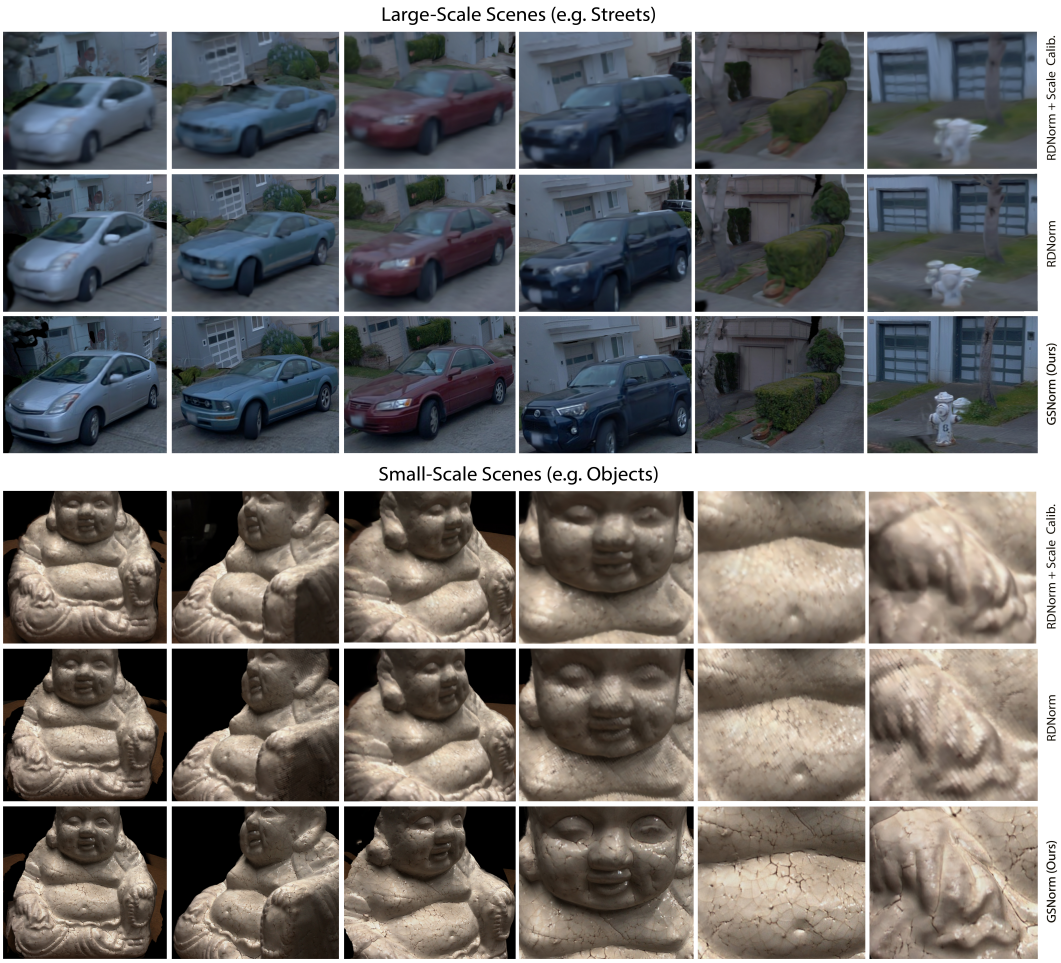

Figure 7: Ablation studies on the effect of GSNorm across different scene types.

5 Conclusion

We identified a key limitation of existing methods: *Radial Distance Normalization (RDNorm)* does not directly normalize Gaussian spacing, leading to ambiguity in Gaussian-scale supervision across scene scales and camera configurations. To address this issue, we introduced **SplatAnything** with *Gaussian Spacing Normalization (GSNorm)*, which derives a scene-level normalization factor from a representative statistic of Gaussian spacing and provides a more consistent spatial reference for Gaussian prediction. We further established a systematic cross-dataset benchmark covering different input sparsity levels and challenging extrapolated viewpoints. Experiments demonstrate that SplatAnything achieves the strongest performance across most evaluation settings, highlighting the effectiveness of GSNorm and marking a substantial step toward feed-forward 3DGS.

Limitations and Future Work. SplatAnything relies on geometric priors from a geometry foundation model and may degrade when reliable geometry cannot be recovered from extremely sparse or challenging inputs. It also primarily reconstructs regions visible in the input views, which can leave holes in unobserved areas. Future work will explore more robust geometry estimation and generative priors for completing unobserved regions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing, language polishing, and code development. In particular, generative AI was used to improve the clarity, grammar, and presentation of the manuscript, and to assist with parts of code implementation and debugging under the authors' direction. All AI-assisted text was carefully reviewed and revised by the authors, and all AI-assisted code was manually inspected, tested, and verified for correctness. Generative AI was not used to independently originate the research ideas, formulate the research questions, or design the core technical contributions of this work. We also did not use generative AI tools to generate experimental or training data, or to independently search for, select, or determine the references used in the paper. The authors take full responsibility for the final content of this work, including all text, technical claims, experimental results, and code produced with the assistance of generative AI.

REFERENCES

Honggyu An, Jaewoo Jung, Mungyeom Kim, Chaehyun Kim, Minkyeong Jeon, Jisang Han, Kazumi Fukuda, Takuya Narihira, Hyuna Ko, Junsu Kim, et al. C3g: Learning compact 3d representations with 2k gaussians. *arXiv preprint arXiv:2512.04021*, 2025.

Anonymous. Infinisplat v2: Analyzing and improving feed-forward 3dgs from any views, 2027. Manuscript under review at ICLR 2027.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pp. 690–708. Springer, 2022.

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6280–6291. IEEE, 2022.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022.

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

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

Yanqin Jiang, Tengfei Wang, Zhengwei Wang, Chenjie Cao, Junta Wu, Wenhan Luo, Weiming Hu, Jin Gao, and Chunchao Guo. Omnix: Any-view and any-time 4d reconstruction via feed-forward trajectory fields. In *European Conference on Computer Vision*, pp. 313–331. Springer, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Yinglong Li, Donghui Shen, Xiaoyu Zhang, Zhichao Ye, Hongyu Wu, Aimin Hao, Guofeng Zhang, and Haomin Liu. Querysplat: Decoupling geometry and appearance representations in 3dgs prediction. *arXiv preprint arXiv:2608.01186*, 2026.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3182–3192. IEEE, 2023.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025.

Lars Mescheder, Wei Dong, Shiwei Li, Xuyang Bai, Marcel Santos, Peiyun Hu, Bruno Lecouat, Mingmin Zhen, Amaël Delaunoy, Tian Fang, et al. Sharp monocular view synthesis in less than a second. In *International Conference on Learning Representations*, volume 2026, pp. 24192–24230, 2026.

Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20524–20531. IEEE, 2025.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. Tokengs: Decoupling 3d gaussian prediction from pixels with learnable tokens. *arXiv preprint arXiv:2604.15239*, 2026.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10892–10902. IEEE, 2021.

Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

HY Team, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv e-prints*, pp. arXiv–2604, 2026.

Rawmantic Team. Rawmantic 3d training field: From the real world, for the real world. *Rawmantic*, 2026.

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8938–8948. IEEE, 2021.

Chen Wang, Hao Tan, Wang Yifan, Zhiqin Chen, Yuheng Liu, Kalyan Sunkavalli, Sai Bi, Lingjie Liu, and Yiwei Hu. tttlrm: Test-time training for long context and autoregressive 3d reconstruction. *arXiv preprint arXiv:2602.20160*, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Jiawei Wang, Hao Yu, Yongzhen Hu, Xinyi Yang, Tao Ni, Xin Zhan, Junbo Chen, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. Infinisplat: Implicit gaussian decoding for large-baseline monocular view synthesis. *arXiv preprint arXiv:2608.02437*, 2026b.

Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Feng Chen, Jia-Wang Bian, Zheng Zhu, et al. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2026c.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814. IEEE, 2023.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Resplat: Learning recurrent gaussian splatting. *arXiv preprint arXiv:2510.08575*, 2025a.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463. IEEE, 2025b.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1787–1796. IEEE, 2020.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 39852–39871, 2026.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22. IEEE, 2023.

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. Infinidepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. *arXiv preprint arXiv:2601.03252*, 2026.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025.

A Appendix

A.1 Training Data.

The training data used for **SplatAnything** is listed in Table 4.

Table 4: Details of the training datasets used by SplatAnything.

Data Type	Dataset	# Scenes	Resolution ($H \times W$)
Synthetic	HyperSim (Roberts et al., 2021)	364	768×1024
	UnrealStereo4K (Tosi et al., 2021)	18	1080×1920
	OmniObject3D (Wu et al., 2023)	5914	800×800
	Rawmantic3D (Team, 2026)	30	1080×1920
	OmniWorld (Zhou et al., 2025)	10464	720×1280
	GTA-SfM (Wang & Shen, 2020)	200	480×640
	MatrixCity (Li et al., 2023)	8	1080×1920
	TartanGround (Patel et al., 2025)	635	640×640
Real-world	ScanNet++ (Yeshwanth et al., 2023)	936	696×1035
	Matterport3D (Chang et al., 2017)	90	1024×1280
	ARKitScenes (Baruch et al., 2021)	2501	480×640
	Taskonomy (Zamir et al., 2018)	537	512×512
	MapFree (Arnold et al., 2022)	460	720×540
	BlendedMVS (Yao et al., 2020)	502	384×504

A.2 Data Construction Strategies.

We design multiple strategies to construct the training pairs for **SplatAnything**.

Source-View Sampling Strategy. We construct source views to cover diverse camera-motion patterns while maintaining sufficient geometric overlap for reliable reconstruction by the geometry foundation model. We consider four representative trajectory patterns: *forward-traversal*, where the camera moves primarily along the viewing direction; *lateral-sweep*, where the camera translates mainly sideways while maintaining a similar viewing direction; *outward-looking*, where the camera remains approximately stationary and rotates to observe different directions; and *inward-orbit*, where the camera moves around a local region while maintaining an inward-facing view toward that region. For trajectories with substantial translation, we further vary the sampling stride to produce either densely overlapping or wide-baseline source views, covering different levels of parallax and input sparsity. During training, the trajectory pattern and sampling stride are randomly selected from a dataset-specific mixture, with distance and stride budgets adapted to each dataset. Validation uses a fixed deterministic sampling configuration for reproducibility. Examples of the different source-view sampling patterns are shown in Fig. 8.

Sampling strategy for target views. Our target-view sampling aims to provide supervision across diverse viewpoint changes, ranging from interpolation to challenging extrapolation, in order to improve the viewpoint generalization ability of our model. For *interpolation views*, we select a held-out view between the input views; for *extrapolation views*, we randomly sample diverse extrapolation patterns, including advancing along the camera trajectory to inspect local scene details at a closer range, as well as moving away from the observed trajectory to view the scene from novel positions and orientations. The visualization of different target-view sampling modes are shown in Fig. 9.

A.3 Evaluation Benchmark Details

We provide additional details of the evaluation benchmark introduced in the main paper. Our benchmark spans seven datasets and evaluates different input-view counts of 4, 8, 16, 32, and 64. The protocol is designed to independently control input-view sparsity and target-view difficulty, while ensuring identical evaluation conditions across methods.

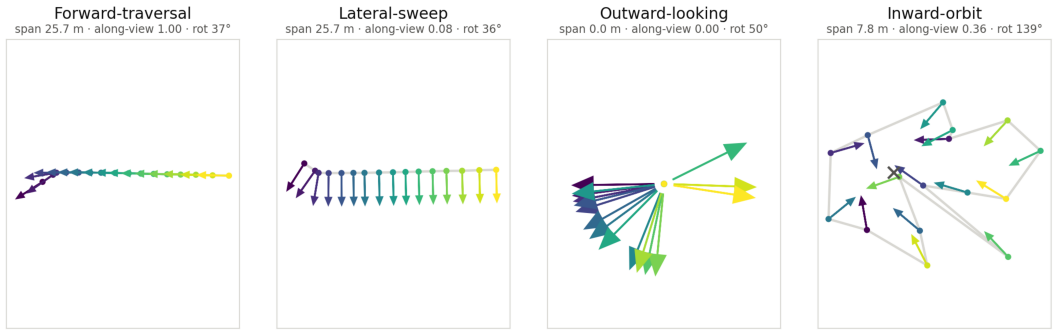

Figure 8: **Source-view trajectory patterns** (16 views per panel), visualized in an approximately top-down projection. Dots and arrows denote camera centers and viewing directions, respectively; the gray polyline connects consecutive views, and color indicates capture order. For *inward-orbit*, $\times$ marks the approximate viewing center.

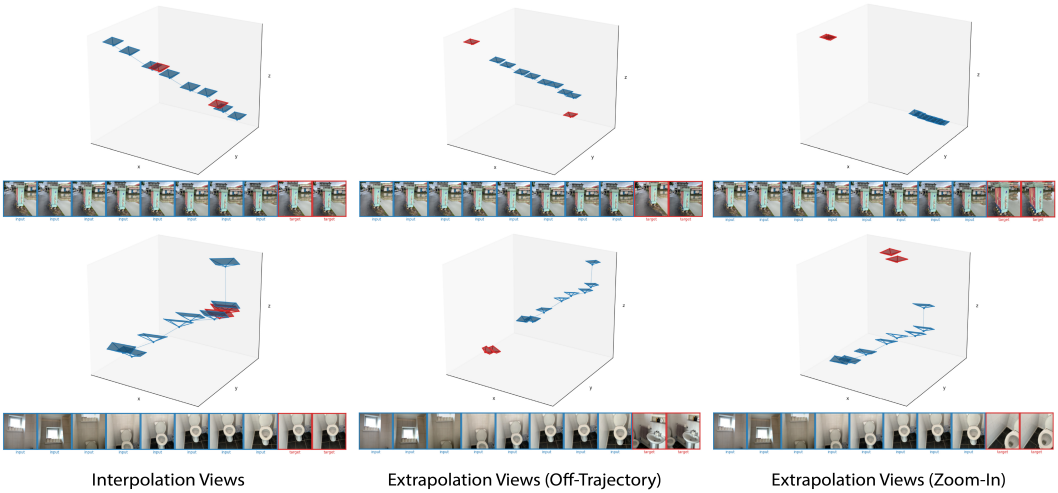

Figure 9: **Target-view sampling modes** used during training. Blue cameras denote the source views provided as input, while red cameras denote the target views used for rendering supervision.

Source-view construction. Instead of distributing the source views uniformly over the entire capture, we sample them from a compact contiguous arc of the camera trajectory. This design leaves a meaningful portion of the capture outside the observed region, making it possible to evaluate genuinely extrapolated novel views rather than views that are always bracketed by the inputs. To make different input-view counts directly comparable, we adopt a *fixed-span* strategy. For each scene, we first determine the trajectory span at the sparsest setting of 4 source views. We select the widest arc whose adjacent source-view pairs maintain sufficient geometric overlap, measured by symmetric co-visibility; specifically, the 20-th percentile of adjacent-pair co-visibility is required to be at least 0.35. The resulting span and offset are then reused for 8, 16, 32, and 64 views, with additional views inserted within the same trajectory range. Therefore, all view-count settings share the same spatial coverage, and only the observation density changes. This allows the effect of input sparsity to be evaluated without conflating it with changes in scene coverage.

Target-view selection. Target views are sampled from frames outside the source-view arc and are selected according to an occlusion-aware symmetric co-visibility measure. For each candidate target view, valid pixels are back-projected using ground-truth depth and projected into the source views. Pixels that are geometrically consistent with at least one source view are treated as observed, while occluded and out-of-frustum pixels reduce the overlap score; pixels for which visibility cannot be determined due to invalid depth are excluded from the statistic.

We compute co-visibility in both projection directions and define

$$C_{\mathrm{sym}} = \min\left(C_{\mathrm{fwd}}, C_{\mathrm{bwd}}\right). \tag{15}$$

The symmetric formulation is important for asymmetric camera motions, such as moving toward the scene, where a one-sided overlap measure can remain high even under a substantial viewpoint change. We evaluate up to 40 candidate target views per scene and retain those whose symmetric co-visibility lies in

$$0.20 \leq C_{\mathrm{sym}} < 0.60. \tag{16}$$

Up to four target views are selected from this interval, prioritizing lower co-visibility and, when necessary, larger parallax. In this way, target-view difficulty is explicitly controlled by geometric overlap rather than by temporal or camera-index distance alone, providing a consistent extrapolation regime across datasets and methods.

Evaluation without input poses. For pose-free methods, we estimate a scene-level similarity transformation between the predicted source-camera poses and their ground-truth counterparts, and apply this transformation to the predicted Gaussian representation. The aligned Gaussians are then rendered from the ground-truth target-camera poses, and the resulting images are evaluated against the corresponding target views. Ground-truth source poses are used only for this post-hoc alignment and are not provided to the model during inference. This allows pose-free and pose-conditioned methods to be evaluated on identical target viewpoints in the same ground-truth coordinate system.

A.4 Implementation Details

GS Head Architecture. The architecture of the GS Head follows WorldMirror 2.0 (Team et al., 2026). For Gaussian attribute prediction, we parameterize the Gaussian scale in log space and apply an exponential activation, followed by clamping to a maximum value of 0.3:

$$\mathbf{s} = \min\left(\exp(\hat{\mathbf{s}}),\, 0.3\right). \tag{17}$$

Gaussian rotations are represented as quaternions and normalized to unit length,

$$\mathbf{q} = \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|_2 + \epsilon}. \tag{18}$$

Opacity is activated using the sigmoid function, constraining them to $[0, 1]$. The spherical harmonic coefficients are used directly without an additional nonlinear activation.

Coordinate alignment. During training, the predicted Gaussians are rendered using the ground-truth target camera poses for supervision. However, the camera trajectory and depth predicted by the geometry model generally have an arbitrary global scale relative to the ground-truth trajectory. Directly combining the predicted depth with the ground-truth poses would therefore lead to inconsistent geometry and camera scales, resulting in misaligned rendering supervision. To resolve this scale ambiguity, we estimate a scene-level alignment factor from the predicted and ground-truth source-camera trajectories. Specifically, we compute the ratio between corresponding pairwise camera baselines and use their median as a robust global scale factor. The predicted depth maps are then rescaled by this factor and back-projected using the ground-truth source-camera poses to construct Gaussian centers in the ground-truth coordinate system. The resulting Gaussians can therefore be rendered directly using the ground-truth target-camera poses for supervision. This alignment is applied to both the pose-free and pose-conditioned variants, since the predicted camera trajectory can exhibit a global scale discrepancy with respect to the ground truth in either setting.

Voxel Merge Strategy. GSNorm canonicalizes the representative single-view point spacing across scenes and camera configurations, but it does not by itself eliminate the variation in primitive density caused by multi-view accumulation. Since each valid pixel predicts a Gaussian primitive, increasing the number of input views over the same spatial region can introduce multiple Gaussians at nearby 3D locations, making the resulting primitive density dependent on the number of overlapping observations. To reduce such redundancy, we apply a spatial voxel merging strategy following AnySplat (Jiang et al., 2025). Specifically, we predict a per-pixel weight map $w^p = \{w_i^p\}_{i=1}^N$ and merge redundant per-view Gaussians within local voxels. This produces the final unified 3D Gaussian scene representation $\mathcal{G}$. GSNorm and voxel merging therefore address two complementary sources of scale variation: GSNorm provides a consistent characteristic sampling scale for Gaussian prediction across scenes, whereas voxel merging regularizes the accumulation of primitives caused by overlapping multi-view observations. Together, they allow the model to operate robustly over varying numbers of input views.

Table 5: Quantitative comparison across seven benchmark datasets using **4 input views**. The top-3 results are highlighted as first , second , and third .

	Input Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.289	0.467	0.249	0.377	0.350	0.319	0.624	0.777	0.575	0.825	0.296	0.173	0.491	0.819
	YoNoSplat	0.253	0.231	0.272	0.216	0.366	0.213	0.427	0.340	0.398	0.573	0.325	0.036	0.484	0.753
	DA3-GS	0.291	0.322	0.243	0.219	0.320	0.222	0.441	0.417	0.388	0.582	0.249	0.090	0.386	0.662
	Mirror2-GS	0.188	0.218	0.234	0.229	0.238	0.177	0.292	0.348	0.263	0.525	0.214	0.080	0.328	0.645
	QuerySplat	0.241	0.237	0.258	0.206	0.349	0.285	0.438	0.396	0.427	0.652	0.296	0.031	0.438	0.716
	Ours	0.185	0.212	0.199	0.185	0.221	0.150	0.240	0.301	0.226	0.472	0.151	0.047	0.229	0.526
w/ Pose	DepthSplat	0.406	0.326	0.393	0.278	0.369	0.188	0.521	0.378	0.642	0.695	0.523	0.084	0.402	0.682
	TokenGS	0.408	0.287	0.352	0.279	0.375	0.202	0.534	0.354	0.516	0.708	0.516	0.049	0.651	0.871
	tttLRM	0.354	0.378	0.402	0.421	0.351	0.223	0.406	0.336	0.404	0.590	0.250	0.066	0.463	0.713
	ReSplat	0.256	0.224	0.249	0.184	0.268	0.140	0.447	0.296	0.402	0.578	0.219	0.028	0.522	0.773
	DA3-GS (Pose Cond.)	0.285	0.314	0.253	0.237	0.306	0.217	0.447	0.402	0.381	0.585	0.245	0.093	0.380	0.677
	Mirror2-GS (Pose Cond.)	0.188	0.219	0.227	0.209	0.232	0.170	0.289	0.344	0.257	0.504	0.213	0.069	0.321	0.587
	Ours (Pose Cond.)	0.185	0.210	0.189	0.161	0.216	0.142	0.238	0.291	0.217	0.448	0.134	0.031	0.209	0.432

Table 6: Quantitative comparison across seven benchmark datasets using **8 input views**. The top-3 results are highlighted as first , second , and third .

	Input Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.291	0.447	0.238	0.356	0.364	0.322	0.635	0.801	0.538	0.806	0.287	0.143	0.476	0.808
	YoNoSplat	0.262	0.234	0.280	0.226	0.374	0.219	0.423	0.343	0.406	0.592	0.330	0.027	0.453	0.727
	DA3-GS	0.302	0.326	0.237	0.219	0.329	0.226	0.442	0.409	0.393	0.591	0.241	0.075	0.365	0.604
	Mirror2-GS	0.190	0.225	0.232	0.226	0.246	0.191	0.302	0.356	0.283	0.543	0.219	0.072	0.334	0.600
	QuerySplat	0.236	0.241	0.259	0.211	0.344	0.299	0.431	0.398	0.424	0.664	0.294	0.025	0.444	0.710
	Ours	0.185	0.218	0.197	0.184	0.221	0.159	0.246	0.306	0.235	0.482	0.156	0.040	0.216	0.467
w/ Pose	DepthSplat	0.301	0.281	0.293	0.221	0.308	0.181	0.414	0.348	0.616	0.701	0.509	0.066	0.366	0.632
	TokenGS	0.387	0.277	0.330	0.268	0.378	0.208	0.528	0.355	0.517	0.734	0.382	0.041	0.606	0.863
	tttLRM	0.270	0.283	0.326	0.290	0.314	0.192	0.394	0.345	0.356	0.560	0.246	0.071	0.369	0.611
	ReSplat	0.267	0.237	0.253	0.186	0.288	0.152	0.415	0.290	0.372	0.565	0.220	0.023	0.466	0.715
	DA3-GS (Pose Cond.)	0.292	0.317	0.240	0.226	0.306	0.217	0.459	0.386	0.389	0.595	0.239	0.079	0.377	0.632
	Mirror2-GS (Pose Cond.)	0.193	0.230	0.226	0.211	0.239	0.183	0.300	0.350	0.274	0.526	0.219	0.063	0.301	0.516
	Ours (Pose Cond.)	0.186	0.217	0.186	0.162	0.217	0.152	0.245	0.294	0.221	0.457	0.145	0.030	0.170	0.349

Table 7: Quantitative comparison across seven benchmark datasets using **16 input views**. The top-3 results are highlighted as first , second , and third .

	Input Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.294	0.438	0.234	0.348	0.370	0.322	0.656	0.811	0.570	0.812	0.286	0.195	0.534	0.806
	YoNoSplat	0.268	0.238	0.287	0.242	0.381	0.222	0.429	0.342	0.400	0.576	0.313	0.029	0.503	0.729
	DA3-GS	0.298	0.328	0.227	0.211	0.341	0.228	0.464	0.407	0.392	0.582	0.209	0.092	0.393	0.601
	Mirror2-GS	0.190	0.232	0.233	0.235	0.247	0.185	0.308	0.354	0.274	0.531	0.181	0.047	0.353	0.599
	QuerySplat	0.234	0.250	0.263	0.217	0.344	0.297	0.428	0.400	0.410	0.654	0.285	0.026	0.409	0.651
	Ours	0.180	0.217	0.191	0.183	0.219	0.155	0.252	0.315	0.227	0.467	0.121	0.037	0.238	0.484
w/ Pose	DepthSplat	0.248	0.264	0.302	0.247	0.305	0.177	0.351	0.345	0.528	0.634	0.516	0.087	0.337	0.519
	TokenGS	0.381	0.281	0.329	0.276	0.377	0.207	0.535	0.354	0.514	0.728	0.354	0.035	0.562	0.805
	tttLRM	0.274	0.282	0.304	0.260	0.310	0.184	0.388	0.347	0.336	0.529	0.205	0.048	0.346	0.544
	ReSplat	0.277	0.246	0.266	0.207	0.307	0.161	0.398	0.282	0.355	0.537	0.196	0.022	0.434	0.657
	DA3-GS (Pose Cond.)	0.288	0.318	0.229	0.219	0.299	0.212	0.480	0.384	0.389	0.584	0.207	0.094	0.391	0.603
	Mirror2-GS (Pose Cond.)	0.192	0.232	0.231	0.229	0.239	0.174	0.306	0.348	0.264	0.515	0.180	0.044	0.314	0.502
	Ours (Pose Cond.)	0.181	0.214	0.185	0.164	0.212	0.143	0.251	0.304	0.212	0.447	0.114	0.031	0.172	0.318

A.5 Additional Quantitative Results

More quantitative comparisons. We report the full comparisons using **4**, **8**, **16**, **32**, and **64** input views, as shown in Table 5, Table 6, Table 7, Table 8, and Table 9.

PSNR metric results. Table 10 reports the average PSNR of different methods. However, PSNR does not always faithfully reflect perceptual rendering quality, as it can favor overly smooth predictions that achieve low pixel-wise errors while losing high-frequency details. Although ReSplat achieves the highest PSNR in several settings, its extrapolation capability remains limited. Under large viewpoint changes, the reconstructed Gaussians often produce severe artifacts, leading to no-

Table 8: Quantitative comparison across seven benchmark datasets using **32 input views**. The top-3 results are highlighted as first , second , and third .

Input	Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.300	0.430	0.232	0.342	0.380	0.338	0.655	0.811	0.618	0.833	0.241	0.106	0.577	0.805
	YoNoSplat	0.268	0.238	0.296	0.253	0.382	0.225	0.430	0.342	0.400	0.581	0.316	0.039	0.497	0.706
	DA3-GS	0.270	0.282	0.249	0.233	0.355	0.234	0.379	0.358	0.352	0.557	0.211	0.091	0.314	0.462
	Mirror2-GS	0.192	0.229	0.238	0.247	0.249	0.195	0.312	0.352	0.288	0.544	0.204	0.082	0.371	0.623
	QuerySplat	0.231	0.253	0.263	0.223	0.343	0.301	0.428	0.400	0.408	0.658	0.286	0.043	0.403	0.627
	Ours	0.181	0.205	0.199	0.182	0.223	0.154	0.253	0.300	0.231	0.460	0.161	0.071	0.234	0.455
w/ Pose	DepthSplat	0.238	0.265	0.313	0.277	0.316	0.189	0.340	0.341	0.530	0.646	0.232	0.035	0.373	0.560
	TokenGS	0.374	0.277	0.332	0.282	0.378	0.213	0.534	0.354	0.519	0.740	0.347	0.043	0.554	0.772
	tttLRM	0.281	0.286	0.292	0.250	0.316	0.190	0.386	0.346	0.334	0.525	0.199	0.066	0.349	0.528
	ReSplat	0.271	0.245	0.302	0.249	0.328	0.170	0.368	0.263	0.323	0.517	0.242	0.035	0.343	0.512
	DA3-GS (Pose Cond.)	0.254	0.259	0.255	0.244	0.336	0.235	0.397	0.354	0.340	0.537	0.202	0.088	0.324	0.473
	Mirror2-GS (Pose Cond.)	0.194	0.230	0.237	0.241	0.241	0.185	0.309	0.346	0.277	0.523	0.193	0.070	0.322	0.521
	Ours (Pose Cond.)	0.184	0.212	0.186	0.161	0.213	0.147	0.254	0.303	0.219	0.452	0.139	0.057	0.185	0.357

Table 9: Quantitative comparison across seven benchmark datasets using **64 input views**. ETH3D and DTU results are unavailable and denoted by "–". The top-3 available results are highlighted as first , second , and third .

Input	Method	ScanNet		NRGBD		DL3DV		Tanks & Temples		Mip-NeRF 360		ETH3D		DTU	
		DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
w/o Pose	AnySplat	0.215	0.287	0.227	0.337	0.393	0.333	0.397	0.483	0.388	0.658	–	–	–	–
	YoNoSplat	0.272	0.237	0.292	0.261	0.373	0.215	0.413	0.321	0.422	0.608	–	–	–	–
	DA3-GS	0.268	0.268	0.211	0.201	0.339	0.231	0.343	0.311	0.331	0.512	–	–	–	–
	Mirror2-GS	0.205	0.208	0.236	0.250	0.249	0.194	0.315	0.346	0.305	0.495	–	–	–	–
	QuerySplat	0.237	0.228	0.261	0.223	0.334	0.274	0.400	0.370	0.390	0.634	–	–	–	–
	Ours	0.190	0.182	0.187	0.183	0.222	0.161	0.271	0.271	0.248	0.434	–	–	–	–
w/ Pose	DepthSplat	0.249	0.256	0.313	0.335	0.349	0.219	0.359	0.355	0.489	0.604	–	–	–	–
	TokenGS	0.294	0.239	0.323	0.281	0.368	0.223	0.445	0.314	0.444	0.671	–	–	–	–
	tttLRM	0.280	0.270	0.276	0.238	0.323	0.193	0.375	0.336	0.322	0.451	–	–	–	–
	ReSplat	0.287	0.250	0.348	0.336	0.329	0.173	0.352	0.244	0.292	0.429	–	–	–	–
	DA3-GS (Pose Cond.)	0.253	0.247	0.213	0.208	0.306	0.213	0.342	0.312	0.313	0.481	–	–	–	–
	Mirror2-GS (Pose Cond.)	0.208	0.205	0.233	0.238	0.242	0.188	0.316	0.340	0.286	0.463	–	–	–	–
	Ours (Pose Cond.)	0.191	0.177	0.182	0.160	0.217	0.155	0.267	0.256	0.228	0.395	–	–	–	–

Table 10: Average PSNR comparison under different numbers of input views. Results are averaged across the different datasets. The top-3 results are highlighted as first , second , and third .

Input	Method	4 Views PSNR↑	8 Views PSNR↑	16 Views PSNR↑	32 Views PSNR↑	64 Views PSNR↑
w/o Pose	AnySplat	12.68	13.20	12.83	13.12	14.26
	YoNoSplat	18.23	18.64	18.73	18.34	15.79
	DA3-GS	16.32	16.94	16.73	17.17	17.26
	Mirror2-GS	17.20	17.60	17.98	17.04	17.09
	QuerySplat	17.77	18.26	18.33	17.84	16.90
	Ours	17.85	18.28	18.61	17.85	17.98
w/ Pose	DepthSplat	15.60	16.89	17.24	17.68	16.35
	TokenGS	17.65	18.42	18.77	18.33	17.95
	tttLRM	15.35	16.68	17.61	17.13	16.94
	ReSplat	19.03	19.52	19.67	19.44	18.23
	DA3-GS (Pose Cond.)	16.18	16.96	16.99	17.32	17.55
	Mirror2-GS (Pose Cond.)	17.53	18.00	18.36	17.56	17.43
	Ours (Pose Cond.)	18.39	18.82	19.12	18.39	18.50

ticeable degradation in rendering quality, as shown in Fig. 5 and Fig. 11. In contrast, our method delivers substantially higher rendering quality under extrapolated viewpoints.

A.6 Additional Qualitative Results

More qualitative comparisons. We provide more comparisons of SplatAnything with other methods under multi-view and monocular input settings. Fig. 11 and Fig. 12 demonstrate the perfor-

Table 11: Median forward-pass latency per scene (ms) of different methods under different numbers of input views. The fastest method within each pose setting is highlighted in bold.

Input	Method	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$
w/o Pose	AnySplat	250	487	1,089	2,743	7,798
	DA3-GS	709	1,513	3,783	10,950	35,904
	Mirror2-GS	**246**	**448**	**959**	**2,268**	**6,060**
	QuerySplat	2,324	3,398	6,092	14,009	39,583
	Ours	411	761	1,649	3,991	11,037
w/ Pose	DepthSplat	**158**	**287**	**541**	**1,045**	**2,087**
	TokenGS	541	1,225	3,460	11,447	41,480
	tttLRM	487	886	1,681	3,271	6,465
	ReSplat	199	376	855	2,041	5,568
	DA3-GS (Pose Cond.)	711	1,515	3,791	10,953	35,892
	Mirror2-GS (Pose Cond.)	250	454	968	2,288	6,131
	Ours (Pose Cond.)	413	764	1,651	4,007	11,045

mance of SplatAnything under multi-view and monocular input settings, respectively. SplatAnything demonstrates superior performance over other methods under large viewpoint shifts.

More qualitative results of SplatAnything. We provide more qualitative results of SplatAnything, including objects, indoor scenes, and large-scale outdoor scenes. Results are shown in Fig. 14.

A.7 Inference Efficiency

We measure the median forward-pass latency per scene for producing Gaussian attributes under different numbers of input views. As shown in Table 11, the runtime of all methods increases with the number of input views. SplatAnything remains substantially faster than computationally heavier baselines such as DA3-GS, QuerySplat, and TokenGS. All methods are infered on a single NVIDIA H20 GPU. Inference uses BF16 precision and a batch size of 1. All input images are resized to the same resolution of $448 \times 896 (H \times W)$ to match the input constraints of all methods.

A.8 Additional Ablation Studies

Normalization Space for Gaussian Prediction. The effect of Gaussian Spacing Normalization is evaluated in the main paper. Here, we provide a more comprehensive comparison of different normalization spaces, including *Radial Distance Normalization (RDNorm)*, *Camera Distance Normalization (CDNorm)*, *Depth Normalization (DNorm)*, and our proposed *Gaussian Spacing Normalization (GSNorm)*. RDNorm is described in the main paper. CDNorm uses the median distance from all camera centers to a reference camera as the scene-level normalization factor, whereas DNorm uses the median depth over all frames. Quantitative results are reported in Table 12.

CDNorm is highly sensitive to camera trajectory: a small camera baseline yields a small normalization factor and excessively large Gaussian spacing in the normalized space, whereas larger camera displacement induces stronger rescaling. DNorm is less sensitive to camera motion, but normalizes only by scene depth and therefore ignores the pixel-space focal length and image sampling that directly affect Gaussian spacing. Consequently, changes in field of view or image resolution can still induce substantial variation in Gaussian-scale targets.

To further illustrate the failure modes of CDNorm and DNorm, we examine a challenging scene with a small camera baseline and a wide field of view ($\sim 100°$). As shown in Fig. 10, both methods produce excessively large Gaussian spacing, but for different reasons: CDNorm is dominated by the small camera displacement, whereas DNorm fails to account for the spacing increase induced by the wide field of view. This shifts the Gaussian-spacing distribution away from that seen during training, causing the predicted Gaussian scales to be too small for adequate surface coverage and resulting in severe holes. In contrast, GSNorm canonicalizes a robust scene-level statistic of Gaussian spacing, providing a more consistent spatial reference across varying camera trajectories and intrinsics.

Table 12: Ablation study of different normalization spaces across different numbers of input views.

Norm. Space	Num. Views=1		Num. Views=4		Num. Views=8		Num. Views=16		Num. Views=32		Num. Views=64	
	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
Radial Distance Norm.	0.308	0.394	0.223	0.278	0.223	0.272	0.218	0.268	0.225	0.277	0.224	0.250
Camera Distance Norm.	0.793	1.027	0.217	0.284	0.216	0.280	0.214	0.280	0.221	0.291	0.224	0.260
Depth Norm.	0.307	0.413	0.224	0.284	0.225	0.278	0.218	0.273	0.224	0.282	0.224	0.254
Gaussian Spacing Norm. (Ours)	**0.294**	**0.367**	**0.198**	**0.245**	**0.196**	**0.237**	**0.190**	**0.232**	**0.197**	**0.241**	**0.217**	**0.229**

Table 13: Ablation study of the training design choices across different numbers of input views.

Methods	Num. Views=1		Num. Views=4		Num. Views=8		Num. Views=16		Num. Views=32		Num. Views=64	
	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓	DISTS↓	LPIPS↓
Ours w/o Fine-tuning Gaussian Backbone	0.2952	0.3809	0.2026	0.2503	0.2006	0.2421	0.1948	0.2379	0.2038	0.2467	0.2239	0.2387
Ours w/o Fixed Gaussian Centers	0.2989	0.4010	0.2063	0.2591	0.2040	0.2512	0.1981	0.2463	0.2079	0.2584	0.2259	0.2507
Ours w/o Data Scaling	0.2948	0.3755	0.2045	0.2499	0.2027	0.2434	0.1960	0.2389	0.2041	0.2489	0.2229	0.2385
Ours w/o View Sampling Strategy	0.2974	0.3996	0.2026	0.2536	0.2001	0.2472	0.1928	0.2392	0.1995	0.2475	0.2231	0.2448
Ours	**0.2937**	**0.3667**	**0.1981**	**0.2450**	**0.1956**	**0.2374**	**0.1895**	**0.2315**	**0.1970**	**0.2412**	**0.2170**	**0.2285**



Figure 10: Qualitative comparisons of different normalization spaces.

Training Design Choices. We further ablate four design choices: fine-tuning the pretrained geometry backbone, fixing the Gaussian centers, scaling the training data, and adopting diverse view-sampling strategies. Results are reported in Table 13. All variants follow the same training settings. The upstream WorldMirror 2.0 geometry estimator remains frozen in all experiments.

For *w/o Fine-tuning Gaussian Backbone*, the Gaussian Backbone is also frozen at its WorldMirror 2.0 initialization, and only GSHead is optimized. For *w/o Fixed Gaussian Centers*, GSHead additionally predicts a per-pixel 3D center offset, instead of fixing the centers to the unprojected geometry. For *w/o Data Scaling*, we train on Hypersim, while keeping the number of training iterations unchanged. For *w/o View Sampling Strategy*, we randomly select n views and then partition them into source views and target views, while retaining the same numbers of source and target views. We find that fixing Gaussian centers produces the largest and most consistent gain across different input-view counts. When the centers are optimized only through rendering losses, the model can reduce image-space errors by moving primitives away from the geometry prior, resulting in geometric drift and poorer generalization to extrapolated viewpoints. Fine-tuning the Gaussian Backbone adapts the pretrained geometric features to appearance and Gaussian-attribute prediction, whereas scaling the training data improves cross-domain robustness. The mixed view-sampling strategy exposes the model to diverse camera motions and viewpoint changes, leading to more reliable reconstruction under sparse inputs and large extrapolation.

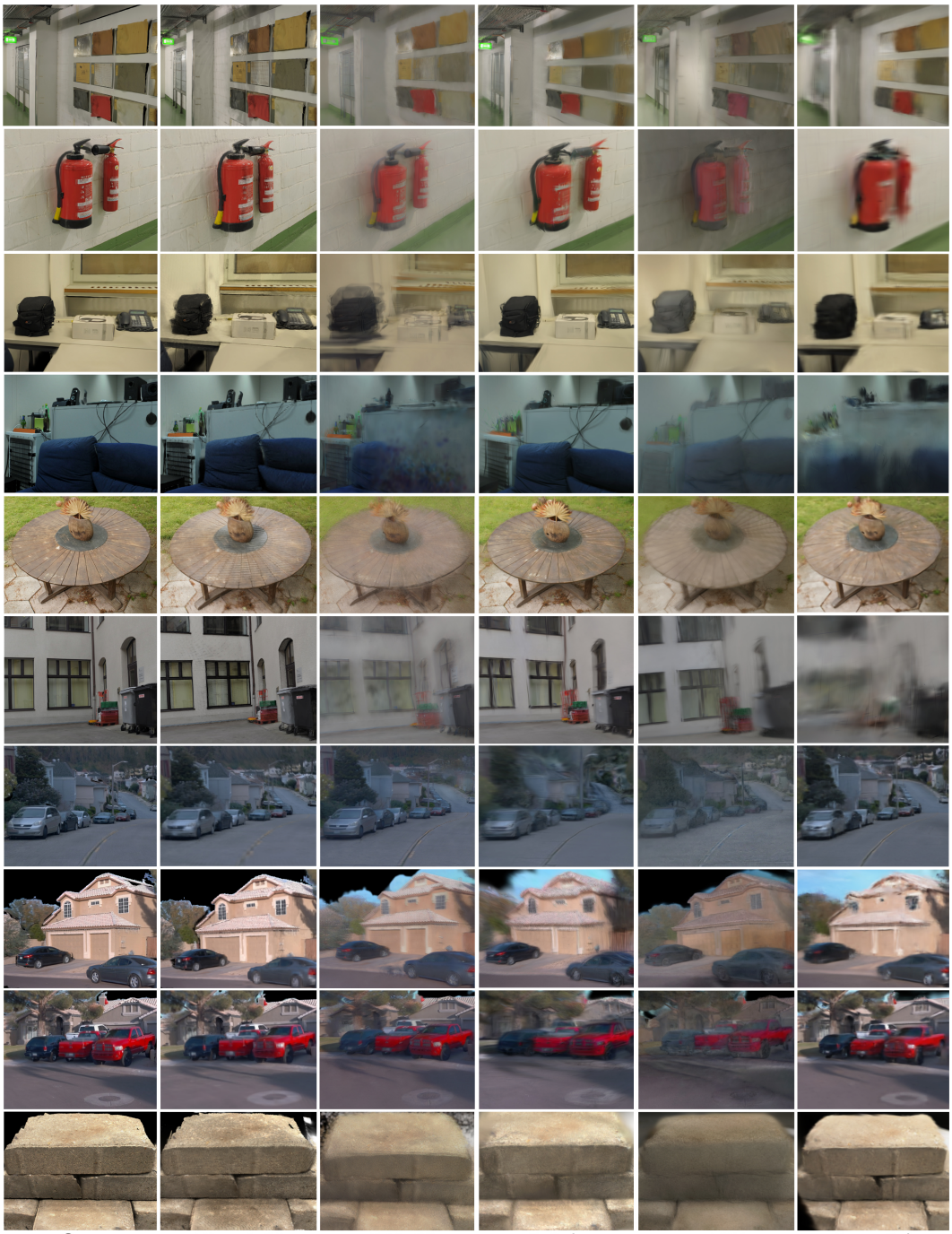

Figure 11: Qualitative comparisons across diverse scenes and viewpoints with multi-view input.

A.9 Application to Panoramic 3D Content Generation.

We explore SplatAnything for panoramic-image-based 3D content generation. We use WorldPano (Team et al., 2026) to generate a panoramic image and convert it into cubemap perspective views with a $95^\circ \times 100^\circ$ FoV, which remains substantially wider than typical perspective inputs. These views are fed directly into SplatAnything for 3D Gaussian reconstruction. As shown in Fig. 13, SplatAnything produces high-quality novel-view synthesis under this challenging wide-FoV setting. These results demonstrate its robustness to substantially different camera configurations and further highlight the potential of SplatAnything as a general feed-forward 3DGS framework.

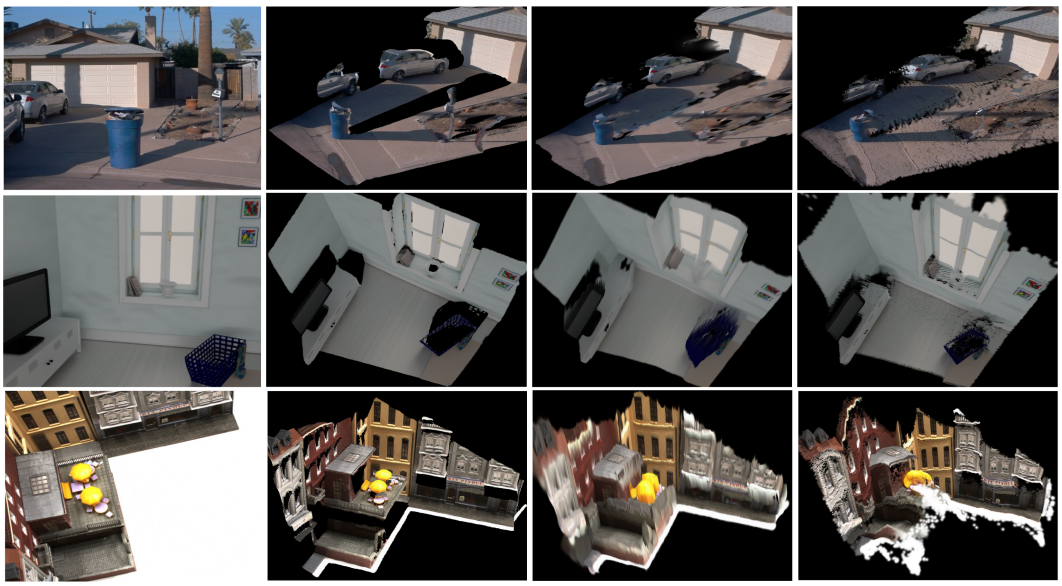

Figure 12: Qualitative comparisons across diverse scenes and viewpoints with monocular input.

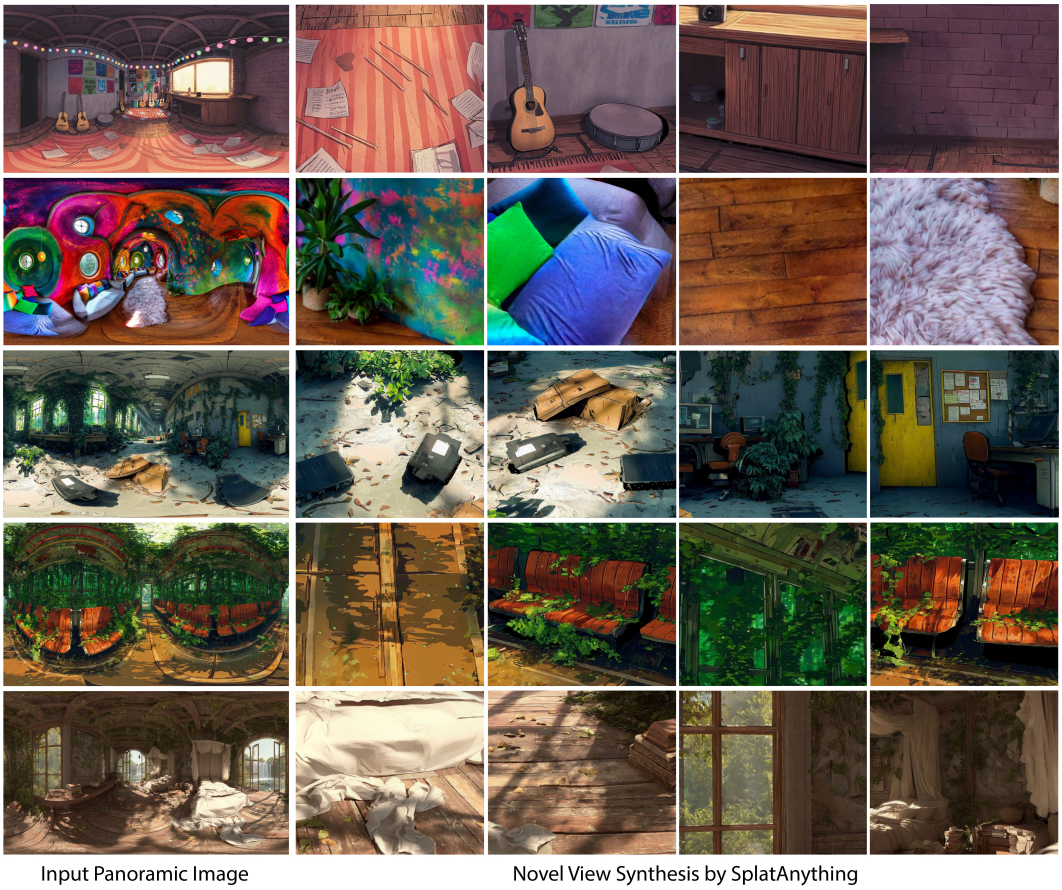

Figure 13: Qualitative results of SplatAnything for panoramic images reconstruction. SplatAnything can reconstruct high-quality 3DGS representations from panoramic images with different field of view, demonstrating its strong generalization ability.



Figure 14: More qualitative results of our method. SplatAnything feed-forward reconstructs high-quality 3DGS representations generalizing to arbitrary scenes, scales, and viewpoints in a few seconds.