360-GS: Layout-guided Panoramic Gaussian Splatting For Indoor Roaming

Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li and Yanwen Guo
Nanjing University

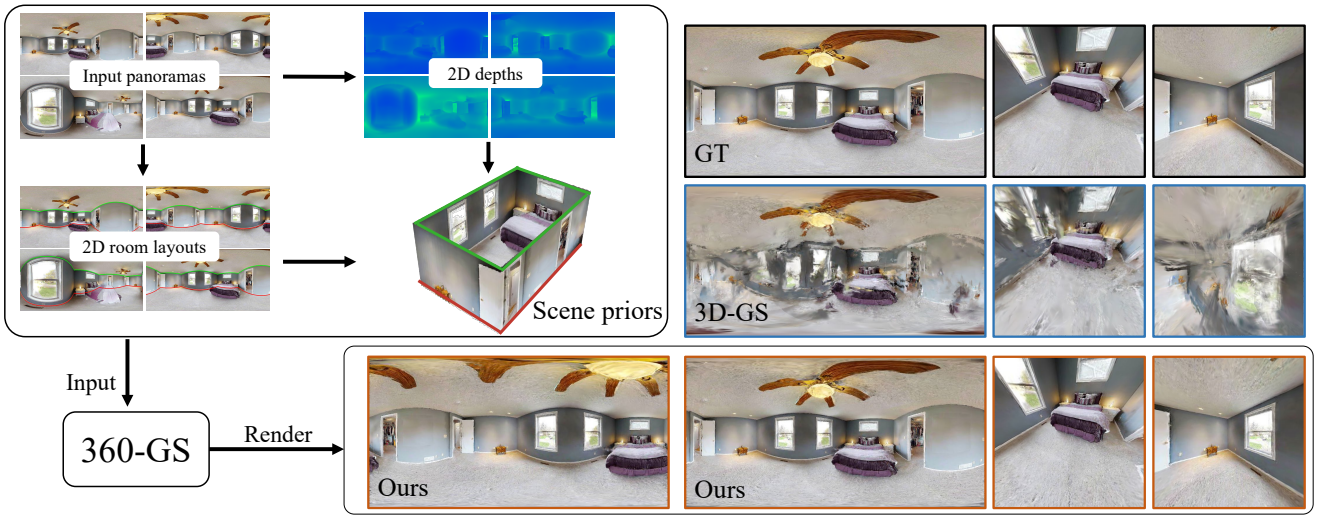


Figure 1. Taking four indoor panoramas as input, our method optimizes 3D Gaussians under the guidance of scene priors including room layouts and depths. Leveraging our 360° splatting algorithm, we are able to render high-quality equirectangular images from the refined 3D Gaussians. The visual comparisons illustrate that our method surpasses 3D-GS [18] with accurate geometry and plausible details.

Abstract

3D Gaussian Splatting (3D-GS) has recently attracted great attention with real-time and photo-realistic renderings. This technique typically takes perspective images as input and optimizes a set of 3D elliptical Gaussians by splatting them onto the image planes, resulting in 2D Gaussians. However, applying 3D-GS to panoramic inputs presents challenges in effectively modeling the projection onto the spherical surface of 360° *images using 2D Gaussians. In practical applications, input panoramas are often sparse, leading to unreliable initialization of 3D Gaussians and subsequent degradation of 3D-GS quality. In addition, due to the under-constrained geometry of texture-less planes (e.g., walls and floors), 3D-GS struggles to model these flat regions with elliptical Gaussians, resulting in significant floaters in novel views. To address these issues, we propose 360-GS, a novel layout-guided* 360° *Gaussian splatting for a limited set of panoramic inputs. Instead of splatting 3D Gaussians directly onto the spherical surface, 360-GS projects them onto the tangent plane of the unit sphere and then maps them to the spherical projections. This adaptation enables the representation of the projection using Gaussians. We guide the optimization of 3D Gaussians by exploiting layout priors within panoramas, which are simple to obtain and contain strong structural information about the indoor scene. Our experimental results demonstrate that 360-GS allows panoramic rendering and outperforms state-of-the-art methods with fewer artifacts in novel view synthesis, thus providing immersive roaming in indoor scenarios.*

1. Introduction

With the popularity of consumer-level 360° cameras, novel view synthesis from a set of panoramic images has been one of the core components of computer graphics and vision applications, including virtual and augmented reality (VR/AR). Recently, Neural Radiance Fields (NeRF) [1, 3, 20, 39] have attracted great attention due to their ability to produce photo-realistic renderings and become a widely used technique to synthesize novel views. However, NeRF samples dense points for each pixel, making real-time ren-

dering challenging. Recently, point-based representation, 3D Gaussian Splatting (3D-GS) [18], has emerged as an alternative representation that achieves real-time speed with comparable rendering quality to NeRF-based methods. This enables real-time indoor room roaming and has practical applications like free-viewpoint navigation, house touring, and virtual-reality games.

However, 3D-GS mainly focuses on perspective images. When given a set of indoor panoramas, synthesizing novel views with 3D-GS encounters several challenges. *First*, splatting 3D Gaussians onto panoramic images has spatial distortion that can not be modeled with 2D Gaussians splatted onto image planes of perspective projection. Thus it is impossible to directly optimize 3D Gaussians with panoramas. *Second*, collecting dense panoramic views of a scene is often expensive and time-consuming [10]. In a typical image collection process, the 360° camera is placed at the center or in a limited set of locations in the rooms, resulting in sparse input. This scarcity of inputs significantly exacerbates the inherent ambiguity of learning 3D structure from 2D images, thus leading to unsatisfying renderings [41, 44]. While many works have attempted to address the few-shot task by leveraging pixel-wise information such as depth supervision [8, 44] and cross-view semantic consistency [16], the scene-level structural information within panoramas remains under-exploited. *Third*, indoor scenes often contain many texture-less and flat regions such as walls, floors, and ceilings, which are insufficient for finding cross-view correspondences. Even though 3D-GS can well fit training pixels, the geometry of planes is inaccurate, leading to floaters above the planes in novel views. Previous works have tackled this problem through geometric regularization [5, 8], but most of them are built on top of NeRF.

To address the aforementioned challenges, we propose 360-GS, a novel layout-guided 3D Gaussian splatting pipeline designed for sparse panoramic images. This approach achieves real-time panoramic rendering while delivering high-quality novel views, significantly reducing undesired artifacts such as floaters, as depicted in Fig 1. The impressive performance is attributed to two core components of 360-GS: 360° Gaussian splatting and the incorporation of room layout priors. 360° Gaussian splatting algorithm decomposes the splatting into two steps: projecting 3D Gaussians onto the tangent plane and then mapping them to the spherical surface. The decomposition avoids the complicated representation of projections while maintaining real-time performance. We further address the under-constrained problem due to few-shot inputs and texture-less planes by introducing room layout priors. With a full field of view, a panorama inherently contains richer global structural information than a perspective image that can be exploited for more regularization. The room layout is the most common and easy-obtained structural information for indoor scenes. From the room layout, we derive a high-quality point cloud for the initialization of 3D Gaussians. Since the room layout describes the scene with flat walls, floors, and ceilings, we further enforce constraints on the positions of 3D Gaussians in these regions. The layout-guided initialization and regularization contribute to the generation of flat planes and a reduction in undesired floaters in novel views. The experiments conducted on real-world datasets have demonstrated the superiority and effectiveness of our method.

In summary, the main contributions of our paper are:

- We propose 360-GS, a layout-guided 3D Gaussian splatting pipeline designed for sparse panoramic images, which allows real-time panoramic rendering.
- We derive a high-quality point cloud generation method for the initialization of 3D Gaussians from room layouts to improve the performance of novel view synthesis.
- We introduce a layout-guided regularization to reduce floaters caused by under-constrained regions.

2. Related Work

2.1. Novel view synthesis

Given a dense set of calibrated images, the task of novel view synthesis aims to generate photo-realistic images of a 3D scene from unseen viewpoints. To improve the quality of the reconstructed 3D scene and novel views, some studies utilize explicit representations such as layered representations [30, 33], voxels [35], mesh [15] and point clouds [25, 40]. Recently, there has been an increasing interest in the use of volumetric representations. Neural radiance fields (NeRF) [20] employs implicit neural networks to represent scenes as continuous volumetric functions of density and color. Volumetric rendering is then employed to generate novel views. Mip-NeRF 360 [2] extends NeRF to address aliasing and model unbounded scenes using volumetric frustums along a cone and a non-linear scene parameterization, respectively. Despite the powerful neural implicit representation of NeRF, it demands significant time for training and rendering, posing a challenge for real-time applications. Recent works have strived to accelerate the rendering speed [22, 27, 38, 43]. Concurrently, another line of work employs point-based representation and rendering. 3D-GS [18] models the scene with explicit 3D Gaussians and renders 2D images using the splatting technique, elevating the photo-realistic rendering quality to real-time levels.

While existing methods for novel view synthesis primarily focus on perspective images, recent research has been adapted for panoramic input. OmniNeRF [9] extends the pinhole camera model of NeRF to a fish-eye projection model and uses spherical sampling to enhance the quality of rendering. 360Roam [12] is the first to construct an omnidirectional neural radiance field from a sequence of

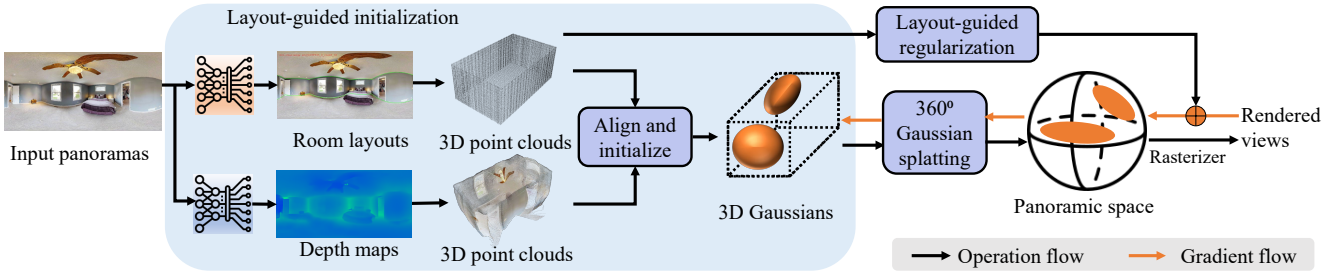


Figure 2. **Overview of 360-GS architecture**. Given a limited set of panoramas, we estimate the room layout and depth to guide the optimization of 3D Gaussian. These priors are transformed into 3D representations and jointly merged to a point cloud, which is used to initialize 3D Gaussians. We propose a 3D Gaussian splatting algorithm to project 3D Gaussians to panoramic space. Based on the projected Gaussians, we can render panoramas through a differentiable tile rasterizer. To reduce floaters in novel views, we regularize the optimization of 3D Gaussians by minimizing the cosine distance between the movement of position vectors and normals of layout point clouds.

panoramic images. 360FusionNeRF [19] introduces a semantic consistency loss with CLIP-ViT [26] to enforce 3D space consistency in panoramas. PanoGRF [7] proposes generalizable spherical radiance fields which incorporates scene priors from 360° dataset into Spherical NeRF. It further leverages a 360° monocular depth network to enhance the quality of geometry features and improve the rendering performance. In contrast, recent works built on 3D Gaussian splatting barely consider panoramas as input.

2.2. Layout priors in panoramas

Panoramic room layout estimation plays a crucial role in indoor scene comprehension and has been extensively studied [17, 23, 31, 32, 36, 37]. Among them, HorizonNet [37] introduces a deep learning network and a post-processing technique that can recover complex room layouts, even with obscured corners from the model output. This estimated panoramic room layout has been widely explored in computer vision problems including indoor navigation [21] and scene reconstruction [11, 14]. In the task of novel view synthesis, Xu et al. [42] utilize the estimated room layout from the reference panorama and extract high-level features as guidance for target views, proving the efficacy of layout priors. However, the neural information of the room layout is underutilized in 3D Gaussians, as 3D Gaussians lack neural components. Unlike this method, we exploit room layout priors through explicit initialization and geometric constraints for 3D Gaussians.

3. Method

We identify challenges in adapting panoramas to 3D-GS (Sec. 3.1) and present 360-GS as a solution. 360-GS proposes a 360° Gaussian splatting algorithm (Sec. 3.2) and exploits room layout priors (Sec. 3.3) to optimize 3D Gaussians and facilitate panoramic rendering. It also designs

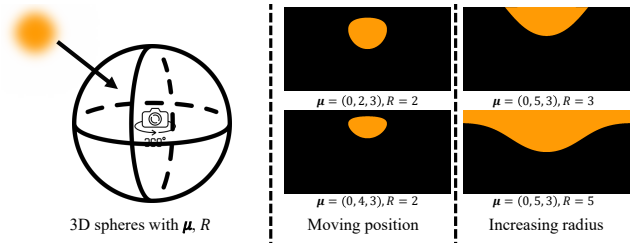


Figure 3. **Illustration of panoramic Gaussian splatting**. We show a toy case for splatting a 3D sphere (a special case of 3D Gaussian) onto panoramic images. The 3D sphere is defined by its position $\boldsymbol{\mu} \in \mathbb{R}^3$ and its radius $R \in \mathbb{R}$. In the middle column, we visualize the projection of 3D spheres under different positions. In the right column, 3D spheres with varying radii are splatted onto the panoramas. These projections are not elliptical and cannot be accurately modeled with 2D Gaussians, as attempted by 3D-GS.

a layout-guided initialization (Sec. 3.4) and regularization (Sec. 3.5) using room layout priors. Fig. 2 shows an overview of 360-GS.

3.1. Preliminary and challenge

3D-GS [18] explicitly represents a 3D scene with a collection of 3D Gaussians in world space. Each Gaussian is defined by a position vector $\boldsymbol{\mu} \in \mathbb{R}^3$ and a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$. The 3D Gaussian distribution can be represented as follows:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \quad (1)$$

For differentiable optimization, 3D-GS renders 2D images by projecting 3D Gaussians to 2D image planes. Given points $\boldsymbol{x} = (x_0, x_1, x_2)$ in world coordinates, we first transform them to camera coordinates $\boldsymbol{t} = (t_0, t_1, t_2)$ using an affine mapping $V(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{d}$, known as the viewing transformation. Subsequently, the camera coordinates are converted to ray coordinates through the mapping $p(\boldsymbol{t})$.

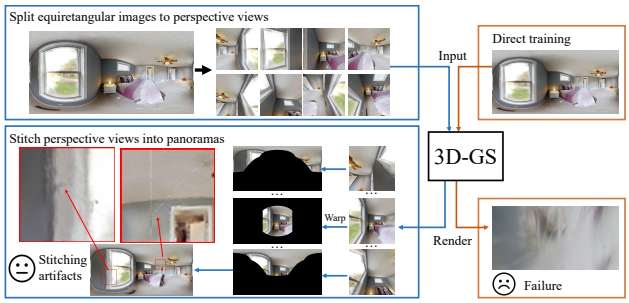


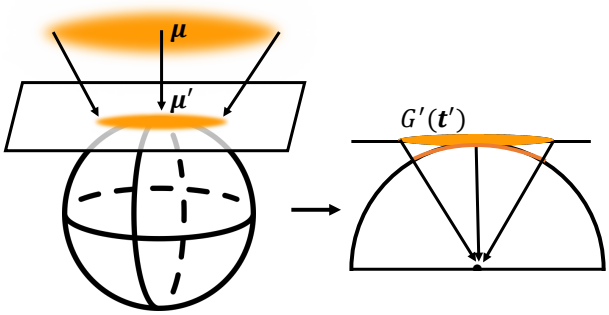
Figure 4. **Two naive pipelines for applying 3D-GS to panoramic inputs**. Left: we split panoramas into perspective images with poses and feed them to 3D-GS. To render panoramas, we render perspective images centered at the camera of panoramas. Subsequently, these images are transformed and stitched into equirectangular projection. These operations introduce stitching artifacts. Right: we fail to directly train 3D-GS with panoramic inputs.

Taking perspective images as inputs, these mappings are in fact not affine. To solve this problem, Zwicker et al. [45] introduce the local affine approximation of the projective transformation with the Jacobian matrix $\boldsymbol{J}$. As a result, the new covariance matrix $\Sigma^{'}$ of projected 2D Gaussians in camera coordinates are formulated as:

$$\Sigma^{'} = \boldsymbol{JW}\Sigma\boldsymbol{W}^{T}\boldsymbol{J}^{T}. \tag{2}$$

Since the local affine approximation relies on projective transformation, it is not appropriate for mapping 3D Gaussians to 2D Gaussians on panoramic images. A panoramic image covers the whole 360° horizontally and the whole 180° vertically, so the top and bottom of the image appear severely distorted. As illustrated in Fig. 3, the panoramic projection assumes distinct shapes that can not be modeled with Gaussians under varying configurations. Employing a 2D Gaussian for fitting such a projection would cause significant errors.

An alternative approach for the application of 3D-GS to panoramic inputs involves transforming the panoramas into perspective images before optimizing 3D Gaussians. An overview of this method is illustrated in Fig. 4. Concretely, we split equirectangular images into N perspective views, each associated with a distinct pose. Then these perspective images can be utilized to optimize 3D Gaussians following 3D-GS. However, this straightforward solution presents two main drawbacks: (1) The complete pipeline is intricate, and the direct acquisition of panoramas is unfeasible. (2) To get a complete panorama, more than six perspective images are supposed to be rendered and concatenated jointly. Unfortunately, this concatenation introduces inevitable stitching artifacts in overlapping regions of the reconstructed panoramas shown in Fig. 4.



Splatting on the tangent plane Mapping to spherical surface

Figure 5. 360° **Gaussian splatting algorithm.** 360° Gaussian splatting splats 3D Gaussians on the tangent plane that passes through the projection point $\boldsymbol{\mu}'$, yielding the distribution $G'(\boldsymbol{t}')$. Then we map the projection to the spherical surface of the unit sphere.

3.2. 360° Gaussian splatting

Our goal is to optimize 3D Gaussian representations from a set of panoramas and enable direct panorama rendering. Considering the challenges of directly representing spherical projection, we leverage the splatting technique [13] that decomposes the splatting on the spherical surface into two sequential steps: splatting on the tangent plane of the unit sphere and mapping to the spherical surface. This allows us to project 3D Gaussians to 2D Gaussians for rendering. An overview of 360° Gaussian splatting is illustrated in Fig. 5.

Given a 3D elliptical Gaussian centered at $\boldsymbol{\mu}$ with a covariance matrix Σ, we first convert it to the camera coordinates with the affine viewing transformation $V(\boldsymbol{x})$. The viewing transformation is then followed by a projective transformation $\boldsymbol{t'} = \varphi(\boldsymbol{t}, \boldsymbol{\mu'})$ that projects camera coordinates to the tangent plane of the unit sphere. This tangent plane passes through the projection point $\boldsymbol{\mu'}$ and is tangential to the unit sphere centered at the origin of the camera coordinates. The transformation is formulated as:

$$\boldsymbol{\mu'}(\boldsymbol{x} - \boldsymbol{\mu'}) = 0 \tag{3}$$

where $\boldsymbol{\mu'}$ is the projection of $V(\boldsymbol{\mu})$ onto the unit sphere. Thus the projection is given by:

$$(t_0^{'}, t_1^{'}, t_2^{'})^T = \varphi(\boldsymbol{t}, \boldsymbol{\mu'}) = \boldsymbol{t}\frac{(\boldsymbol{\mu'})^T\boldsymbol{\mu'}}{(\boldsymbol{\mu'})^T\boldsymbol{t}}. \tag{4}$$

Following Zwicker et al. [45], we define the local affine approximation $\varphi_k(\boldsymbol{t}, \boldsymbol{\mu'})$ by the first two terms of the Taylor expansion of φ at the point $\boldsymbol{t}_k$:

$$\varphi_k(\boldsymbol{t}, \boldsymbol{\mu'}) = \varphi_k(\boldsymbol{t_k}, \boldsymbol{\mu'}) + \boldsymbol{J}_k \cdot (\boldsymbol{t} - \boldsymbol{t}_k) \tag{5}$$

where $\boldsymbol{t}_k = (t_0, t_1, t_2)^T = V(\boldsymbol{\mu})$ is the center of the 3D Gaussian in camera coordinates. The Jacobian $\boldsymbol{J}_k$ is given by the partial derivatives of φ at the point $\boldsymbol{t}_k$:

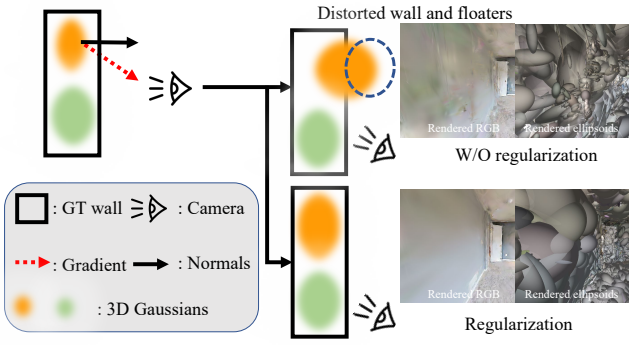


Figure 6. **Impact of layout-guided regularization.** We present a 2D toy case for optimizing 3D Gaussians, marked with an orange color. Without our regularization, 3D Gaussians gravitate towards the gradient direction, disrupting the inherent layout structure. This results in some Gaussians appearing outside the walls, causing distorted planes and "floaters" in novel views. Our layout-guided regularization effectively preserves the overall structure during optimization.

$$\boldsymbol{J}_k = \begin{bmatrix} \frac{\mu_1' t_1 + \mu_2' t_2}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{-\mu_1' t_0}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{-\mu_2' t_0}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} \\ \frac{-\mu_0' t_1}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{\mu_0' t_0 + \mu_2' t_2}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{-\mu_2' t_1}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} \\ \frac{-\mu_0' t_2}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{-\mu_1' t_2}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} & \frac{\mu_0' t_0 + \mu_1' t_1}{(\mu_0' t_0 + \mu_1' t_1 + \mu_2' t_2)^2} \end{bmatrix}. \tag{6}$$

We splat 3D Gaussians onto the tangent plane by concatenating $\boldsymbol{t} = V(\boldsymbol{x})$ and $\boldsymbol{t'} = \varphi(\boldsymbol{t}, \boldsymbol{\mu'})$, yielding the function as follows:

$$G'(\boldsymbol{t'}) = e^{-\frac{1}{2}(\boldsymbol{t'}-\boldsymbol{\mu'})^T (\boldsymbol{J}_k \boldsymbol{W} \Sigma \boldsymbol{W}^T \boldsymbol{J}_k{}^T)^{-1} (\boldsymbol{t'}-\boldsymbol{\mu'})}. \tag{7}$$

For panorama rendering, we map the tangent plane in camera coordinates to the spherical surface in spherical polar coordinates. Since the mapping process is efficient, our approach maintains real-time performance. Consequently, we are enabled to render panoramas directly.

3.3. Layout prior for panoramas

In the context of sparse panoramas lacking 3D information, 3D-GS struggles to identify cross-view 3D correspondences and construct the geometry of scenes, leading to a significant degradation in the quality of novel view synthesis. In this paper, we exploit the room layout, a form of 3D structural information within panoramas, to alleviate these issues. Incorporating the room layout with 3D Gaussians has three advantages. Firstly, it contains whole-room contextual information and 3D priors that are consistent across diverse views. Secondly, unlike depth maps and point-cloud representations, the room layout describes the scene with a smooth surface structure, yielding seamless planes including walls and floors [17]. Third, room layouts are easily accessible and robust to the scale of scenes. Recent advancements [37] have significantly propelled the field of layout estimation, attaining frame rates exceeding 20 FPS.

Under the assumption that room layouts conform to the Atlanta World assumption [23], room layouts are composed of vertical walls, horizontal floor and ceiling. As illustrated in Fig. 2, we depict room layouts using floor-wall boundaries $\boldsymbol{B}_f$ and ceiling-wall boundaries $\boldsymbol{B}_c$. Specifically, we sample N points with equal longitude intervals on these boundaries. These points are denoted as : $\{p^i\}_{i=1}^N = \{(\theta^i, \phi^i)\}_{i=1}^N$ where $\theta^i = 2\pi(\frac{i}{N} - 0.5)$ is the longitude and $\phi^i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the latitude. Consequently, floor-wall boundaries $\boldsymbol{B}_f = \{(\theta_c^i, \phi_c^i)\}_{i=1}^N$ and ceiling-wall boundaries $\boldsymbol{B}_c = \{(\theta_c^i, \phi_c^i)\}_{i=1}^N$ are represented with collections of points.

Given the known camera height h_c, we derive the horizon-depth values from points in the ceiling-wall boundary:

$$F_d(h_c, \phi_c^i) = \frac{h_c}{||\tan \phi_c^i||}. \tag{8}$$

Consequently, 2D points $\{(\theta^i, \phi^i)\}_{i=1}^N$ on the boundaries are transformed to a sequence of 3D points:

$$\begin{cases} p_x^i = F_d(h_c, \phi_c^i) \sin \theta^i \\ p_y^i = F_d(h_c, \phi_c^i) \tan \phi_c^i \\ p_z^i = F_d(h_c, \phi_c^i) \cos \theta^i \end{cases} \tag{9}$$

These 3D points $\{(p_x^i, p_y^i, p_z^i)\}_{i=1}^N$ are located on the floor and ceilings, thereby delineating the planes of the floor and ceilings. The space enclosed by the walls and ceiling is identified as the indoor area. Consequently, we construct a 3D bounding box that represents the scene's layout, which comprises the walls, ceiling, and floor.

3.4. Layout-guided initialization

Previous studies [6, 18] have demonstrated the importance of a reasonable geometric initialization in 3D Gaussians. 3D-GS advocates for starting with an initial set of sparse points derived from Structure-from-Motion (SfM) [28, 29]. However, SfM fails with sparse-view inputs, and so cannot reliably provide point cloud initializations [34].

Given that room layouts reveal the global geometric structure of the scene, we integrate the layout point cloud into the initialization. Specifically, we estimate a floor-wall boundary $\boldsymbol{B}_f$ and ceiling-wall boundary $\boldsymbol{B}_c$ as the layout for each panorama using off-the-shelf network [37]. To derive the global layout of the scene, we merge the 2D ceiling-wall boundaries of all panoramas through a 2D union operation. Similarly, $\boldsymbol{B}_c$ is unified into a global floor-wall boundary. Subsequently, we construct the corresponding 3D bounding box from the global layout following Eq. 9. The global 3D bounding box is converted into a point cloud

Figure 7. Qualitative comparison of our methods and some SOTA methods with 32-view inputs. 3D-GS needs to stitch perspective images into a panorama, resulting in stitching artifacts shown in the second row. Our method circumvents these artifacts and faithfully produces texture on the planes such as walls and floors. This is attributed to our exploitation of room layout priors and effective panoramic Gaussian splatting design.

through uniform sampling. To augment information for objects not included in the layout, we also estimate depth for panoramas and convert depth maps into point clouds [24]. We merge these depth point clouds to a global point cloud and then downsample it to reduce the number of points while maintaining the structure of objects. The merged depth point cloud is aligned to the layout point cloud with a global scale factor. Finally, we combine the layout and depth point clouds to initialize 3D Gaussians.

3.5. Layout-guided regularization

Although 3D Gaussians are initially set with a layout-guided point cloud, the room layout priors within panoramas suffer from catastrophic forgetting. As illustrated in Fig 6, the parameters of 3D Gaussians, such as position vectors $\boldsymbol{\mu}$, are optimized in the direction of the gradient. Consequently, 3D-GS struggles to preserve the geometric structure initialized with layout priors, leading to uneven surfaces and the emergence of "floaters" in novel views.

To address this issue, we introduce a layout-guided regularization to enforce 3D Gaussians to maintain the consistency of the room layout. Specifically, during the initialization of 3D Gaussians using the layout point cloud, we record the initial positions $\boldsymbol{u_0}$ and normals $\boldsymbol{n}$ of 3D layout points. We regularize the optimization of 3D Gaussians by minimizing the cosine distance between the displacement vector of each Gaussian's position and its associated normal vector. Finally, we aggregate these cosine distances across all 3D Gaussians to form the regularization term:

$$\mathcal{L}_{\text{layout}} = \sum \frac{\|\boldsymbol{n} \cdot (\boldsymbol{\mu} - \boldsymbol{u_0})\|}{\|\boldsymbol{n}\|\|\boldsymbol{\mu} - \boldsymbol{u_0}\|}. \tag{10}$$

4. Experiments

4.1. Implementation details

Our 360-GS is implemented based on the Pytorch framework in 3D-GS [18]. To obtain priors for our layout-guided initialization, we utilize the pretrained HorizonNet [37] for layout estimation and SliceNet [24] for monocular panoramic depth estimation. Our final loss function for optimization is defined as:

$$\mathcal{L} = \lambda_1 \|\boldsymbol{C} - \hat{\boldsymbol{C}}\|_1 + \lambda_2 \mathcal{L}_{\text{D-SSIM}} + \lambda_3 \mathcal{L}_{\text{layout}} \tag{11}$$

where $\mathcal{L}_{\text{D-SSIM}}$ is the D-SSIM term between rendered panoramas $\boldsymbol{C}$ and ground truth panoramas $\hat{\boldsymbol{C}}$. $\mathcal{L}_{\text{layout}}$ stands for the layout-guided regularization terms.

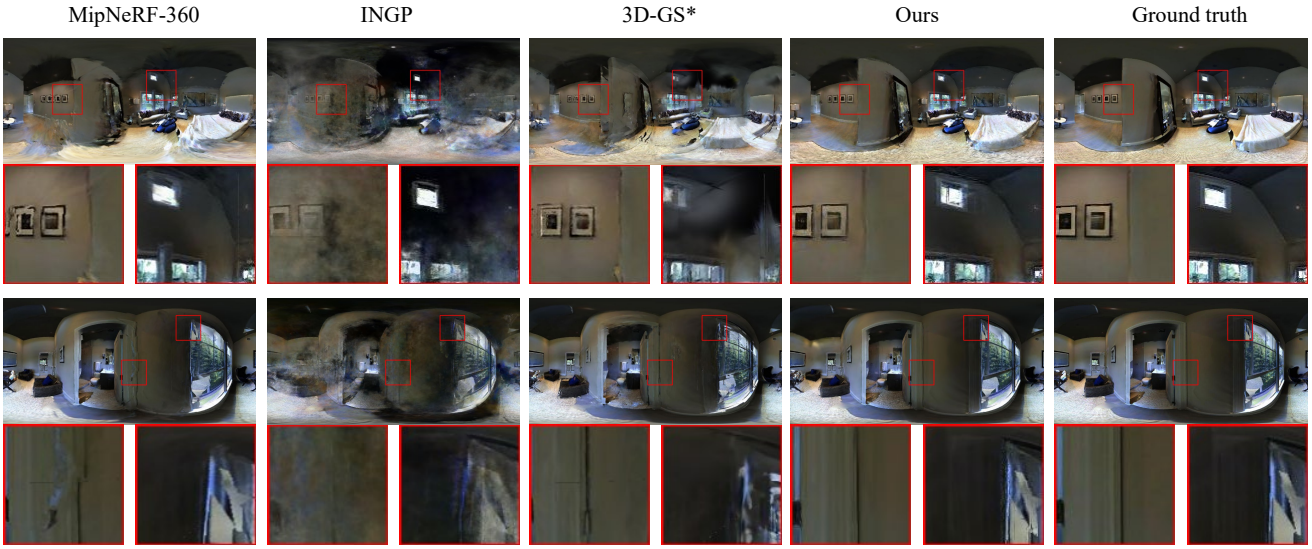


Figure 8. Qualitative comparison of our methods and some SOTA methods with 4-view inputs. Due to the unavailability of the SfM point cloud in sparse views, we only present results of 3D-GS* initialized with a random point cloud. The random initialization introduces noises and blurry artifacts in such under-constrained cases. INGP suffers from inadequate views, leading to over-smoothed outcomes and a tendency to overfit on training views. In contrast, our method produces visually appealing renderings comparable to those of MipNeRF-360. Moreover, our method holds the advantage of faster rendering while preserving better details.

Table 1. Quantitative evaluation of our method against NeRF-based methods and 3D-GS with different initialization strategies. The **best** and second-best scores are highlighted, respectively. 3D-GS* refers to the random initialization method and 3D-GS is initialized with SfM point clouds. Note that SfM point clouds are not available in the sparse 4-view inputs, resulting in some blank sections.

Metrics		M-360	INGP	3D-GS*	3D-GS	Ours
FPS		0.07	3.08	60		60
4-view	PSNR↑	**19.15**	15.49	13.92	-	18.96
	SSIM↑	**0.633**	0.432	0.438	-	0.600
	LPIPS↓	0.374	0.586	0.547	-	**0.344**
32-view	PSNR↑	26.72	**28.23**	21.65	26.74	28.22
	SSIM↑	0.835	0.860	0.704	0.837	**0.871**
	LPIPS↓	0.186	**0.099**	0.334	0.168	0.107

4.2. Experimental setting

Dataset. For both quantitative and qualitative evaluations, we gathered a total of 10 real-world scenes from the publicly available Matterport3D dataset [4]. Each scene, characterized by varied styles and furniture configurations, contains over 40 panoramas, each with a resolution of 512×1024 pixels. From these panoramas, we uniformly selected 4 and 32 panoramas as the training views for each scene. The remaining panoramas constitute the test set.

Baseline and metrics. We compare 360-GS with 3D-GS [18] and two state-of-the-art NeRF-based methods: MipNeRF-360 (M-360) [2] and INGP [22]. Given that 3D-GS only processes perspective images, we split each training panorama into eight perspective images, each with a resolution of 512×512 pixels. For evaluation, we report the average PSNR, SSIM, and LPIPS scores for all the methods under different numbers of training views. In addition, we report the FPS for rendering a 512×1024 image.

4.3. Results

Quantitative comparisons. Tab. 1 reports the quantitative results of SOTA methods and our 360-GS. Our method outperforms 3D-GS in terms of all metrics and input settings. In a 4-view setting, our method surpasses 3D-GS* with a remarkable 5.04 PSNR improvement. Despite the substantial performance improvement of 3D-GS with SfM point cloud initialization, it still falls short when compared to our method due to stitching artifacts. In the 32-view evaluation, INGP excels with the highest PSNR and LPIPS, while our proposed method leads in SSIM. This showcases the competitiveness of our method with INGP, especially considering that INGP's performance dramatically degrades in the 4-view setting. In the 4-view evaluation, our method is comparable with MipNeRF-360 and outperforms other methods in terms of LPIPS. However, the training and rendering time for MipNeRF-360 is considerably longer than ours. The quantitative comparison demonstrates that our

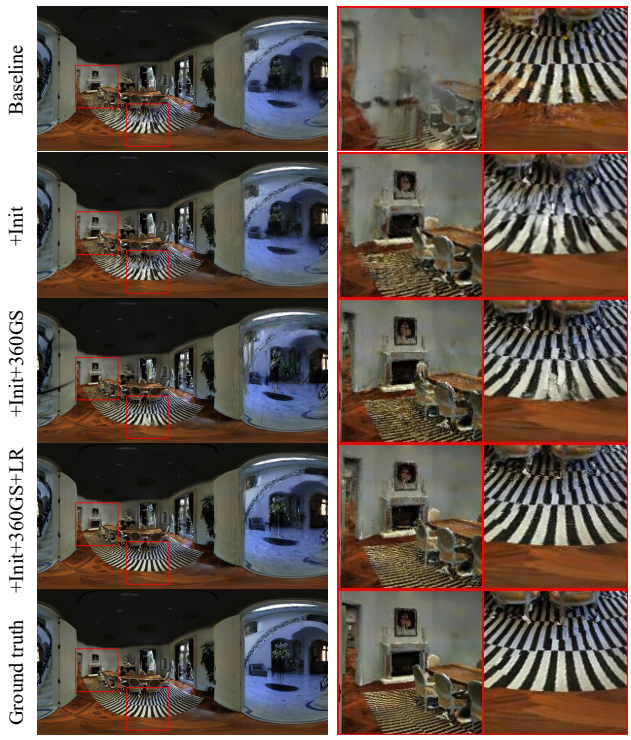


Figure 9. **Visualization of ablation study.**

method achieves state-of-the-art performance while ensuring fast rendering and robustness to the limited views.

Qualitative comparisons. We present a qualitative comparison of the rendering results across all methods in Fig. 7 and Fig. 8. Our method exhibits superior visual quality with 4-view inputs. Our results are comparable to those from MipNeRF-360 but at a lower computational cost. 360-GS effectively reconstructs the overall scene structure under the guidance of room layout priors, delivering visually pleasing results at first glance. Given a sufficient input of 32 panoramas, all methods yield satisfactory results. However, our method excels in recovering intricate patterns on planes.

4.4. Ablation study

In Tab. 2, we validate the effectiveness of our design choice on a scene from the Matterport3D dataset under the 4-view inputs. We visualize the ablation results in Fig. 9.

Layout-guided initialization. The baseline, 3D-GS, initialized with random point clouds, presents a substantial challenge for 3D Gaussians in learning the scene's geometry. This results in novel views exhibiting noise and blurry artifacts due to the under-constrained random 3D Gaussians shown in the first row of Fig. 9. In contrast, the integration of layout-guided initialization into the baseline offers a plausible geometry for 3D Gaussians, leading to a significant PSNR enhancement of 2.34. This also aids in visually reconstructing the scene's overall structure and wall-adjacent details like the fireplace in Fig. 9.

Table 2. **Ablation studies.** We start with the baseline 3D-GS (first row), then incorporate our layout-guided initialization denoted as "Init", resulting in a significant enhancement (second row). The application of 360° Gaussian splatting (360GS) further enhances the quantitative results in the third row. 360-GS is refined towards an improved solution with layout-guided regularization (LR) in the fourth row.

Init	360GS	LR	PSNR↑	SSIM↑	LPIPS↓
✗	✗	✗	13.64	0.458	0.459
✓	✗	✗	15.98	0.571	0.385
✓	✓	✗	16.66	0.588	0.334
✓	✓	✓	17.72	0.622	0.318

360° *Gaussian splatting.* The performance of 3D-GS on panoramas is limited by stitching artifacts that arise during the concatenation process. Our 360° Gaussian splatting fundamentally addresses this issue, leading to enhancements across all quantitative metrics as demonstrated in Tab. 2. As shown in the third row of Fig. 9, 360° Gaussian splatting not only eliminates the stitching artifacts but also enriches the structural and visual details on the checkerboard-patterned carpet.

Layout-guided regularization. While room layout priors offer a plausible initial state for 3D Gaussians, there might be inconsistencies between optimized 3D Gaussians and the layout, leading to some artifacts on the carpet. In the fourth row of Fig. 9, we observe that our layout-guided regularization efficiently eliminates these artifacts, ensuring planes that are more consistent with geometric coherence. The effectiveness of the regularization is further demonstrated by a notable PSNR improvement of 1.06.

5. Conclusion

We present a novel layout-guided panoramic Gaussian splatting pipeline named 360-GS, which enables direct panoramic rendering and is robust to sparse inputs. The cornerstone of 360-GS is our 360° Gaussian splatting algorithm and the incorporation of room layout priors. The 360° Gaussian splatting algorithm utilizes a perspective projection and mapping, thereby enabling the direct optimization of 3D Gaussians with equirectangular images. We leverage room layout priors within panoramas during the initialization, providing a more accessible and robust alternative to the SfM point cloud. We additionally introduce a layout-guided regularization to mitigate floater issues and preserve the geometric structure of the room layout. 360-GS supports real-time roaming and delivers state-of-the-art performance on real-world scenes for novel view synthesis.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021. 1

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2021. 2, 7

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 1

[4] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017. 7

[5] Zhengjie Chen, Chen Wang, Yuanchen Guo, and Song-Hai Zhang. Structnerf: Neural radiance fields for indoor scenes with structural hints. *ArXiv*, abs/2209.05277, 2022. 2

[6] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *ArXiv*, abs/2309.16585, 2023. 5

[7] 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, 2024. 3

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12872–12881, 2021. 2

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

[10] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[11] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *CVPR*, 2022. 3

[12] Huajian Huang, Ying-Rui Chen, Tianjian Zhang, and Sai-Kit Yeung. 360roam: Real-time indoor roaming using geometry-aware 360° radiance fields. *ArXiv*, abs/2208.02705, 2022. 2

[13] Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. On the error analysis of 3d gaussian splatting and an optimal projection strategy. *arXiv preprint arXiv:2402.00752*, 2024. 4

[14] Hamid Izadinia, Qi Shan, and Steven M. Seitz. Im2cad. 2016. 3

[15] Dominic Jack, Jhony Kaesemodel Pontes, Sridha Sridharan, Clinton Fookes, Sareh Abolahrari Shirazi, Frédéric Maire, and Anders P. Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, 2018. 2

[16] Ajay Jain, Matthew Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021. 2

[17] Zhigang Jiang, Zhongzheng Xiang, Jinhua Xu, and Mingbi Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1644–1653, 2022. 3, 5

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 1, 2, 3, 5, 6, 7

[19] 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. 3

[20] 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. In *European Conference on Computer Vision*, 2020. 1, 2

[21] Piotr Wojciech Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, L. Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. *ArXiv*, abs/1611.03673, 2016. 3

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41:1 – 15, 2022. 2, 7

[23] Giovanni Pintore, Marco Agus, and E. Gobbetti. Atlantanet: Inferring the 3d indoor layout from a single 360°image beyond the manhattan world assumption. In *European Conference on Computer Vision*, 2020. 3, 5

[24] Giovanni Pintore, Eva Almansa, and Jens Schneider. Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11531–11540, 2021. 6

[25] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2016. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 3

[27] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with

thousands of tiny mlps. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14315–14325, 2021. 2

[28] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[29] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[30] Jonathan Shade, Steven J. Gortler, Li wei He, and Richard Szeliski. Layered depth images. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998. 2

[31] Zhijie Shen, Zishuo Zheng, Chunyu Lin, Lang Nie, Kang Liao, and Yao Zhao. Disentangling orthogonal planes for indoor panoramic room layout estimation with cross-scale distortion awareness. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17345, 2023. 3

[32] Zhijie Shen, Zishuo Zheng, Chunyu Lin, Lang Nie, Kang Liao, Shuai Zheng, and Yao Zhao. Disentangling orthogonal planes for indoor panoramic room layout estimation with cross-scale distortion awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17345, 2023. 3

[33] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8025–8035, 2020. 2

[34] Samarth Sinha, Jason Y. Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B. Lindell. Sparsepose: Sparse-view camera pose regression and refinement. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21349–21359, 2022. 5

[35] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2441, 2018. 2

[36] Jheng-Wei Su, Chi-Han Peng, Peter Wonka, and Hung-Kuo Chu. Gpr-net: Multi-view layout estimation via a geometry-aware panorama registration network. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6469–6478, 2022. 3

[37] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 5, 6

[38] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5449–5459, 2021. 2

[39] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, New York, NY, USA, 2023. Association for Computing Machinery. 1

[40] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, 2019. 2

[41] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting. *ArXiv*, abs/2312.00206, 2023. 2

[42] Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. Layout-guided novel view synthesis from a single indoor panorama. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16433–16442, 2021. 3

[43] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2021. 2

[44] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *ArXiv*, abs/2312.00451, 2023. 2

[45] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 4