Point2GS: Generalized Plug-and-Play 3D Gaussian Splatting

Mengqi Zhang[1*] Yang Fu[2*] Judy Hoffman[1] Sifei Liu[3] Xiaolong Wang[2]
[1] Georgia Institute of Technology [2]University of California, San Diego [3]NVIDIA

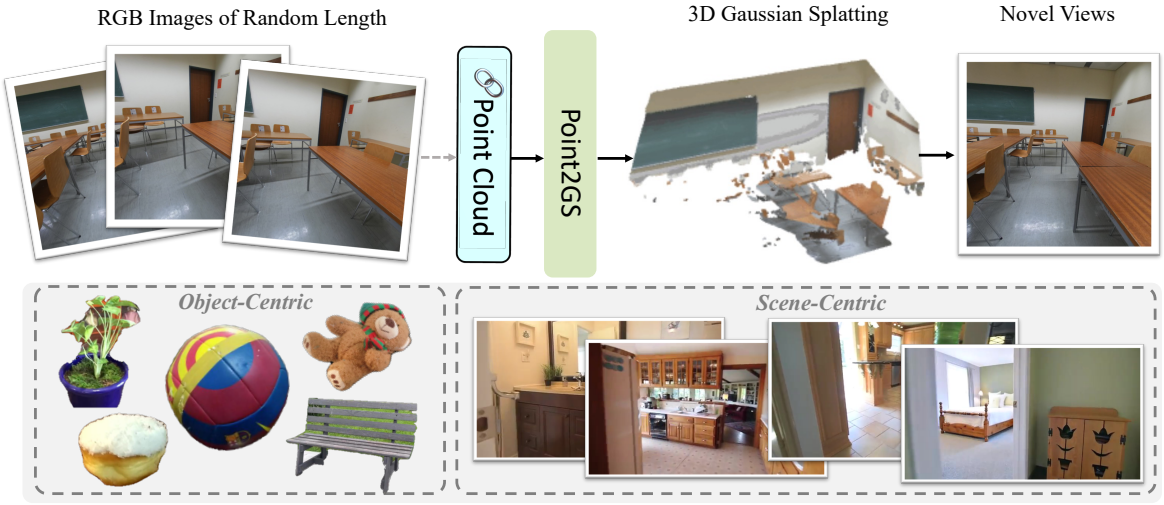

Figure 1. **Point2GS is a generalized plug-and-play feed-forward 3D Gaussian splatting model**. It uses point clouds as an intermediate representation to build the mapping between 3D points and Gaussian primitives. The model can be seamlessly applied after any point prediction model or scanned point cloud, eliminating the need for joint training. With training on only two datasets, our method achieves good performance on both object and scene reconstruction and shows strong generalization on out-of-domain data.

Abstract

Reconstructing 3D scenes has long been a challenging task and has witnessed great improvements recently, especially in an uncalibrated pose-free setting. Most existing models rely on one additional Gaussian prediction head, jointly trained with the large pointmap prediction network, leading to point overlap and limited flexibility. In this paper, we introduce Point2GS, an innovative approach for 3D Gaussian reconstruction that can be flexibly used with various point inputs, whether from uncalibrated RGB images or scanned points. By integrating both rich RGB appearance features from multi-view images and geometry information from point clouds, Point2GS efficiently predicts the 3D Gaussian parameters for reconstruction. Trained on only two public datasets CO3Dv2 and ScanNet++, Point2GS exhibits strong generalization capabilities across both object-centric and scene-level reconstruction, enabling zero-shot inference on new datasets and user-provided sequential images. The model demonstrates robust, flexible and versatile reconstruction performance across diverse scenarios.

1. Introduction

With the advent of 3D Gaussian Splatting (3DGS), 3D reconstruction has been revolutionized by efficient, fast learning and differentiable rasterization. This approach eliminates the expensive volumetric sampling required by previous methods like NeRF [21, 26, 61], yet still achieves high-quality reconstruction and synthesis results. However, the conventional 3DGS framework [12, 17, 57] remains limited by its reliance on per-scene optimization, which hinders its applicability in real-time and casual settings. Additionally, it requires supervision from hundreds of input views along with precise annotations, such as camera poses and intrinsic matrices for rendering, which are usually obtained from undifferentiable Structure-from-Motion (SfM) solvers [15, 33, 39], significantly increasing data collection overhead.

These limitations have driven recent advancements in generalized Gaussian Splatting models for sparse-view reconstruction [2, 5, 7, 16, 36, 64] and differentiable geometry prediction networks using uncalibrated data [20, 24, 45, 48, 53, 54, 58]. Some generalizable 3D reconstruction approaches[2, 7] leverage feed-forward networks with

inter-view interaction modules such as full cross-attention or epipolar attention, to achieve photorealistic results without per-scene optimization. However, these methods still require accurate camera poses and fixed length of input views, limiting their usage in real-world scenarios. To address this problem, more recent approaches [5, 16, 36, 64] have emerged to integrate pose estimation, depth estimation, and 3D reconstruction into a unified framework, representing a significant step forward. However, they often suffer from fixed input length constraints and generally require joint training of introduced Gaussian head layers and pointmap prediction backbone, leading to pixel redundancy in overlapping areas among images and reduced flexibility in Gaussian parameter prediction.

To address these challenges, we introduce **Point2GS**, a generalizable, plug-and-play 3D reconstruction model that predicts 3D Gaussian parameters using point cloud as an intermediate input representation. Unlike prior approaches that rely on joint optimization with depth or pose estimation on pixel space (each pixel is regressed to one Gaussian primitive), Point2GS establishes a robust mapping from point cloud to Gaussian primitives. We propose two novel modules:

- **Hybrid geometric-appearance feature**. Point2GS represents each 3D point by leveraging global geometry features extracted from a module akin to Point-BERT [62], along with rich RGB appearance features from DINOv2 [29]
- **Point densification strategy**. To address potential sparsity problem during inference, we sample new points within the voxelized space of input point cloud with an adaptive feature sampling method to efficiently extract features for new points without introducing much computation overhead.

With these indispensable modules, Point2GS is flexible to handle various point cloud sources and image length, ranging from learning-based sparse-view SfM [52] to diverse pixel-wise geometry prediction methods [20, 53, 54] without requiring additional training. Since the model is not specifically designed like previous methods, it can be seamlessly plugged-and-played. We trained the model on only two public datasets [31, 60] for both object-centric and scene-scale reconstruction. Point2GS demonstrates strong generalization abilities. Notably, it can reconstruct 3D scenes in a zero-shot manner for unseen and in-the-wild scenarios, including phone-captured and web-downloaded image sequences.

We comprehensively evaluate Point2GS across multiple key aspects. Our contributions are as follows:

- We propose Point2GS, a novel method for 3D reconstruction which decouples Gaussian reconstruction from point cloud generation. By using point clouds as an intermediate representation, Point2GS supports flexible frame length and various input point types
- We demonstrate superior results in both object-centric and scene-level reconstruction with training on only two datasets, which is data-efficient.
- We show that Point2GS generalized effectively across domains, achieving zero-shot performance on real-world datasets.

2. Related Works

Multi-View stereo. Traditional multi-view stereo reconstruction follows a four-step pipeline: keypoint detection and trajectory prediction [1, 10, 25, 33]; camera parameter estimation [46, 47, 49], point triangulation [15], and bundle adjustment refinement [44, 55]. With the emergence of deep learning, recent approaches [13, 51, 59] integrate depth prediction and camera pose estimation into neural networks. While some methods [46, 50, 55] maintain the traditional four steps, recent innovations [20, 24, 45, 48, 53, 54, 58] directly predict point maps from uncalibrated stereo images. For instance, DUSt3R [54] leverages large-scale datasets and a Transformer-based network to infer 3D point positions from RGB image pairs. Its successor, MASt3R [20], enhances DUSt3R with a dense local feature head and an additional matching loss for improved point accuracy.

3D Reconstruction and Novel View Synthesis. Various scene representations [3, 11, 34, 40] enable 3D reconstruction and photorealistic novel view synthesis from multi-view RGB images. Early neural fields [27, 34] encoded 3D scenes in latent codes, restricting reconstruction to single objects. Subsequent works [14, 22] mapped pixel space to radiance fields, extending to unbounded scenes. Alternative light field representations [9, 35, 41], improved rendering speed at the cost of flexibility. Recently, 3D Gaussian Splatting [17] introduced explicit 3D Gaussian primitives with differentiable rasterization, significantly enhancing both training and rendering efficiency.

Generalized Gaussian Splatting. While 3DGS offers advantages over NeRF-based methods, it still requires per-scene optimization with hundreds of views and precomputed camera poses. Additionally, 3DGS depends on a coarse point cloud for Gaussian initialization, making it less suitable for in-the-wild scenarios with sparse views and unknown camera poses [3, 28]. These limitations have driven research into generalized Gaussian Splatting [2, 4, 6, 7, 63].

Splatter Image [43] predicts Gaussian parameters from a single image using a U-Net but is limited to object reconstruction. While subsequent works [42] attempt to generalize to scene-level tasks, they remain constrained by sparse single-image inputs. More recent methods like pixelSplat [2] and MVSplat [7] improve on this by regressing Gaus-

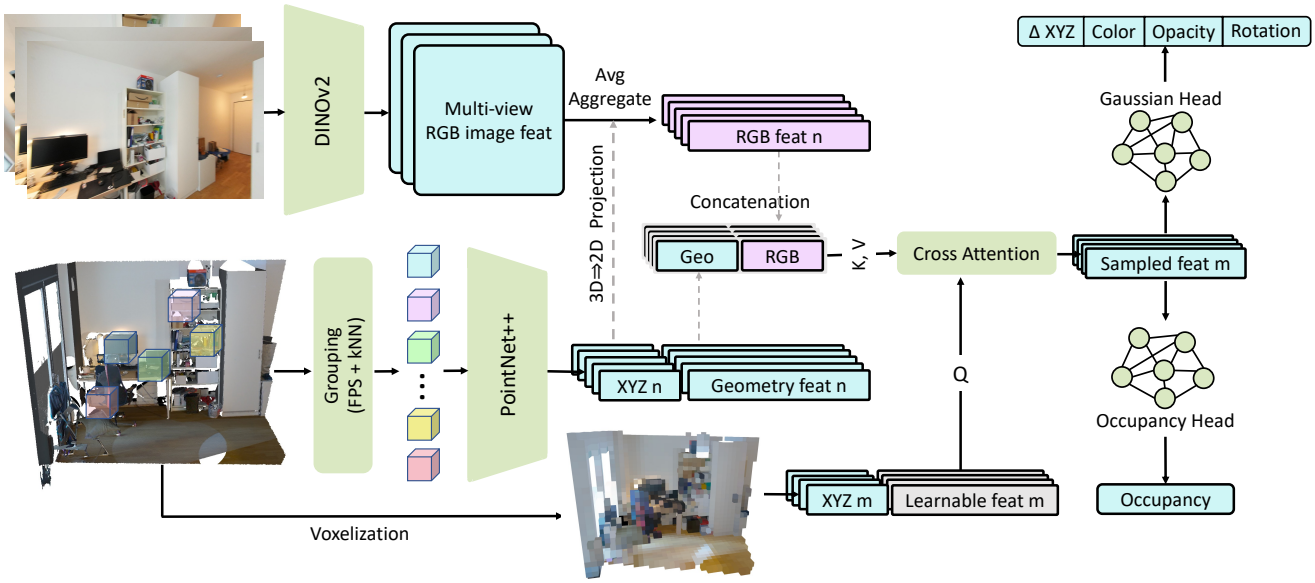

Figure 2. **Model architecture of Point2GS.** Point2GS learns the mapping between the point cloud and Gaussian parameters. It leverages geometric features extracted from the point cloud using a Point-BERT akin module and rich multi-view RGB features from pretrained DINOv2. To deal with the sparsity problem, we propose a point densification strategy with adaptive feature sampling in voxelized point space. The new sampled features are then used to predict the Gaussian parameters

sian parameters from dual-view inputs, extendable to multiple views. These models rely on cross-attention or cost-volume techniques to predict depth maps, which are then unprojected to determine Gaussian positions. However, their computational cost and reliance on camera poses for depth estimation limit their scalability and applicability in unconstrained environments.

Most closely aligned with our setting is Splatt3r [36], which operates solely on uncalibrated RGB images during inference. Built upon a pointmap prediction network [20], Splatt3r incorporates an additional head for Gaussian parameter prediction, enabling seamless application in unconstrained environments. However, its architecture is specifically designed for dual-view inputs, making its extension to multiple views non-trivial. In contrast, our proposed Point2GS accommodates arbitrary numbers of input views and sequence lengths while maintaining the advantage of requiring only uncalibrated RGB images during inference.

3. Method

Given a raw RGB image sequence, our goal is to reconstruct the scene and synthesize novel views using 3D Gaussian parameters as the underlying representation. Unlike previous methods that directly create a 1-1 mapping between 2D pixel space to Gaussian primitives, often resulting in redundant and overlapping points in 3D space, Point2GS introduces a novel structured, feed-forward approach that operates directly in the 3D point domain. To achieve this, Point2GS extracts structural features from input point clouds using a geometry-aware encoder akin to Point-BERT [62]. Simultaneously, a pretrained DINOv2 [29] backbone provides localized rich RGB features from multi-view images, which are then averaged and lifted up to 3D points. A point densification strategy together with feature sampling is leveraged to mitigate the sparsity problem. Then the sampled features are then fused through a MLP head, which predicts the Gaussian attributes for each 3D point. This design allows for flexible 3D reconstruction without per-scene optimization or reliance on input point network.

We first provide a brief overview of 3D Gaussian Splatting in Section 3.1, followed by a detailed explanation of the Point2GS framework in Section 3.2. Finally, we describe how point clouds can be obtained using uncalibrated RGB images in Section 3.3.

3.1. Preliminary

3D Gaussian Splatting (3DGS) [17] represents a scene using a set of 3D Gaussian primitives, each parameterized by spatial and appearance attributes. Each Gaussian can be interpreted as an elliptical region centered at a point, emitting radiance across its area. The representation consists of the following parameters: mean $\mu \in \mathbb{R}^3$, covariance $\Sigma \in \mathbb{R}^{3\times3}$, opacity $\alpha \in \mathbb{R}$, rotation quaternion $q \in \mathbb{R}^4$, scale $s \in \mathbb{R}^3$ and color $S \in \mathbb{R}^{3\times d}$, where d is the spherical harmonics degree. The differentiable rasterization module enables backpropagation of gradients to these parameters,

making the representation learnable. Conventional 3DGS relies on per-scene optimization. However, this approach is prone to local minima, sensitive to point cloud initialization, and difficult to generalize across diverse scenes. Recent works [2, 4, 6, 7, 63] introduce feed-forward approaches that predict Gaussian parameters from sparse input views using neural networks, which can be formulated as:

$$(\mu, \Sigma, \alpha, q, s, S) = f_\theta(I, A), \tag{1}$$

where I are input 2D images, and A represent corresponding annotations, such as ground-truth camera poses. While these feed-forward approaches improve generalization, they often require a fixed number of input frames or joint tuning with point or depth prediction network.

In contrast, our method is independent of the input frames or points by reconstructing the Gaussian from voxelized 3D point cloud space, making it a more flexible and scalable pipeline.

3.2. Point2GS Model

Point2GS predicts 3D Gaussian parameters directly from point cloud representation, integrating both geometric and visual cues:

$$(\mu, \Sigma, \alpha, q, s, S) = f_\theta(P, I), \tag{2}$$

where $P \in \mathbb{R}^{N\times 3}$ represents the input point cloud and $I \in \mathbb{R}^{v\times 3\times H\times W}$ denotes the set of multi-view context images providing additional visual information. The model architecture illustrated in Figure 2 consists of several key components: geometry feature extraction, RGB visual feature extraction, point densification and feature sampling, and Gaussian parameter prediction, which we detail below.

Geometry feature extraction Unlike previous approaches that rely solely on RGB features to predict Gaussian parameters, Point2GS incorporates explicit geometric information. Given an input point cloud, we extract structural features using a Geometry Encoder inspired by Point-BERT [62]. The encoder processes the point cloud hierarchically. First, farthest point sampling (FPS) selects a subset of key points $p_i \subset P, i = 1 \dots m$ to ensure uniform coverage. For each key point p_i, we gather its k-nearest neighbors (kNN) to form a local group $G_i = \{p_i, p_{i1}, p_{i2}, \dots, p_{ik}\}$. Each local group is then processed using a module similar to PointNet++ [30]:

$$F_G(p_i) = g_\theta(G_i) = \max\left(h_\theta(p_{i_j})\right), p_{i_j} \in G_i, \tag{3}$$

where h_θ is a nonlinear function, and max-pooling aggregates them into geometry feature F_G. Finally, the structural features of the entire point cloud are obtained by concatenating all processed group 3D features.

RGB visual feature extraction Apart from the geometry features, we extract rich visual features from multi-view context images using a pretrained DINOv2 model [29]. During training, ground-truth or predicted camera poses are used to project each key point p_i onto the multi-view images. The projected pixels are used to guide feature extraction:

$$F_V(p_i) = \frac{1}{v}\sum_{j=1}^{v} D(I_j, \pi_j(p_i)), \tag{4}$$

where $D(I_j, \pi_j(p_i))$ extracts the DINOv2 feature of point p_i from image I_j. The features are averaged across all context views. The final per-point feature is represented by concatenating geometry and visual features $F(p_i) = [F_G(p_i), F_V(p_i)]$

Point densification and feature sampling A potential challenge introduced by point-based Gaussian prediction is handling sparsity, especially at inference. Sparse points make it difficult to reconstruct continuous surfaces, resulting in holes in the rendered scene. Additionally, operations such as FPS, kNN, and projection become inefficient if directly sampling new points. To mitigate these issues, we introduce a voxel-based densification strategy. We first partition the 3D space into voxel grids and sample additional points within each occupied voxel. Once additional points are introduced, we propagate features from the original point cloud $F(p_i)$ using an attention-based interpolation mechanism.

$$\begin{aligned} F(p') &= \sum_i w_i F(p_i), \\ w &= Attention(Q(F_{p'}), K(F(p)), V(F(p)), \end{aligned} \tag{5}$$

where $F_{p'}$ is learnable preset features for new point query and $F(p')$ is the aggregated feature . In real implementation, pairwise Euclidean distance is also modulated into the attention condition.

Gaussian parameter prediction With the aggregated features, the final step is to predict the 3D Gaussian parameters. A series of MLP layers map each point feature to its corresponding Gaussian attributes:

$$(\delta\mu, \Sigma, \alpha, q, s, S) = f_\theta(P, I) = f_{\theta'}(F(p'), \tag{6}$$

Instead of directly predicting the Gaussian centers, we model them as offsets from the input point positions. The predicted Gaussians are then used for rendering novel views, with supervision from the ground-truth images.

3.3. Point Cloud from Uncalibrated RGB Images

During training, we leverage scans provided in the dataset or lifted points from ground-truth depth maps to obtain high-quality point clouds. However, at inference, such data is generally unavailable. To address this, we leverage pretrained geometry prediction models such as VGGSfM [52] or DUSt3R and succeeding works [20, 54] to provide inputs. Since these models are learning-based and annotation-free, they can be seamlessly used into our model. Although

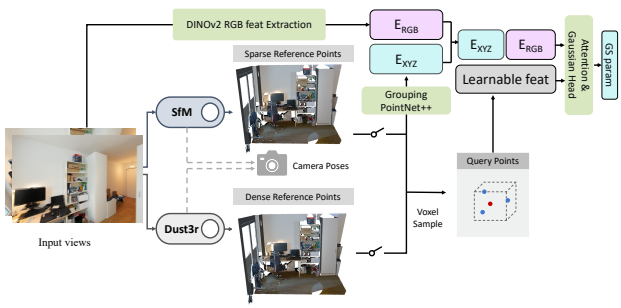

Figure 3. **Overview of the pipeline using uncalibrated RGB images.** During inference, the point cloud can be obtained from the learning-based structure-from-motion model [52] or point map prediction networks [54], with camera poses and intrinsic matrices predicted simultaneously. The point cloud and RGB images are used as inputs to Point2GS for reconstruction. Benefit from the decoupling of Gaussian prediction from points, the model can be seamlessly integrated with any succeeding point prediction maps without joint training.

our model is trained using scanned or depth-lifted points, it effectively transfers to the reconstructed point clouds during inference, allowing for flexible integration and scalable generalization to out-of-domain data thanks to the decoupling of point and Gaussian prediction in Point2GS. More visual results are shown in Figure 5.

4. Experiment

In this section, we provide details on our implementation for both training and inference. We evaluate our model through qualitative and quantitative comparisons against baseline methods, and further demonstrate its zero-shot generalization on unseen datasets and user-provided images.

4.1. Implementation Details

Training details The training process for Point2GS is divided into two phases. In phase 1, we pretrain the model on CO3D-v2 [31], an object dataset that provides point clouds derived from traditional structure-from-motion methods. This phase helps the model develop a strong initialization for scene reconstruction. Notably, even after training solely on objects, Point2GS already demonstrates reasonable zero-shot generalization to scene-level reconstruction. In phase 2, the model is further trained on ScanNet++ [60], a large-scale indoor scene dataset with ground-truth depth maps and laser-scanned point clouds. Directly using the entire scene point cloud, however, is impractical due to its large scale. To address this, we randomly sample RGB frames from the original sequence and lift corresponding pixels into 3D space using ground-truth depth information.

A key point is the scale discrepancy between training and inference conditions, as scene point clouds from ground-truth scans or depth-based lifting may vary significantly. To bridge this gap, we normalize the scale during training, ensuring robustness across different input sources.

For CO3D-v2, we train on 512×512 images with a learning rate of 3.0×10^{-4}. For ScanNet++, images are resized to 480×640, and the learning rate is adjusted to 1.0×10^{-4}. We use the Adam optimizer [18] with 1cycle learning rate policy [38] to anneal the learning rate. The training is supervised with a combination of MSE and LPIPS losses. The model is trained on 8 A100 GPUs, running for approximately 100 epochs on CO3D-v2 and 20 epochs on ScanNet++.

Inference details During inference, we use DUSt3R [54] for two-view inputs, VGGSfM [52] and Spann3R [48] across multiple views to provide point cloud and camera poses. However, the predicted point cloud often contains outliers and redundancies, which degrade reconstruction quality. Outliers disrupt the feature sample process (Eq. 5) for newly sampled points, leading to abnormal opacity and scale predictions. Meanwhile, redundant points reduce the efficiency of input point cloud sampling by FPS, increasing under-reconstructed regions. To mitigate these issues, we apply voxel downsampling and statistical outlier removal, ensuring more stable and accurate reconstructions.

Evaluation metrics We evaluate the reconstruction quality by using three widely used metrics: PSNR, LPIPS, and SSIM between ground-truth images and rendered novel views. PSNR measures pixel-wise differences, quantifying the overall image fidelity; LPIPS is a deep-learning-based perceptual similarity metric that computes the perceptual distance between two images; SSIM evaluates similarity based on contrast and structure information. These metrics are used to compare Point2GS against existing baseline models.

Baselines We evaluate Point2GS against models that require no ground-truth annotations during inference. The most comparable method for scene reconstruction is Splatt3R [36], which extends MASt3rR [20] with an additional Gaussian parameter prediction head. We also include direct point cloud rendering from MASt3rR for comparison in quantitative evaluation. Both models have been trained on ScanNet++. For object reconstruction, Point2GS focuses primarily on the object regions, ignoring background information. Since no existing generalized feed-forward Gaussian Splatting models operate without camera poses on objects, we compare against three prior works based on different reconstruction techniques Vid-AE [19], RUST [32], and FlowCam [37]. These comparisons highlight Point2GS's ability to reconstruct scenes directly from 3D point clouds.

4.2. Object Reconstruction

In this section, we present mainly quantitative comparisons for object reconstruction. We evaluate our model's performance on CO3D-v2 [31] with baseline models, showcas-

Method	CO3D-v2 Hydrant			CO3D-10		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Vid-AE [19]	15.47	-	0.5113	17.78	-	0.5376
RUST [32]	18.81	-	0.6071	19.45	-	0.6046
FlowCam [37]	19.37	-	0.4143	21.14	-	0.3703
Ours	**22.29**(16.42)	0.88	**0.1107**(0.1546)	**26.93**(17.05)	0.93	**0.0697**(0.1294)

Table 1. **Quantitative comparison with previous baseline methods on CO3D-v2**. The results are tested on the specific "hydrant" category and ten other categories from CO3D-v2 testset. Values in the bracelet are average over the mask, values outside are average over the whole image.

ing its ability to generalize across object instances and categories.

Quantitative Comparison We compare Point2GS against existing generalizable object reconstruction methods on CO3D-v2. To the best of our knowledge, no prior generalized 3D Gaussian Splatting model has been trained for object reconstruction. Therefore, we benchmark against three alternative approaches: (1) FlowCam [37], a generalizable neural field network that reconstructs objects without camera poses using pixel-aligned scene flow; (2) Vid-AE [19], a model that learns 3D representations from camera motion and video observations; and (3) RUST [32], a method that learns neural scene representations from novel views without requiring camera poses. We evaluate all models on the specific "hydrant" category and ten other categories from the test subset. As previous baselines do not report SSIM scores, we omit them from our comparison. Since Point2GS primarily reconstructs the object-centric regions, we apply object masks when computing evaluation metrics, which will be further explained in Section 4.3. To ensure a comprehensive evaluation, we report both the overall image evaluation score(numbers outside parentheses) and object-region evaluation score(numbers inside parentheses). Given that CO3D-v2 images contain relatively small object-covered area, LPIPS values appear lower than usual. Our model achieves superior PSNR and LPIPS scores on full-image evaluations, demonstrating its effectiveness in reconstructing high-fidelity, object-centric scenes.

4.3. Scene Reconstruction

Another key application of our method is scene reconstruction. While achieving high-quality results on in-distribution data is important, strong generalization to out-of-distribution (OOD) scenes is crucial for evaluating robustness. In this section, we evaluate our model's performance on both the training dataset and OOD data, using two different types of input points for visualization.

ScanNet++ Reconstruction Comparison Table 2 presents our quantitative results on ScanNet++ dataset. We compare Point2GS with Splatt3R, which also reconstructs scenes from uncalibrated RGB images, along with two adapted settings: MASt3R point cloud direct rendering and pixelSplat (tested with either MASt3R-reconstructed or ground-truth cameras). Results of these two adapted models were reported in Splatt3R. Following prior evaluations, we use two input views for fair comparison. To ensure consistency, we generate a point cloud using Dust3R from two context views without any joint training and render a novel view in the first frame's coordinate system. Sparse-view inputs can lead to missing novel-view regions, resulting in black areas in reconstructions. To account for this during fair evaluation, we apply the loss mask strategy from Splatt3R, where a mask is computed via frustum culling using ground-truth pose and depth maps. The PSNR and LPIPS values in brackets represent scores computed within the masked regions, while those outside brackets reflect full-image averages.

Overall, Point2GS surpasses other baselines in PSNR and SSIM, although its LPIPS score is slightly higher, likely due to blurriness in 3D space. Despite this, our model's architecture enables efficient training and seamless usage with different point inputs.

Figure 4 provides a visual comparison against baseline methods. Since a pixelSplat checkpoint trained on ScanNet++ is not released, and MASt3R only produces point maps rather than direct reconstructions, we mainly compare against Splatt3R for visual results. Notably, since our point cloud initialization relies on Dust3R rather than the point network weights used in Splatt3R, differences in mask areas can lead to inconsistencies such as invalid regions and holes in novel views. From the results, we could see that our model demonstrates superior handling of invalid borders, whereas Splatt3R exhibits distortions along the boundaries between valid Gaussian primitives and empty regions. Additionally, its pixel-based nature introduces outliers, whereas Point2GS produces fewer artifacts, resulting in more stable reconstructions.

Zero-shot Cross-dataset Reconstruction Since Point2GS learns a direct mapping from pixels to 3D Gaussian parameters, it naturally generalizes to diverse and unseen datasets. Instead of relying on a fixed global scene distribution, the model emphasizes local geometric structures, making it more adaptable to out-of-distribution inputs. To illustrate this, Figure 5 showcases zero-shot reconstruction results on three previously unseen datasets. RealEstate10K [65] and ScanNet [8] primarily feature indoor scenes, similar

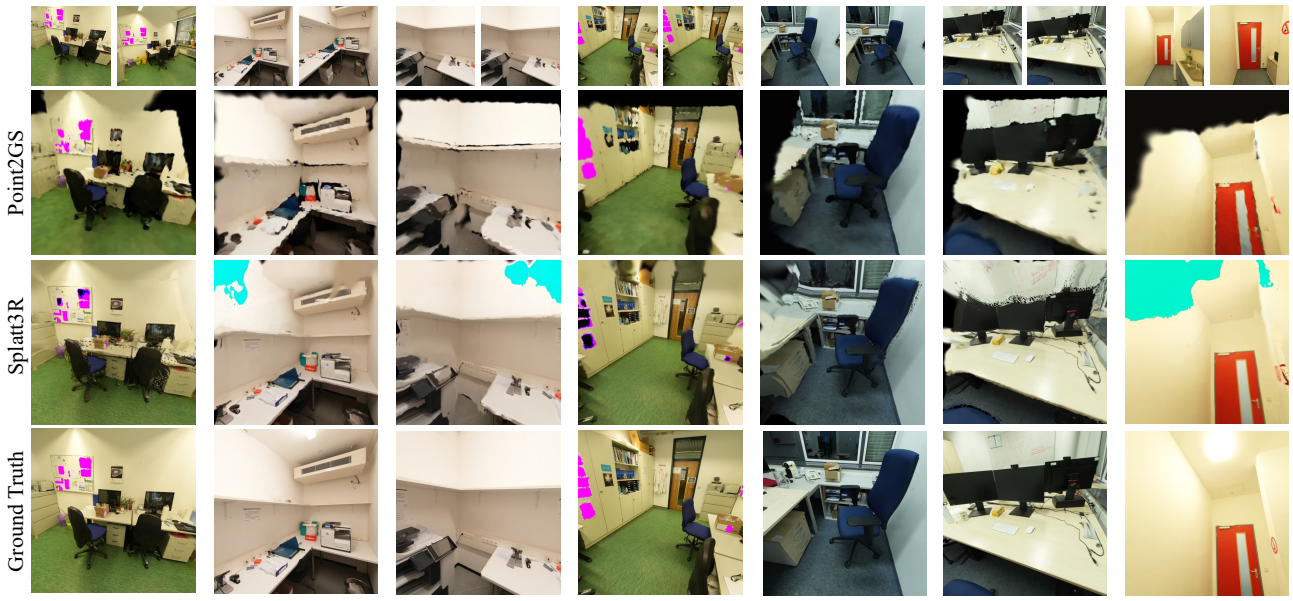

Figure 4. **Novel view synthesis comparison on ScanNet++.** We evaluate our model on the ScanNet++ test set and compare the visual results with Splatt3R. The first row shows input context views, while the subsequent rows present novel view renderings. The last row displays the ground-truth results

to ScanNet++. ACID consisting of mainly outdoor environments, introducing greater distribution shift challenges due to infinite-depth sky regions (see the last row of Figure 5). We evaluate novel view synthesis using two types of point cloud inputs: SfM points and DUSt3R point maps. Our model seamlessly adapts across datasets, demonstrating strong generalization ability across domains. More results are provided in the Supplementary.

To quantitatively evaluate our approach on zero-shot tasks, we randomly sampled 50 subsequences from the zero-shot ScanNet dataset, and sample according to the evaluation file in RealEstate10K for scene reconstruction. Our results show that Point2GS achieves performance comparable to Splatt3R on ScanNet, but generally better performance on RealEstate10K. These results confirm that even without additional training, Point2GS achieves good performance, demonstrating strong adaptability across both limited and in-the-wild datasets.

	ScanNet++		
Method	PSNR↑	SSIM↑	LPIPS↓
MASt3R [20]	18.51 (12.96)	0.718	0.259 (0.280)
pixelSplat (MASt3R cams) [2]	15.96 (10.64)	0.648	0.379 (0.405)
pixelSplat (GT cams) [2]	15.92 (10.61)	0.643	0.381 (0.407)
Splatt3R [36]	19.66 (14.38)	0.770	**0.229** (0.243)
Ours	**21.54** (15.99)	**0.818**	0.264(0.294)
	ScanNet		
Method	PSNR↑	SSIM↑	LPIPS↓
Splatt3R [36]	**19.66** (14.49)	0.729	**0.217** (0.228)
Ours	19.55 (**15.99**)	**0.732**	0.410(0.409)
	RealEstate10k (small)		
Method	PSNR↑	SSIM↑	LPIPS↓
DUSt3R [54]	14.101	0.432	**0.468**
MASt3R [20]	13.534	0.407	0.494
Splatt3R [36]	14.352	0.475	0.472
Ours	**15.112**	**0.562**	0.517

Table 2. **Quantitative Comparison with baseline models on scene reconstruction.** Point2GS is tested on ScanNet++ which has been seen during training, and two zero-shot datasets, ScanNet and RealEstate10k to evaluate the fidelity and generalization ability of our method.

5. Ablation Study

In this section, we conduct ablation experiments to evaluate different components of our method. Specifically, we analyze the impact of the point densification, the sample strategy, and parameter choices.

There are two approaches for densification: one involves directly sampling points from the provided point clouds, and the other is our densification module in Point2GS. We compare the results to highlight the importance of our proposed method. In direct sampling, our model converges too slowly, exhibits higher loss, and produces significantly blurred reconstructions. To further refine our model, we conduct experiments to determine optimal parameter choices. For efficient evaluation, we train different model

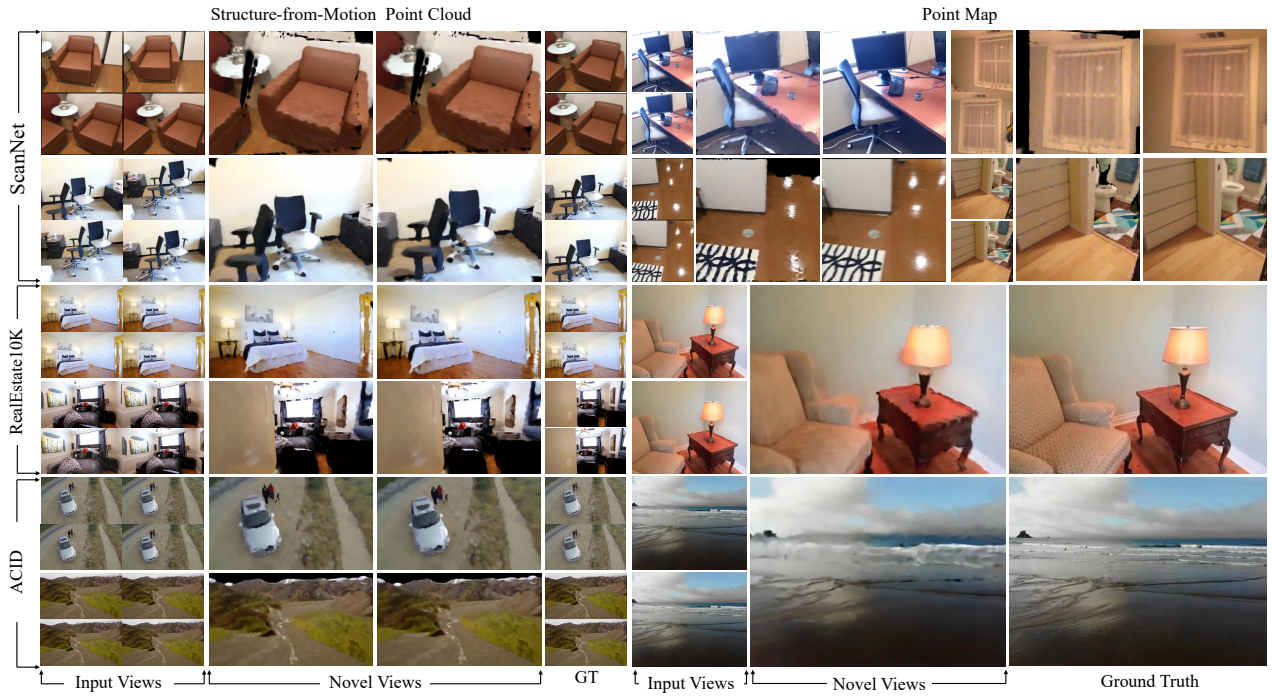

Figure 5. **Cross-domain novel view synthesis on ScanNet, RealEstate10K and ACID.** The model is trained on CO3D-v2 and ScanNet++, and directly tested on these three datasets in a zero-shot manner without any training. Structure-from-Motion points are sparse while point maps are relatively dense

	PSNR↑	SSIM↑	LPIPS↓
- Point densification	13.33 (7.77)	0.739	0.411 (0.450)
- XYZ offset	21.21 (15.65)	0.811	0.268 (0.298)
+ Spherical Harmonics	19.22 (13.66)	0.787	0.311 (0.35)
- Voxel Resolution (test-time)	20.62 (14.76)	0.807	0.286 (0.333)
- Group number (test-time)	20.31 (14.75)	**0.814**	0.287 (0.322)
Ours	**21.31** (15.74)	0.813	**0.265**(0.295)

Table 3. **Ablation Study**. The importance of point densification module and other parameters is investigated to demonstrate the model design.

variants on a subset of the datasets and test them on 500 ScanNet++ samples. As shown in Table 3, higher voxel resolution during test-time sampling improves reconstruction quality, while increasing the number of reference points enhances the model's ability to capture fine-grained scene geometry.

Another key design choice in our model is the sampling strategy, which includes voxel sampling, adjacent sampling, and uniform sampling. Adjacent sampling prioritizes nearby voxels to maintain local consistency, while uniform sampling distributes points across the entire scene, a strategy commonly used in prior methods [23, 56]. However, as shown in Figure 6, these two approaches lead to blurrier results. In contrast, voxel sampling preserves finer scene details and performs better in reconstructions and novel view synthesis.

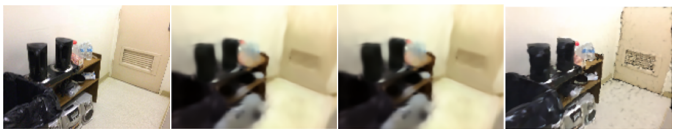

Figure 6. Visual results of different query point sampling methods: uniform space sampling, adjacent space sampling and our adopted voxel space sampling.

6. Conclusion

In this paper, we present Point2GS, a generalized plug-and-play 3D Gaussian Splatting model using point cloud as intermediate representation, to achieve object and scene reconstruction. By building the mapping between 3D point clouds and Gaussian parameters, our model can be seamlessly generalized to diverse scenarios and input sources. We show the results on three test datasets using zero-shot way, which also demonstrates the potential of Point2GS to in-the-wild applications.

References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 2

[2] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image

pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024. 1, 2, 4, 7

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 2

[4] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision*, pages 338–355. Springer, 2025. 2, 4

[5] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *arXiv preprint arXiv:2411.04924*, 2024. 1, 2

[6] Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3r: Gradient guided generalizable reconstruction. In *European Conference on Computer Vision*, pages 305–323. Springer, 2025. 2, 4

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025. 1, 2, 4

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6

[9] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023. 2

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[11] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[12] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20796–20805, 2024. 1

[13] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 2

[14] Pengsheng Guo, Miguel Angel Bautista, Alex Colburn, Liang Yang, Daniel Ulbricht, Joshua M Susskind, and Qi Shan. Fast and explicit neural view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3791–3800, 2022. 2

[15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2

[16] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024. 1, 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9730–9740, 2021. 5, 6

[20] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 1, 2, 3, 4, 5, 7

[21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021. 1

[22] Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 806–815, 2023. 2

[23] Stefan Lionar, Xiangyu Xu, Min Lin, and Gim Hee Lee. Numcc: Multiview compressive coding with neighborhood decoder and repulsive udf. *Advances in Neural Information Processing Systems*, 36, 2024. 8

[24] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. *arXiv preprint arXiv:2412.09401*, 2024. 1, 2

[25] David G Low. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*, 60(2):91–110, 2004. 2

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[27] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3504–3515, 2020. 2

[28] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis

from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 4

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4

[31] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 2, 5

[32] Mehdi SM Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lučić, and Klaus Greff. Rust: Latent neural scene representations from unposed imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17297–17306, 2023. 5, 6

[33] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2

[34] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[35] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34: 19313–19325, 2021. 2

[36] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibarated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 1, 2, 3, 5, 7

[37] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. *arXiv preprint arXiv:2306.00180*, 2023. 5, 6

[38] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019. 5

[39] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 1

[40] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 2

[41] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022. 2

[42] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 2

[43] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10208–10217, 2024. 2

[44] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[45] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*, 2024. 1, 2

[46] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 2

[47] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2

[48] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 1, 2, 5

[49] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2021. 2

[50] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Visual geometry grounded deep structure from motion. *arXiv preprint arXiv:2312.04563*, 2023. 2

[51] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 2

[52] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21686–21697, 2024. 2, 4, 5

[53] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 1, 2

[54] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 1, 2, 4, 5, 7

[55] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 230–247. Springer, 2020. 2

[56] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9065–9075, 2023. 8

[57] Haolin Xiong. *SparseGS: Real-time 360° sparse view synthesis using Gaussian splatting*. University of California, Los Angeles, 2024. 1

[58] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. *arXiv preprint arXiv:2501.13928*, 2025. 1, 2

[59] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[60] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 2, 5

[61] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1

[62] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4

[63] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. *arXiv preprint arXiv:2408.13770*, 2024. 2, 4

[64] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. *arXiv preprint arXiv:2502.12138*, 2025. 1, 2

[65] 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. 6