Splatter-360: Generalizable 360° Gaussian Splatting for Wide-baseline Panoramic Images

Zheng Chen[1*†], Chenming Wu[2*], Zhelun Shen[2†], Chen Zhao[2], Errui Ding[2], Song-Hai Zhang[1‡]

[1]Tsinghua University
[2]Baidu VIS
[3]Zhejiang University

Abstract

Wide-baseline panoramic images are frequently used in applications like VR and simulations to minimize capturing labor costs and storage needs. However, synthesizing novel views from these panoramic images in real time remains a significant challenge, especially due to panoramic imagery's high resolution and inherent distortions. Although existing 3D Gaussian splatting (3DGS) methods can produce photo-realistic views under narrow baselines, they often overfit the training views when dealing with wide-baseline panoramic images due to the difficulty in learning precise geometry from sparse 360° views. This paper presents Splatter-360, a novel end-to-end generalizable 3DGS framework designed to handle wide-baseline panoramic images. Unlike previous approaches, Splatter-360 performs multi-view matching directly in the spherical domain by constructing a spherical cost volume through a spherical sweep algorithm, enhancing the network's depth perception and geometry estimation. Additionally, we introduce a 3D-aware bi-projection encoder to mitigate the distortions inherent in panoramic images and integrate cross-view attention to improve feature interactions across multiple viewpoints. This enables robust 3D-aware feature representations and real-time rendering capabilities. Experimental results on the HM3D [27] and Replica [32] demonstrate that Splatter-360 significantly outperforms state-of-the-art NeRF and 3DGS methods (e.g., PanoGRF, MVSplat, DepthSplat, and HiSplat) in both synthesis quality and generalization performance for wide-baseline panoramic images. The source code will be released.

1. Introduction

In recent years, 360-degree cameras and VR/AR headsets have gained widespread popularity, enabling seamless scene capture and delivering immersive experiences by presenting visuals directly to users. However, due to the labor-intensive nature of data acquisition and the high storage cost, panoramic images in industry settings often exhibit wide baselines. Generating novel views from these wide-baseline panoramas is essential to provide users full freedom of movement within virtual environments.

The emergence of neural radiance fields (NeRF) has demonstrated impressive performance in synthesizing novel views from perspective images. However, NeRF typically requires dense images captured from various angles and positions to address the well-known shape-radiance ambiguity. To reduce the need for dense input, generalizable NeRF appears [36, 41, 49]. Building on the success of these approaches, previous work such as PanoGRF [8] aims to mitigate overfitting in spherical radiance fields by leveraging 360-degree scene priors. Despite their strengths, NeRF-based methods rely on implicit representations, and the rendering process demands extensive network evaluations. Consequently, these methods require powerful GPUs, making them unsuitable for real-time applications on lightweight mobile devices like VR headsets, where low-power processing is essential. In contrast to NeRF, 3D Gaussian Splatting (3DGS) [17] moves away from implicit representation, instead adopting an explicit ellipsoid primitive representation. This representation is particularly well-suited for real-time rendering on traditional rasterization-based devices. However, like NeRF, 3DGS faces challenges with overfitting training views, especially when handling wide-baseline panoramic images.

This paper addresses the challenges of learning from wide-baseline panoramic images for real-time novel view rendering of 3DGS. Unlike previous generalizable 3DGS methods such as [4, 5, 19], our approach, dubbed as *Splatter-360*, operates end-to-end with panoramic image in-

[*]Denotes equal contribution.

[†]This work was done when Zheng Chen interned in Baidu VIS. E-mail: zhengchenecho@gmail.com

[‡]Corresponding author. E-mail: shz@tsinghua.edu.cn

puts and outputs. The design of our method is motivated by the following key factors:1) Panoramic images provide a full 360° field-of-view (FoV), whereas cubemap perspective images split the scene into six independent views, causing the issue of projecting points behind the camera in the source view when sampling point in the planar cost volume of the reference view — an issue that panoramic images naturally circumvent; 2) Our method eliminates the explicit back-and-forth conversion between panoramic and cube map representations, where backward conversions often introduce seam artifacts during stitching [2].

The end-to-end approach for generalizable 3D Gaussian Splatting (3DGS) presents two major challenges. First, panoramic images have significantly higher resolution than perspective images, requiring an extremely efficient network to avoid excessive memory consumption. To address this, we propose performing multi-view matching directly in the spherical domain by constructing a spherical cost volume through a spherical sweep algorithm. Second, the uneven distortion inherent in panoramic images complicates accurate depth estimation. We introduce a 3D-aware bi-projection encoding that leverages monocular depth, equirectangular projection, and cube-map branch information to tackle this issue. Additionally, by incorporating a cross-view attention mechanism to enhance feature interaction across different viewpoints, we obtain a robust 3D-aware feature representation.

Our contributions can be summarized as follows:

- We propose *Splatter-360*, a generalizable 3D Gaussian Splatting method for wide-baseline panoramic images, which is an end-to-end trainable network that can generate 3DGS primitives for novel panoramic view synthesis and enables real-time rendering experience.
- We introduce a spherical cost volume based on a spherical sweep algorithm in the feed-forward Gaussian splatting framework, which improves the geometry estimation capabilities of the network.
- Extensive experiments conducted on the HM3D [27] and Replica [32] datasets demonstrate that our Splatter-360 significantly outperforms state-of-the-art methods in handling wide-baseline panoramic images, both in terms of synthesis quality and generalization performance.

2. Related Work

2.1. Generalizable Novel View Synthesis

Recent advancements in novel view synthesis have been largely driven by NeRF and 3D Gaussian Splatting (3DGS). These methods can be broadly classified into two categories: per-scene optimization approaches [9, 37, 43] and generalizable models [4, 5, 10, 33, 45, 49]. Generalizable methods [4, 5, 10, 33, 45, 49] enable rapid reconstruction by learning priors from large-scale datasets, allowing for efficient feedforward inference. Below, we provide an overview of generalizable NeRF and 3DGS techniques:

Generalizable NeRF: Early attempts in generalizable NeRF models aim to reconstruct objects and scenes by generating pixel-aligned features for radiance field prediction, pioneered by [36, 41, 49]. Subsequent work such as NeuRay [24] predicts the visibility of 3D points relative to input images, focusing more on visible features. [10] and [40] further propose a multi-view transformer encoder with epipolar line sampling to capture multi-view geometric priors efficiently. MuRF [45] uses a target view frustum volume aligned with the target image plane to enable sharp, high-quality rendering. More recently, LRM [13] and Instant3D [20] designed a large transformer model to regress triplane NeRF feature for single- or sparse-view reconstruction.

Generalizable 3DGS: To address NeRF's high computational demands, generalizable 3DGS has emerged to simplify view synthesis through rasterization-based splatting, bypassing the need for expensive volume sampling. Methods like Splatter Image [34] demonstrate efficient object-level reconstruction by learning Gaussian parameters from single views. In contrast, pixelSplat [4] and Flash3D [33] have pushed this technique to scene-level reconstruction. More recently, [52] extends the paradigm of LRM [13] to scene-level with 3DGS, MVSplat [5] introduces a cost volume representation using plane sweeping to boost performance further. HiSplat [35] proposes a hierarchical structure for generalizable 3DGS. DepthSplat [46] leverages depth estimation to address multi-view depth methods' limitations and failure cases. However, most of these methods are designed for perspective images and struggle with panoramic inputs due to the large field of view and wide baselines. In response, we propose Splatter-360, extending the strengths of 3DGS to panoramic images.

2.2. Panoramic View Synthesis and Generation

Unlike perspective, novel view synthesis, panoramic novel view synthesis introduces unique challenges due to the distortions caused by equirectangular projection. Early works [1, 12] attempted to synthesize panoramic views using multi-sphere images. More recent approaches have focused on reconstructing radiance fields or 3D Gaussian scenes from dense panoramic inputs [2, 11]. The challenge intensifies with sparse 360° inputs, as depth estimation becomes significantly more difficult. A few methods seek to address this by leveraging the predicted depth and geometric warping to construct neural radiance fields for panoramic novel view synthesis [3, 18]. 360Roam [15] proposes to adapt neural rendering methods for panoramic inputs, struggling in wide-baseline scenarios. PERF [39], on the other hand, explores an inpainting-based approach to generate neural radiance fields from a single panoramic image. Most relevant to our work is PanoGRF [8], which introduces Generalizable Spherical Radiance Fields for wide-baseline panoramas. PanoGRF achieves state-of-the-art performance by directly aggregating geometry and appearance features of 3D sample points from each panoramic view. However, its inference and rendering speed are limited, making it unsuitable for real-time applications. In contrast, our work proposes a generalizable panoramic view synthesis pipeline designed to generate 3D Gaussian primitives, carefully considering network design and inherent real-time performance during rendering.

In contrast to the generalizable setting, several works have explored text-based panoramic view generation, albeit with distinct objectives. Nonetheless, these approaches often share similar network design philosophies. For example, Text2Light [7] focuses on high dynamic range panorama generation, while FastScene [26] employs coarse view synthesis followed by progressive inpainting-based generation. DiffPano [48] introduces a spherical epipolar-aware diffusion model for panorama synthesis. Additionally, Panfusion [51], Dreamscene360 [53], and SceneDreamer360 [21] propose novel architectures built on 2D diffusion models. Although these works are orthogonal to ours, they share several challenges in handling panoramic view synthesis.

Concurrent work. While preparing our work, we found several concurrent works address similar challenges using diffusion priors. For instance, ViewCrafter [50], GaussianEnhancer [23], ReconX [22], and FreeVS [42] all use video diffusion models to improve the novel view synthesis. However, these works require computationally intensive de-noising and are not designed for panoramic view synthesis. A very recent work, MVSplat-360 [6], focuses on synthesizing novel perspective images that enable arbitrary position and viewpoint exploration (i.e., 360° navigation) using video diffusion priors, aligning with the goals of aforementioned methods but in generalizable 3DGS setting. In contrast, our work specifically tackles generating novel views for panoramic images (i.e., 360° images).

3. Proposed Method

Our proposed framework, Splatter-360, is designed to synthesize novel views from wide-baseline 360° panoramic images by leveraging the strengths of 3DGS. Unlike conventional 3DGS methods [5, 46], which are trained on perspective images, Splatter-360 operates directly on panoramic images, thereby mitigating the information loss associated with perspective transformation. The overall pipeline is depicted in Fig. 1. A detailed explanation of the 3D-aware bi-projection encoding is provided in Sec. 3.1, spherical depth estimation is discussed in Sec. 3.2, and the pixel-aligned Gaussian decoding is outlined in Sec. 3.3. Additionally, preliminary on 3DGS is included in the supplementary material to ensure this paper is self-contained.

3.1. 3D-Aware Bi-Projection Encoding

Although equirectangular projection (ERP) provides a wide field of view and seamless image continuity, the significant distortions, especially near the poles, hinder the network's ability to learn meaningful features effectively. To address this, we propose a bi-projection encoder that leverages both the advantages of ERP and the complementary strengths of cube-map projection (CP). By extracting auxiliary features through CP, our approach enhances the network's capacity to handle these distortions better.

The overall structure of the bi-projection encoder is illustrated on the left side of Fig. 1. In each branch of the bi-projection network, we begin by applying the convolutional neural network from Unimatch [44] to independently extract local features from both the ERP and CP representations. Unlike prior panorama feature extraction methods [16, 38], which primarily focus on 2D features, our network is designed to learn 3D-aware features. To achieve this, we employ a cross-view attention mechanism to facilitate feature interactions between different viewpoints in the ERP and across the various views in the CP. We obtained $\boldsymbol{F}_{ERP}$ and $\boldsymbol{F}_{CP}$ after the cross-view attention module. $\boldsymbol{F}_{CP}$ is then stitched up into ERP feature $\boldsymbol{F}_{C2E}$.

For wide-baseline panoramas, naive 360° multi-view matching methods are hard to handle occlusions and textureless areas. To enhance the 3D-aware capability of the encoder in reasoning about geometric details, we introduce the CP single-view depth encoder. This is accomplished by incorporating the single-view geometric priors from a pre-trained monocular depth network into the CP branch at the feature level. Specifically, we extract monocular depth features from each perspective view using a pre-trained depth estimation network DepthAnythingV2 [47]. Once the monocular depth feature maps $\boldsymbol{F}_{CP}^{mono}$ for all perspective views are obtained, we stitch these CP features into the ERP view, denoted as $\boldsymbol{F}_{C2E}^{mono}$. Next, we concatenate $\boldsymbol{F}_{C2E}^{mono}$ with the transformed CP feature $\boldsymbol{F}_{C2E}$, and use a two-layer MLP to get the final CP branch feature:

$$\boldsymbol{F}'_{C2E} = \mathcal{F}_1([\boldsymbol{F}_{C2E}^{mono}, \boldsymbol{F}_{C2E}]), \tag{1}$$

where $[.,.]$ denotes the concatenation operation and $\mathcal{F}_1$ is a two-layer MLP. Then, we propose a fusion module, $\mathcal{F}_2$, to fuse the CP branch feature $\boldsymbol{F}'_{C2E}$ and ERP branch feature $\boldsymbol{F}_{ERP}$. The final output feature of our encoder is represented as:

$$\boldsymbol{F} = \mathcal{F}_2(\boldsymbol{F}'_{C2E}, \boldsymbol{F}_{ERP}), \tag{2}$$

where $\mathcal{F}_2$ is the proposed fusion module, which consists of convolutional layers augmented with squeeze-and-excitation blocks [14, 16].

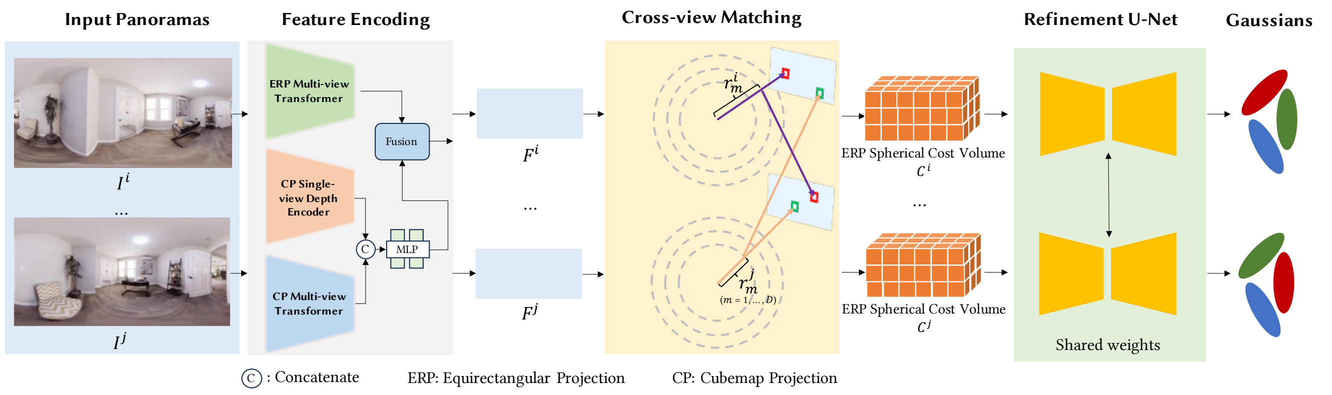

Figure 1. Our Splatter-360 processes 360° panoramic images using a bi-projection encoder that extracts features from both equirectangular projection (ERP) and cube-map projection (CP) through multi-view transformers. These features are used for spherical cost volume construction, and multi-view matching is performed between the reference and source views in spherical space. Next, a refinement U-Net is applied to enhance the spherical cost volume, yielding refined cost volumes and more accurate spherical depth estimations. These refined outputs are then fed into the Gaussian decoder, which produces pixel-aligned Gaussian primitives for synthesizing novel views.

3.2. Spherical Depth Estimation

After obtaining a robust feature representation for panoramic images, the next step is cost volume construction. One straightforward approach is to convert the panorama into cube-map-like perspective images and then follow the perspective-based methods such as MVSplat [5] and DepthSplat [46] to construct cost volume. However, such a method has an unavoidable shortcoming: for each 3D sampling point in the planar cost volume of the reference view, these points may project behind the camera in the source view (i.e., z-depth < 0). When this occurs, incorrect local image features from the source view are sampled, leading to inaccurate local similarity computations and, ultimately, erroneous depth estimates. This issue is a consequence of the narrow field-of-view plane sweep algorithm traditionally used for the perspective-based method.

To address this, we leverage the 360-degree field of view inherent to panoramas and apply the spherical projection formula derived from [48] to construct a spherical cost volume. More concretely, the spherical sweep algorithm samples depth candidates within the spherical domain, computing feature correlations across multiple views to construct the cost volume. For each reference view, we sample D depth candidates in the spherical space, where the depth values $r_m \in [r_{\text{near}}, r_{\text{far}}]$ are sampled in the logarithmic space between the near range r_{near} and the far range r_{far}.

To transform the equirectangular coordinates to spherical coordinates, we use the following equations:

$$\begin{cases} \theta = (0.5 - \dfrac{u}{W}) \cdot 2\pi \\ \phi = (0.5 - \dfrac{v}{H}) \cdot \pi, \end{cases} \tag{3}$$

where u and v represent the pixel coordinates in the equirectangular grid, while θ and ϕ denote the longitude and latitude, respectively. Additionally, W and H refer to the width and height of the ERP feature maps on which multi-view matching is performed.

Next, we transform the spherical polar coordinates (r, θ, ϕ) into Cartesian coordinates (x, y, z) to obtain the camera coordinates $\boldsymbol{p}^i_{\text{camera}}$

$$\begin{cases} x_{cam} = r\cos(\phi) \cdot \sin(\theta) \\ y_{cam} = r\sin(\phi) \\ z_{cam} = r\cos(\phi) \cdot \cos(\theta), \end{cases} \tag{4}$$

To obtain the pixel coordinates (u, v) at the source view j, we back-project the camera coordinates using the relative camera pose $W^{i\to j}$ between the reference view and the source view:

$$\boldsymbol{p}^j_{camera} = W^{i\to j}\boldsymbol{p}^i_{camera}$$

With known corresponding pixel coordinates, we sample the local image feature $F^{j\to i}$ from the source view feature and compute the feature similarity between $F^{j\to i}$ and F^i as follows:

$$\boldsymbol{C}^i_{r_m} = \frac{\boldsymbol{F}^i \cdot \boldsymbol{F}^{j\to i}_{r_m}}{\sqrt{C}} \in \mathbb{R}^{H\times W}, \quad m = 1, 2, \cdots, D, \tag{5}$$

where C denotes the total feature channel and $C^i_{r_m}$ denotes the feature similarity between $F^{j\to i}$ and F^i at the m^{th} depth candidate. Then, we can concatenate each depth candidates' feature similarity to get the spherical cost volume.

$$\boldsymbol{C}^i = [\boldsymbol{C}^i_{r_1}, \boldsymbol{C}^i_{r_2}, \cdots, \boldsymbol{C}^i_{r_D}] \in \mathbb{R}^{H\times W\times D}. \tag{6}$$

As the initial spherical cost volume only considers the pixel-level similarity and is hard to handle occlusions and textureless areas. We further use a U-Net [5] to refine it. Specifically, the proposed U-Net mainly learns a residual volume value $\Delta \boldsymbol{C}^i$, which can be added to the initial spherical cost volume to get the final one:

$$\tilde{\boldsymbol{C}}^i = \boldsymbol{C}^i + \Delta \boldsymbol{C}^i \in \mathbb{R}^{H \times W \times D}. \tag{7}$$

Finally, we can use the softmax operation to get the final spherical depth estimation:

$$\boldsymbol{D}^i = \text{softmax}(\tilde{\boldsymbol{C}}^i)\boldsymbol{G} \in \mathbb{R}^{H \times W}, \tag{8}$$

where $\boldsymbol{G} = [r_1, r_2, \ldots, r_D] \in \mathbb{R}^D$ is the predefined depth candidates and $softmax$ denotes the softmax function which can transform spherical cost volume to the probability distribution for each depth candidates.

3.3. Pixel-Aligned Gaussian in Equirect-Coordinate

Once the 3D-aware feature representation and spherical cost volume are obtained, we proceed to predict pixel-aligned Gaussians on the equirectangular coordinate grid. Specifically, for each pixel, we estimate Gaussian centers, quaternions, spherical harmonics (SH), and scaling factors, respectively.

Gaussian centers $\boldsymbol{\mu}$. We use the estimated spherical depth $\boldsymbol{D}$ to calculate the gaussian centers. Specifically, for each pixel, we first compute the camera cartesian coordinates $\boldsymbol{p}_{\text{camera}}$ of the gaussian center using its spherical depth $\boldsymbol{D}$ via Eq. 3 and Eq. 4. Next, the world coordinates of the Gaussian center are calculated as $\boldsymbol{p}_{\text{world}} = W\boldsymbol{p}_{\text{camera}}$, where W represents the camera-to-world transformation matrix.

Opacity α. The opacity is calculated according to the matching confidence. Specifically, we first use the probability distribution of spherical cost volume(i.e., the softmax output of Eq. 8) to get the matching confidence. Then, we feed the matching confidence along with the image features into a two-layer convolutional network to predict the opacity.

Covariance $\boldsymbol{\Sigma}$ and SH $\boldsymbol{c}$. We predict these parameters using two convolutional layers, which take as input the concatenation of image features, the refined cost volume, and the original ERP images. The covariance matrix Σ is computed as:

$$\Sigma = R(\theta)^\top \,\text{diag}(s)\, R(\theta),$$

following the formulation in [4]. Here, $R(\theta)$ is the rotation matrix parameterized by quaternions, s represents the scaling matrix, and $\boldsymbol{c}$ is the spherical harmonics for direction-dependent color encoding.

4. Experiments

4.1. Implementation Details

We implement our proposed method using PyTorch and conduct all experiments on a cluster of NVIDIA V100 GPUs, each equipped with 32GB of VRAM. Splatter-360 is trained with a loss function that is a linear combination of mean squared error (MSE) and LPIPS loss, with weights set to 1.0 and 0.05, respectively. More details about the implementation are presented in the supplementary material.

4.2. Datasets, Baselines, and Metrics

Datasets. We evaluate Splatter-360 on two large-scale panoramic datasets: HM3D [27] and Replica [32]. These datasets contain diverse indoor scenes captured with 360° panoramic cameras, providing a challenging benchmark for wide-baseline novel view synthesis.

Baselines. To proveide a fair comparisions both quantitatively and qualitatively, we compare our Splatter-360 with several state-of-the-art generalizable 360-degree method PanoGRF [8] and generalizable perspective methods including HiSplat [35], DepthSplat [46] and MVSplat [5].

Metrics. We evaluate the performance of Splatter-360 using standard metrics for novel view synthesis, including PSNR, SSIM, and LPIPS.

4.3. Quantitative Results

Our results demonstrate that Splatter-360 outperforms existing methods in both synthesis quality and generalization to novel views. As for the perspective-based method, we convert panoramas to cube maps (12 perspective images) and input these cube maps into HiSplat, DepthSplat, and MVSplat. We did not compare with PixelSplat [4] due to out-of-memory errors during inference with 12 perspective image views. Furthermore, HiSplat and DepthSplat cannot be trained on the HM3D dataset with 12 perspective images due to GPU memory constraints; therefore, we tested their official pre-trained models on Re10K [54] for evaluation on both HM3D and Replica. MVSplat and Splatter-360 are trained on HM3D using 8 Tesla V100 GPUs.

Table 1 shows that existing perspective-based methods, including HiSplat, DepthSplat, and MVSplat, struggle with generalization to cube-map inputs and cannot be directly applied to panoramic datasets. We tested the fine-tuned version of MVSplat, but it still lags behind Splatter-360 in terms of generalization, both on HM3D and Replica. Specifically, the PSNR of MVSplat is 1.114 dB lower than Splatter-360 on HM3D and 1.489 dB lower on Replica. We also compared it with PanoGRF [8], specifically designed for panoramic inputs. Splatter-360 consistently outperforms PanoGRF across all metrics on Replica and HM3D. On HM3D, the PSNR of Splatter-360 is 2.662 dB higher than PanoGRF, and on Replica, it is 1.968 dB higher.

Moreover, Table 2 provides a quantitative comparison of novel-view depth estimation between MVSplat and Splatter-360. Splatter-360 consistently outperforms MVSplat across all depth metrics on HM3D and Replica, verifying that the proposed method can better capture geometry for panoramic inputs.

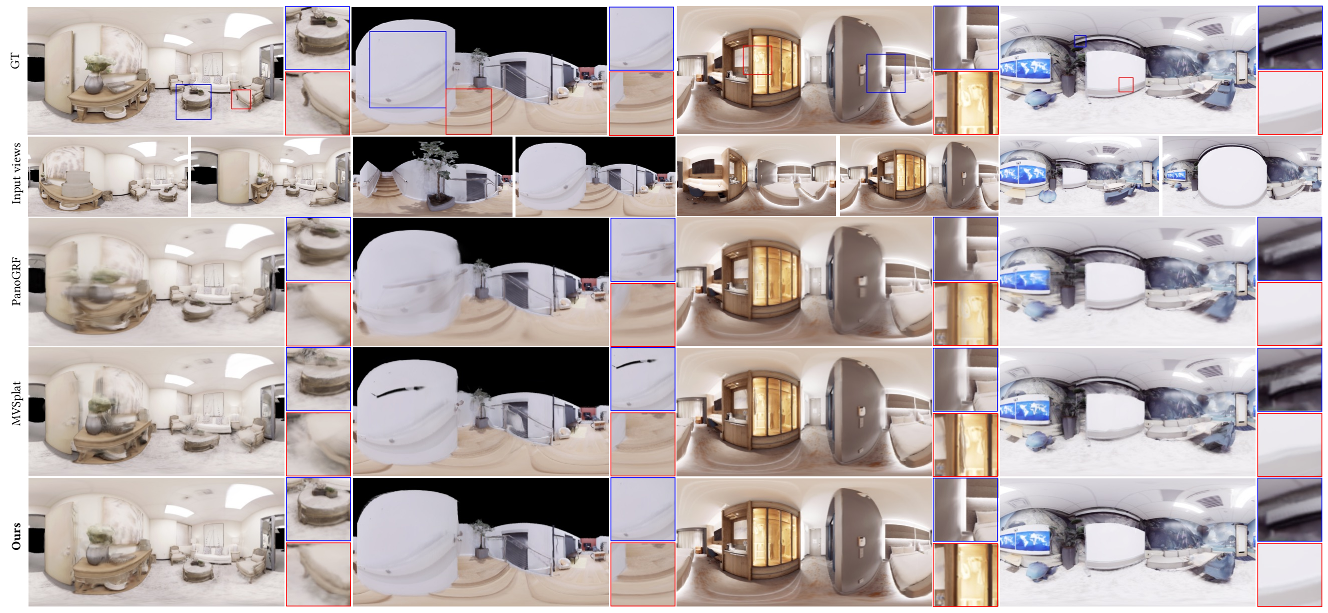

Figure 2. Qualitative comparison between our Splatter-360 and PanoGRF, MVSplat on the Replica dataset. Regions with notable differences are highlighted using red and blue rectangles. Please zoom in for a clearer view.

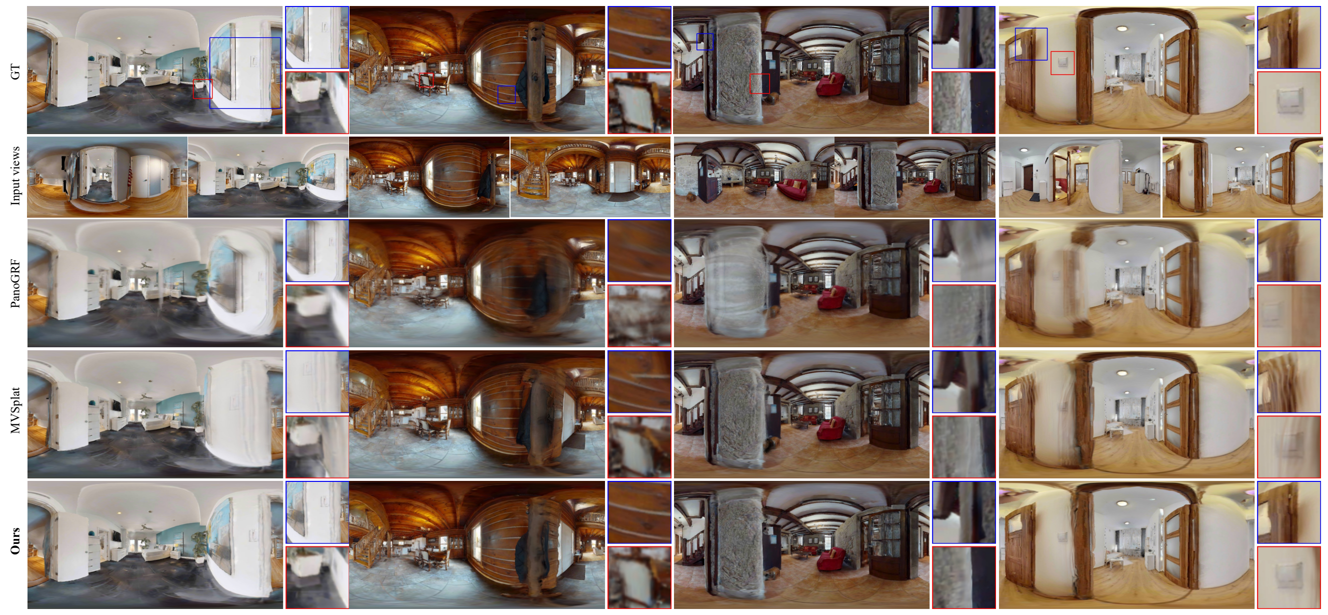

Figure 3. Qualitative comparison between our Splatter-360 and PanoGRF, MVSplat on the HM3D dataset. Regions with notable differences are highlighted using red and blue rectangles. Please zoom in for a clearer view.

4.4. Qualitative Results

Qualitative comparisons are provided in Fig. 3 and Fig. 2, with key differences highlighted using red and blue rectangles. As shown in the first sample of Fig. 2, Splatter-360 produces much sharper edges and more accurate textures, particularly on objects like the table and chair. In the second and third samples, PanoGRF demonstrates less defined edges, particularly on the floor, while MVSplat exhibits noticeable artifacts in the form of floaters near the white wall.

A comparison of learned geometry between Splatter-360 and MVSplat is presented in Fig. 4. MVSplat shows significantly poorer depth estimation, as evidenced by its incorrect reconstruction of surfaces such as the lamp and table. In contrast, Splatter-360 delivers more accurate and detailed

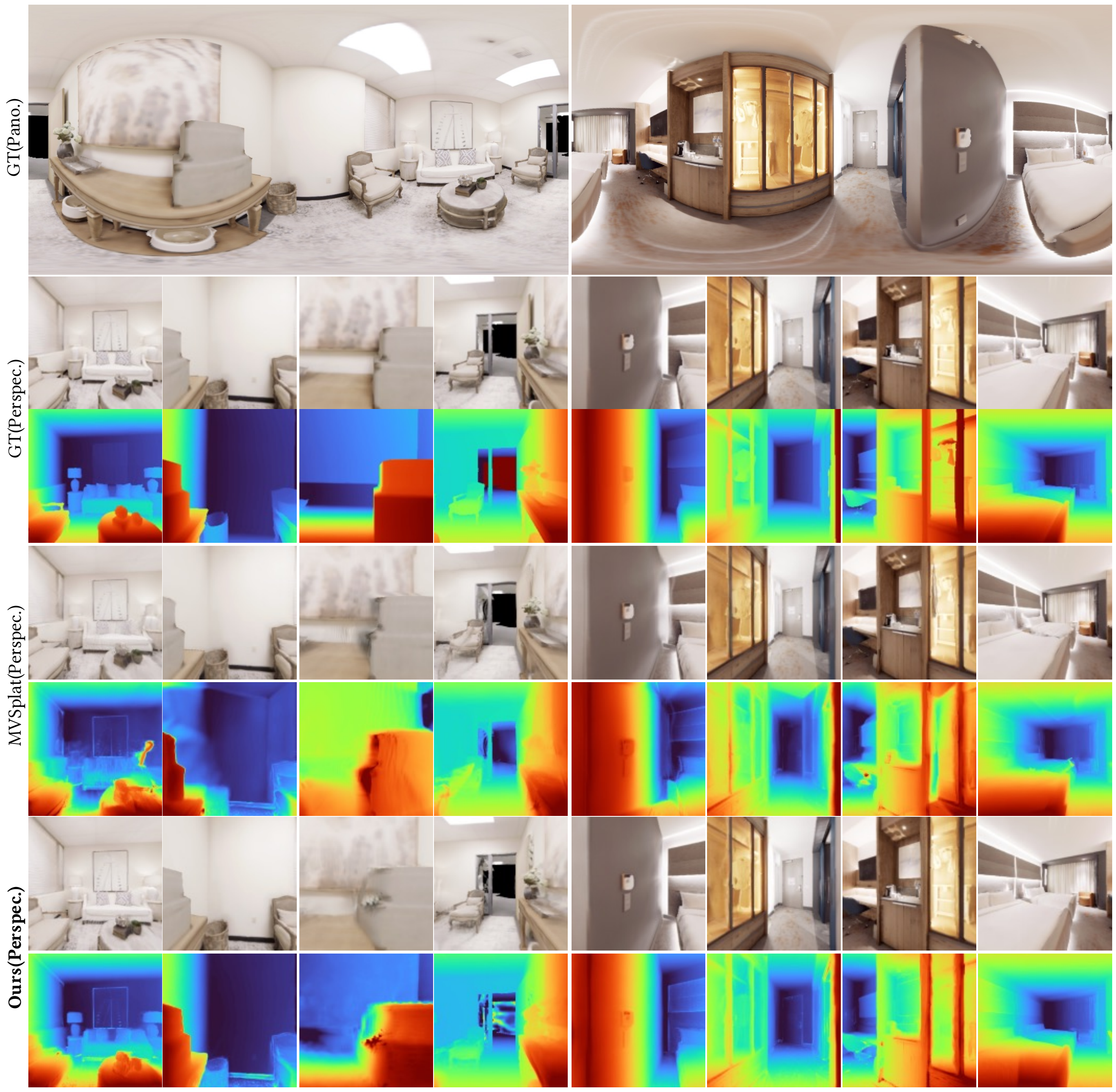

Figure 4. Novel view depth comparison between Splatter-360 and PanoGRF on the Replica dataset. "Pano." denotes panoramic view and "Perspec." denotes perspective view.

depth predictions with clearer geometry and well-defined surfaces.

4.5. Ablation Study

We validate the effectiveness of different modules proposed in Splatter-360 in this section. Due to the large number of ablation studies and limited resources, we utilized 2 NVIDIA Tesla V100 GPUs for each group of experiments in this section.

Spherical cost volume. Eliminating the spherical cost volume leads to a notable decline in performance across all metrics, highlighting its crucial role in the model's ability to generate high-quality reconstructions. Specifically, as shown in the first row of Table 3 , compared to the full model, the PSNR drops by 5.271 dB and 2.263 dB on the Replica and HM3D datasets, respectively.

ERP encoder vs. CP encoder. We separately removed

Table 1. Quantitative comparison with baseline methods on the HM3D and Replica datasets. † indicates models that were trained by us on the panoramic dataset, whereas for all other methods, we used the pre-trained models provided by the original authors.

	HM3D [27]			**Replica** [32]		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HiSplat [35]	17.268	0.624	0.488	17.157	0.642	0.417
MVSplat [5]	17.574	0.636	0.441	18.005	0.631	0.512
DepthSplat [46]	20.224	0.695	0.383	19.369	0.732	0.334
PanoGRF [8]	25.631	0.813	0.268	27.920	0.892	0.171
MVSplat† [5]	27.179	0.851	0.176	28.399	0.908	0.115
Splatter-360†	**28.293**	**0.875**	**0.155**	**29.888**	**0.924**	**0.097**

Table 2. Estimated depth comparison between MVSplat and Splatter-360 on the Replica and HM3D datasets.

Dataset	**Metric**	**MVSplat**	**Splatter-360**
Replica [32]	Abs Diff↓	0.132	**0.102**
	Abs Rel↓	0.088	**0.063**
	RMSE↓	0.247	**0.197**
	$\delta < 1.25$↑	89.913	**94.572**
HM3D [27]	Abs Diff↓	0.130	**0.106**
	Abs Rel↓	0.094	**0.076**
	RMSE↓	0.271	**0.223**
	$\delta < 1.25$↑	90.469	**93.851**

Table 3. Ablation studies were conducted on the HM3D and Replica datasets. For simplicity, we use the following abbreviations: 'SCV' for spherical cost volume and 'CVA' for cross-view attention.

Ablated	Replica [32]			HM3D [27]		
module	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
× SCV	23.850	0.818	0.210	25.224	0.802	0.223
× CVA	28.217	0.905	0.124	26.918	0.851	0.182
× ERP	26.985	0.887	0.142	25.905	0.827	0.202
× CP	28.673	0.909	0.117	27.277	0.857	0.174
× Mono Feat.	28.654	0.911	0.116	27.380	0.858	0.173
Full	**29.121**	**0.914**	**0.111**	**27.487**	**0.860**	**0.171**

the ERP and CP encoders to evaluate their contributions. The results of these ablations are presented in the third and fourth rows of Table 3. The ERP encoder is crucial for maintaining high PSNR and SSIM values and achieving low LPIPS values, highlighting its vital role in panoramic feature extraction and encoding. In contrast, the removal of the CP encoder results in a smaller performance drop compared to the removal of the ERP encoder. However, removing the CP encoder on the Replica dataset still leads to a PSNR decrease of approximately 0.448 dB, suggesting that the CP encoder provides auxiliary support in the Gaussian reconstruction of panoramic images.

Cross-view attention. Corss-view attention is essential to extract the 3D-awareness feature. Here, we test the influence of removing this module. As shown in the second row of Table 3 , such a removing resulted in a noticeable decline in PSNR and SSIM, alongside an increase in LPIPS, indicating the effectiveness of cross-view attention.

Monocular depth network feature. Removing the monocular depth encoder led to a PSNR drop of 0.467 dB in the Replica dataset. This suggests that the pre-trained monocular depth features offer a robust single-view geometry-aware prior, which is especially advantageous for reconstructing 360-degree sparse views in textureless indoor scenes.

5. Conclusion

This paper introduced Splatter-360, a novel framework for generalizable 3D Gaussian Splatting designed for wide-baseline panoramic images. By leveraging 3D-Aware Bi-Projection Encoding and spherical cost volume, our method significantly improves the quality of novel view synthesis and geometry estimation from panoramic inputs. Experimental results on the HM3D and Replica datasets demonstrate that Splatter-360 sets a new state-of-the-art performance in the panoramic setting, making it a promising solution for applications in VR, simulation rendering, and 360° video streaming.

Limitations and future work. Our method achieves state-of-the-art performance on several benchmarks but shares some limitations with existing approaches. It requires pose input, lacks generative capabilities, and is currently limited to indoor scenes due to the absence of a large-scale 360° outdoor dataset. In the future, we plan to create a comprehensive synthetic outdoor dataset and enhance our method with a video diffusion model. More importantly, we aim to explore pose-free 360° reconstruction in diverse outdoor environments.

References

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *European Conference on Computer Vision*, pages 441–459. Springer, 2020. 2

[2] Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li, and Yanwen Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. *arXiv preprint arXiv:2402.00763*, 2024. 2

[3] Wenjie Chang, Yueyi Zhang, and Zhiwei Xiong. Depth estimation from indoor panoramas with neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2023. 2

[4] 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, 5

[5] 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. *CoRR*, abs/2403.14627, 2024. 1, 2, 3, 4, 5, 8

[6] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3

[7] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3

[8] Zheng Chen, Yan-Pei Cao, Yuan-Chen Guo, Chen Wang, Ying Shan, and Song-Hai Zhang. Panogrf: generalizable spherical radiance fields for wide-baseline panoramas. *Advances in Neural Information Processing Systems*, 36:6961–6985, 2023. 1, 3, 5, 8

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[10] 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

[11] Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. Omni-nerf: neural radiance field from 360 image captures. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 2

[12] Tewodros Habtegebrial, Christiano Gava, Marcel Rogge, Didier Stricker, and Varun Jampani. Somsi: Spherical novel view synthesis with soft occlusion multi-sphere images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15725–15734, 2022. 2

[13] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[15] Huajian Huang, Yingshu Chen, Tianjian Zhang, and Sai-Kit Yeung. 360Roam: Real-Time Indoor Roaming Using Geometry-Aware 360° Radiance Fields. *arXiv preprint arXiv:2208.02705*, 2022. 2

[16] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360 panorama depth estimation. *IEEE Robotics and Automation Letters*, 6 (2):1519–1526, 2021. 3

[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

[18] Shreyas Kulkarni, Peng Yin, and Sebastian Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7202–7209. IEEE, 2023. 2

[19] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. *arXiv preprint arXiv:2403.10147*, 2024. 1

[20] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 2

[21] Wenrui Li, Yapeng Mi, Fucheng Cai, Zhe Yang, Wangmeng Zuo, Xingtao Wang, and Xiaopeng Fan. Scenedreamer360: Text-driven 3d-consistent scene generation with panoramic gaussian splatting. *arXiv preprint arXiv:2408.13711*, 2024. 3

[22] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024. 3

[23] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. *arXiv preprint arXiv:2410.16266*, 2024. 3

[24] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 2

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[26] Yikun Ma, Dandan Zhan, and Zhi Jin. Fastscene: Text-driven fast 3d indoor scene generation via panoramic gaussian splatting. *arXiv preprint arXiv:2405.05768*, 2024. 3

[27] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 1, 2, 5, 8

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[29] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[30] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 1

[31] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[32] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 2, 5, 8

[33] 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

[34] 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

[35] Shengji Tang, Weicai Ye, Peng Ye, Weihao Lin, Yang Zhou, Tao Chen, and Wanli Ouyang. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. *arXiv preprint arXiv:2410.06245*, 2024. 2, 5, 8

[36] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021. 1, 2

[37] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023. 2

[38] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020. 3

[39] Guangcong Wang, Peng Wang, Zhaoxi Chen, Wenping Wang, Chen Change Loy, and Ziwei Liu. Perf: Panoramic neural radiance field from a single panorama. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10): 6905–6918, 2024. 2

[40] Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, et al. Is attention all that nerf needs? *arXiv preprint arXiv:2207.13298*, 2022. 2

[41] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 1, 2

[42] Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Freevs: Generative view synthesis on free driving trajectory. *arXiv preprint arXiv:2410.18079*, 2024. 3

[43] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21551–21561, 2024. 2

[44] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3, 2

[45] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024. 2

[46] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 2, 3, 4, 5, 8

[47] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 3

[48] Weicai Ye, Chenhao Ji, Zheng Chen, Junyao Gao, Xiaoshui Huang, Song-Hai Zhang, Wanli Ouyang, Tong He, Cairong Zhao, and Guofeng Zhang. Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. *arXiv preprint arXiv:2410.24203*, 2024. 3, 4

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 1, 2

[50] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 3

[51] Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming stable diffusion for text to 360 {\deg} panorama image generation. *arXiv preprint arXiv:2404.07949*, 2024. 3

[52] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 2

[53] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*, pages 324–342. Springer, 2025. 3

[54] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4): 65, 2018. 5, 2

Splatter-360: Generalizable 360° Gaussian Splatting for Wide-baseline Panoramic Images

Supplementary Material

6. Additional Quantitative Results

6.1. Comparisons with More Input Views

Table 4 presents a quantitative comparison of MVSplat and Splatter-360 using three-view inputs. Splatter-360 demonstrates superior performance to MVSplat in SSIM and LPIPS, while exhibiting comparable PSNR values. Additionally, we evaluate the estimated novel view depth using depth metrics used in [30]. Splatter-360 significantly outperforms MVSplat across all the used depth metrics. Our Splatter-360 exhibits robust general performance with three-view inputs, despite being trained on two-view inputs.

6.2. Comparisons under a Narrow-baseline

Table 5 presents a quantitative comparison between MVSplat and Splatter-360 under the narrow-baseline setting. In the main text, we sample an input pair with a frame interval of 100. Here, the frame interval of the input pair is further reduced to 50. Splatter-360 consistently outperforms MVSplat across all metrics. This result indicates that Splatter-360 exhibits superior performance under the narrow-baseline condition.

6.3. Additional Ablation Studies

We conduct additional ablation studies on the cost volume refinement U-Net and the depth refinement U-Net. Table 6 presents the statistical results obtained after removing these modules individually. In comparison to the complete model using all components, the model without depth refinement U-Net exhibits significantly degraded PSNR performance. Specifically, PSNR decreased by approximately 0.7 dB on Replica and by about 0.69 dB on HM3D. Upon removing the cost volume refinement U-Net, PSNR decreased by 0.119 dB on Replica and by 0.125 dB on HM3D.

7. More Implementation Details

7.1. Dataset Details

The datasets are built based on Replica [32] and HM3D [27] textured mesh dataset. In particular, we sample camera trajectories to render videos with AI-Habitat simulation tool [29]. Since AI-habitat only provides the API for capturing perspective views, we first get cube maps for each viewpoint and stitch them into panoramas.

For HM3D [27], we split the train and test set following their original split. HM3D contains 800 training scenes and 100 test scenes. We sample 5 camera trajectories for each scene. In total, we finally rendered 4000 training scenes and 500 test scenes.

For Replica, we use all the scenes for testing. Replica has 18 scenes in total, and we sample 5 camera trajectories for each scene. In total, we render 90 test scenes. We randomly sample 3 target views between the context image pair for

Table 4. Quantitative comparison with three context views between MVSplat and Splatter-360 on the Replica and HM3D datasets.

Dataset	Metric	MVSplat	Splatter-360
Replica [32]	PSNR↑	**29.121**	29.109
	SSIM↑	0.908	**0.913**
	LPIPS↓	0.123	**0.116**
	Abs Diff↓	0.125	**0.103**
	Abs Rel↓	0.078	**0.060**
	RMSE↓	0.233	**0.193**
	$\delta < 1.25$↑	90.771	**94.367**
HM3D [27]	PSNR↑	27.858	**27.905**
	SSIM↑	0.861	**0.868**
	LPIPS↓	0.174	**0.168**
	Abs Diff↓	0.118	**0.095**
	Abs Rel↓	0.083	**0.067**
	RMSE↓	0.251	**0.209**
	$\delta < 1.25$↑	91.684	**94.545**

Table 5. Quantitative comparison under a narrow baseline between MVSplat and Splatter-360 on the Replica and HM3D datasets.

Dataset	Metric	MVSplat	Splatter-360
Replica [32]	PSNR↑	32.521	**33.282**
	SSIM↑	0.951	**0.957**
	LPIPS↓	0.064	**0.058**
	Abs Diff↓	0.109	**0.090**
	Abs Rel↓	0.057	**0.048**
	RMSE↓	0.214	**0.171**
	$\delta < 1.25$↑	94.257	**96.645**
HM3D [27]	PSNR↑	30.851	**31.493**
	SSIM↑	0.915	**0.925**
	LPIPS↓	0.109	**0.101**
	Abs Diff↓	0.102	**0.092**
	Abs Rel↓	0.060	**0.058**
	RMSE↓	0.228	**0.189**
	$\delta < 1.25$↑	94.802	**96.031**

Table 6. Additional ablation studies were conducted on the HM3D and Replica datasets. For simplicity, we use the following abbreviations: 'CVRU' for spherical cost volume refinement U-Net and 'DRU' for depth refinement U-Net.

Ablated	Replica [32]			HM3D [27]		
module	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
× DRU	28.306	0.905	0.129	26.800	0.846	0.189
× CVRU	29.002	0.912	0.115	27.362	0.856	0.175
Full	**29.121**	**0.914**	**0.111**	**27.487**	**0.860**	**0.171**

testing.

7.2. Experiment Details

In the comparisons of the main paper, for HiSplat, MVSplat, and DepthSplat, we utilize their model pre-trained models on RE10K [54] for evaluation. We set $near = 0.5$ and $far = 10$ for these models as these parameters are relatively close to their training setting. We match features under the resolution of $\frac{1}{4}H \times \frac{1}{4}W$, where H and W are the height and width of input images. We apply $near = 0.1$ for HiSplat, MVSplat, and DepthSplat, but the results get worse as $near = 0.1$ is much different from their training setting on RE10K [54].

For MVSplat[†] and Splatter-360[†] trained on HM3D, we set $near = 0.1$ and $far = 10$ to match our indoor dataset for the consideration of fairness. We perform cross view matching under the resolution of $\frac{1}{8}H \times \frac{1}{8}W$ due to GPU memory limits.

7.3. Network Details

We adopt the encoder of UniMatch [44] as our backbone. The first convolution layer downsamples images with a stride of 2. Next, we utilize six residual layers to extract features. The first two residual layers contain utilize the stride of 1. Subsequently, we downsample features in half after every two residual layers with a stride of 2. We then get $\frac{1}{8}H \times \frac{1}{8}W$ feature maps. The downsampled feature maps are fed into a cross-view transformer composed of six stacked transformer blocks. Each transformer block contains a self-attention and a cross-view attention layer. Similar to MVSplat [5], we utilize the local window attention of SwinTransformer [25]. We apply the network architecture for our ERP multi-view transformer and CP multi-view transformer.

For the cost volume refinement U-UNet, we adopt the U-Net from Stable Diffusion 1.5 [28] as our implementation with an unchanged feature channel of 128 throughout the network. We apply two times $2\times$ down-sampling and one self-attention layer at the $4\times$ down-sampled level. We flatten the feature map before feeding them to the attention module, to interact with the features among different views utilizing the multi-view attention similar to [31]. For the depth refinement U-Net which we omitted in the main text for simplicity, we apply 4 times $2\times$ down-sampling and add the multi-view attention at $16\times$ down-sampled level.

We set $D = 128$ in the depth sampling consistently with MVSplat [5]

8. Preliminary of 3DGS

The 3D Gaussian ellipsoid is formally defined as:

$$G(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{9}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ represents the spatial mean, and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3}$ denotes the covariance matrix. To ensure numerical stability during optimization, the covariance matrix $\boldsymbol{\Sigma}$ is decomposed into a scaling matrix $\boldsymbol{S}$ and a rotation matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top \boldsymbol{R}^\top \tag{10}$$

During the rendering process, the 3D Gaussians are projected onto a 2D image plane. Using the intrinsic matrix $\boldsymbol{K}$ and extrinsic matrix $\boldsymbol{T}$, the 2D mean $\boldsymbol{\mu}'$ and covariance matrix $\boldsymbol{\Sigma}'$ are computed as:

$$\boldsymbol{\mu}' = \boldsymbol{K}[\boldsymbol{\mu}, 1]^\top, \quad \boldsymbol{\Sigma}' = \boldsymbol{J}\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^\top \boldsymbol{J}^\top \tag{11}$$

Here, $\boldsymbol{J}$ represents the Jacobian matrix of the affine approximation of the projective transformation. Each Gaussian is associated with an opacity value o and a view-dependent color $\boldsymbol{c}$, which is determined by a set of spherical harmonics coefficients. The pixel color $\boldsymbol{C}$ is computed via alpha-blending over the 2D Gaussians, sorted from front to back:

$$\boldsymbol{C} = \sum_{i \in N} T_i G_i \left(\boldsymbol{u} \mid \boldsymbol{\mu}', \boldsymbol{\Sigma}'\right) \sigma_i \boldsymbol{c}_i \tag{12}$$

where the transmittance T_i is defined as:

$$T_i = \prod_{j=1}^{i-1} \left(1 - G_i \left(\boldsymbol{u} \mid \boldsymbol{\mu}', \boldsymbol{\Sigma}'\right) \sigma_i\right) \tag{13}$$