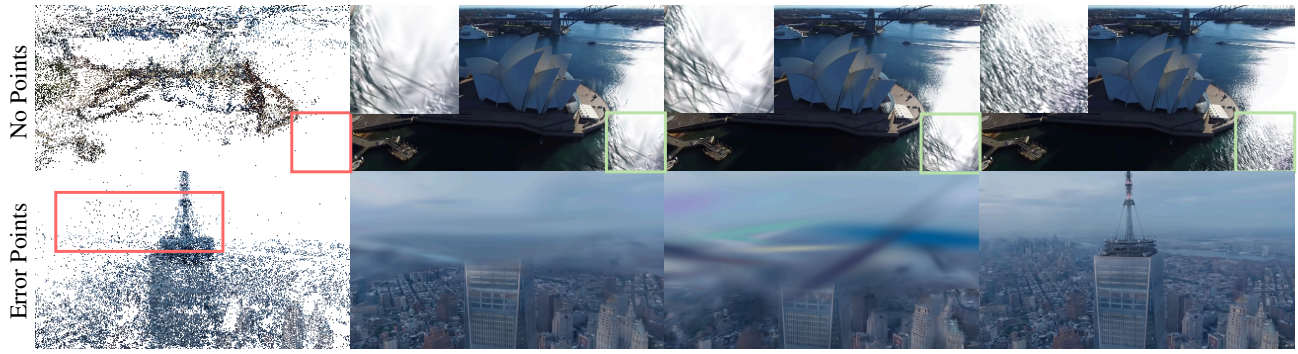
Liberated-GS: 3D Gaussian Splatting Independent from SfM Point Clouds

Weihong Pan[1*] Xiaoyu Zhang[2*] Hongjia Zhai[1] Xiaojun Xiang[2] Hanqing Jiang[2]
Guofeng Zhang[1†]
[1]State Key Lab of CAD&CG, Zhejiang University [2]SenseTime Research



(a) Erroneous SfM Points (b) 3D-GS (w/ SfM) (c) RAIN-GS (w/o SfM) (d) Ours (w/o SfM)

Figure 1. **Novel View Synthesis Comparison.** We propose a novel Gaussian Splatting initialization pipeline to address the degradation in novel view rendering quality caused by SfM point clouds. Specifically, the missing initial points from SfM lead to an inability to recover fine texture details, while numerous erroneous initial points result in excessive floaters, as shown in Fig.(a) and Fig.(b). Although RAIN-GS optimizes the initialization process, it still remains ineffective in resolving these issues. By incorporating depth priors from off-the-shelf models, our method recovers finer details and reduces floaters, ultimately improving rendering quality.

Abstract

3D Gaussian Splatting (3DGS) has demonstrated impressive performance in novel view synthesis and real-time rendering. However, it heavily relies on high-quality initial sparse points from Structure-from-Motion (SfM), which often struggles in textureless regions, degrading the geometry and visual quality of 3DGS. To address this limitation, we propose a novel initialization pipeline, achieving high-fidelity reconstruction from dense image sequences without relying on SfM-derived point clouds. Specifically, we first propose an effective depth alignment method to align the estimated monocular depth with depth rendered from an under-optimized coarse Gaussian model using an unbiased depth rasterization approach and ensemble them afterward. After that, to efficiently process dense image sequences, we incorporate a progressive segmented initialization process to generate the initial points. Extensive experiments demonstrate the superiority of our method over previous approaches and its compatibility with other advanced 3D Gaussian models. Notably, our method outperforms the SfM-based method by a 14.4% improvement in LPIPS on the Mip-NeRF360 datasets and a 30.7% improvement on the Tanks and Temples datasets.

1. Introduction

Novel view synthesis (NVS) [3, 27] represents a long-standing goal in computer vision and graphics, aimed at rendering images from arbitrary novel viewpoints of a 3D scene with a collection of images. It has demonstrated substantial potential across diverse applications, including VR/AR [42, 52], autonomous driving [58], and robotics [1].

Recently, Neural Radiance Fields (NeRF) [28] have shown their impressive novel view rendering performance. Despite the considerable success of NeRF, its practical applicability remains limited by high computational demands. Different from NeRF-based approaches [4–6, 29], 3D Gaussian Splatting (3DGS) [20] represents the scene with a set of learnable 3D Gaussian primitives explicitly, achieving faster rendering speed and better novel view synthesis performance. While 3DGS effectively addresses the slow rendering problem caused by radiance fields, it introduces additional input requirements. In addition to accurate camera calibration, 3DGS also requires additional point clouds to initialize these 3D primitives. Although sensors like LiDAR and ToF can provide point cloud data, they come with

*Authors contributed equally.
†Corresponding authors.

equipment constraints and limited adaptability in certain environments. Therefore, image-based Structure from Motion (SfM) techniques [2, 13, 33] are more commonly used. However, SfM can fail in certain real-world scenes, particularly in regions that are symmetrical or textureless [7, 54], resulting in erroneous or even unobtainable initial point clouds. This significantly degrades the rendering performance of 3DGS, as it cannot transport Gaussians far away from their initialized positions [18], leading to a lack of details and excessive floaters, as illustrated in Fig. 1.

Colmap-Free 3D Gaussian Splatting [14] proposes a progressive training strategy that leverages monocular depth to estimate relative camera poses and reconstruct from a sequence of unposed images. However, as [14] does not consider the scale of monocular depth, directly substituting estimated poses with fixed ground-truth poses results in scale inconsistencies. Moreover, [14] generates Gaussians for every pixel of each input image, which leads to substantial computational costs. Most recently, RAIN-GS [18] explored random initialization for 3DGS, demonstrating that sparse Gaussians with large variance can yield favorable results. However, the rendering performance of this method is also not competitive, suffering from poor detail recovery and floaters due to the lack of prior geometric constraints.

To address these problems, we propose an effective framework that does not rely on SfM point cloud initialization and achieves significantly better results than previous works. The contributions of our method are as follows.

- We propose Librated-GS, a novel initialization approach to eliminate the reliance on SfM points of 3D Gaussian Splatting.
- We align monocular depths using a coarse Gaussian model to resolve scale ambiguity and ensemble them to obtain high-quality geometric priors.
- We design a progressive segmented initialization process to generate the initial points, followed by an importance-aware resampling process to further reduce redundancy.
- We conduct extensive experiments on both standard and in-the-wild datasets to demonstrate the performance of our method and its compatibility with other advanced Gaussian models.

2. Related Work

Novel View Synthesis. Neural Radiance Field (NeRF) [28] has successfully achieved photo-realistic novel view synthesis by implicitly learning a multilayer perceptron (MLP) to represent scenes. It predicts the density and radiance at a given 3D point, using its position and direction as input, and employs a volumetric rendering technique to render images. Many subsequent works [4–6, 15, 16, 23, 24, 29, 30, 32, 35, 41, 44, 46, 50, 53, 56] have extended NeRF to unbounded/dynamic scenes [4–6, 23, 32], fewer training images [30, 34, 41, 44], and better efficiency [15, 24, 29, 35, 46]. Despite progress, NeRF remains computationally intensive due to its volumetric rendering with MLP. 3D Gaussian Splatting (3DGS) [20] and its follow-up studies [17, 26, 40, 51, 57, 59] offer a faster approach utilizing explicit 3D Gaussians and rendering through differentiable rasterization. While efficient, 3DGS typically depends on precise point clouds, commonly derived from Structure from Motion (SfM), to initialize the spatial locations and color attributes. Since 3DGS struggles to transport Gaussians initialized far from their optimal positions, such reliance limits the applicability of 3DGS when SfM fails. This motivates us to design an alternative that removes dependence on external point clouds for initialization.

3DGS without External Point Cloud. RAIN-GS [18] is proposed to use sparse 3D Gaussians with large variance for random initialization, removing the need for additional point clouds in 3DGS. While effective for scene reconstruction, this method struggles to recover fine details compared to point-based 3DGS due to the lack of prior information and is more prone to floaters. Colmap-Free 3D Gaussian Splatting [14] employs a progressive method with monocular depth as a prior for dense view reconstruction but are limited to scenes with a narrow range of viewpoint changes. Furthermore, since the poses are optimized based on the estimated depth, they are inherently aligned in scale, which does not necessarily hold for the ground truth poses. InstantSplat [10], designed for sparse views, relies on Dust3r [38] for globally aligned point clouds, limiting its ability to handle scenes with hundreds of images. Kheradmand *et al*. [21] proposes a stochastic optimization method by formulating the training process of 3D Gaussians as a sampling procedure for better densification, improving the rendering quality and the robustness to initialization. **However, there still remains a gap between initializing with SfM points and completely randomly, and certain problems caused by insufficient/erroneous SfM points are still unsolved.** Additionally, another line of work explores combining NeRF with Gaussian Splatting. [31] attempts to distill ZipNeRF [6] into 3DGS to improve the rendering speed without sacrificing the rendering quality. [12] proposes to draw initial points from a pre-trained NeRF model and optimize with the depths rendered from it.

3. Method

We propose a pipeline to reconstruct photo-realistic scenes from posed image sequences without requiring an input point cloud. In Sec. 3.1, we briefly review 3D Gaussian Splatting [20]. In Sec. 3.2, we introduce an effective depth alignment method to resolve scale ambiguity, using unbiased rendered depth as reference and combining them to build high-quality geometry priors. In Sec. 3.3, we present a progressive segmented initialization with importance resampling. The optimization stage remains unchanged.

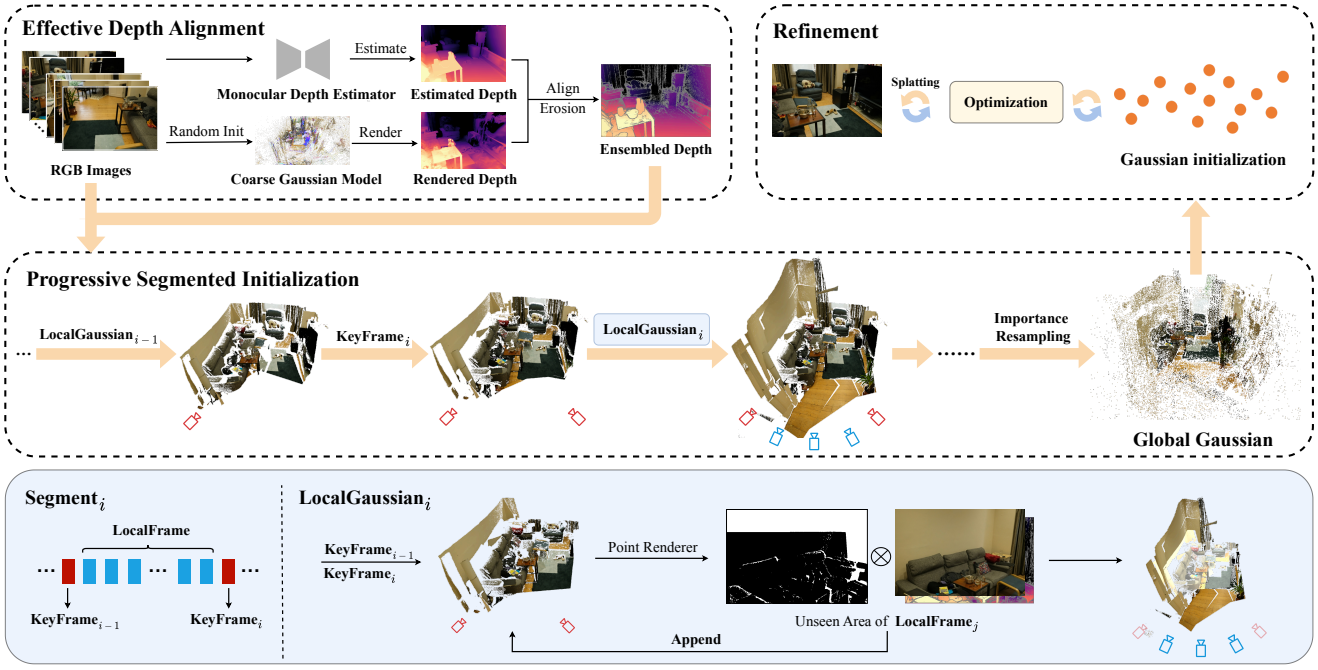


Figure 2. **Overview of our method**. First, we propose an effective depth alignment method to establish high-quality geometry priors, as described in Sec. 3.2. We then perform progressive segmented initialization to obtain an initial solution, followed by an importance resampling step to reduce redundancy further, as described in Sec. 3.3. Finally, we introduce the local Gaussian for each segment to fill unseen areas between adjacent keyframes, subsequently merging it into the global Gaussian. Standard refinement is then applied to optimize this solution.

3.1. Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents a 3D scene explicitly by employing a collection of 3D Gaussians. Each Gaussian $\mathcal{G}_i$ can be formulated in the world space with its mean $\mu \in \mathbb{R}^3$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ defined in the world space as follows:

$$\mathcal{G}_i(x) = e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i(x-\mu_i)}. \tag{1}$$

The mean μ of each 3D Gaussian is typically initialized from a sparse point cloud obtained through Structure from Motion (SfM). Each Gaussian is associated with an opacity $\sigma \in \mathbb{R}$ and several spherical harmonic (SH) coefficients.

To render an image from a given pose, the 3D Gaussian ellipsoids need to be splatted onto the image plane. For a given 2D pixel $\boldsymbol{u}$, The corresponding 2D Gaussian is represented as $\mathcal{G}_i^{2D}(\boldsymbol{u}) = e^{-\frac{1}{2}(\boldsymbol{u}-\mu_i^{2D})^T\Sigma_i'(\boldsymbol{u}-\mu_i^{2D})}$, where Σ' is the projected covariance matrix in the camera coordinates. Then, the differentiable rasterizer adopts alpha-blending to render the depth and color with N ordered Gaussians covering this pixel as follows:

$$C(\boldsymbol{u}) = \sum_{i=1}^{N} c_i\alpha_i\Pi_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

$$D(\boldsymbol{u}) = \sum_{i=1}^{N} d_i\alpha_i\Pi_{j=1}^{i-1}(1-\alpha_j), \tag{3}$$

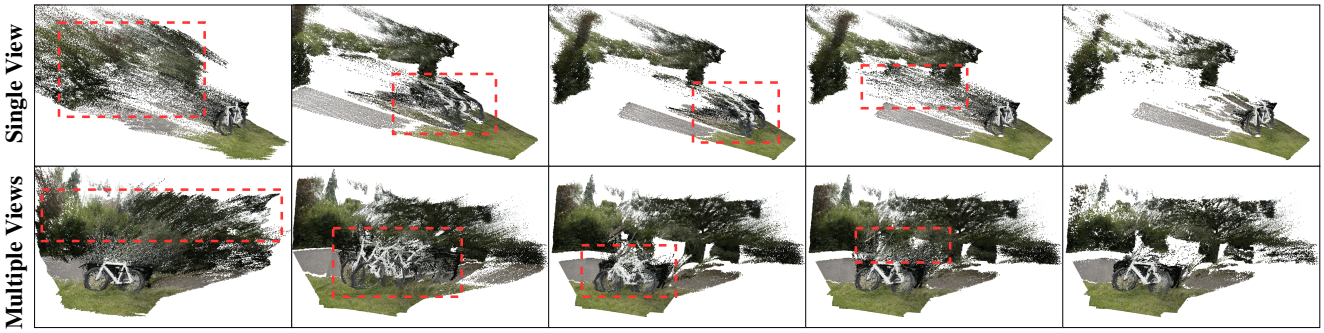
where c_i is the view-dependent color calculated from the SH coefficients, and $d_i = z \in \mathbb{R}$ denotes the z-coordinate for the Gaussian's center in the camera space (the red depth in Fig. 4). $\alpha_i = \sigma_i \cdot \mathcal{G}_i^{2D}(\boldsymbol{u})$ is the opacity contribution of each 2D Gaussian and $T_i = \Pi_{j=1}^{i-1}(1-\alpha_j)$ denotes the accumulated transmittance. Subsequently, 3DGS employs adaptive density control (ADC) to update Gaussians to achieve a denser and more accurate scene representation.

However, in an under-optimized Gaussian model, the Gaussian centers may deviate from the current ray due to incomplete densification. These geometric inaccuracies introduce noise during initialization. To address this, we propose an unbiased depth rendering method detailed in Sec. 3.2.

3.2. Effective Depth Alignment

To address scale ambiguity and obtain reliable geometry priors, we align the estimated depth with that rendered from a coarse model. As the coarse model is not fully optimized, direct alpha-blending introduces noise. We therefore render depth in an unbiased manner, align depths via region-specific affine transformations, and ensemble these two depths while filtering out floaters. This process is illustrated in the top left of Fig. 2.

Unbiased Depth Rendering. As highlighted in RAIN-GS [18], the sparse-large-variance (SLV) initialization enables effective signal prediction within few training steps.



(a) Rendered Depth **(b) Estimated Depth** **(c) Aligned Depth** **(d) Ensembled Depth** **(e) Ours**

Figure 3. **Point Cloud from different depths**. We compare the point cloud from different depths for single view and multiple views. (a) Rendered Depth from an under-optimized 3DGS model can achieve view-consistent recovery in well-observed areas but underperforms in low-texture regions. (b) Estimated Depth is rescaled under the scale-consistency assumption, which tends to generate incorrect 3D points and causes misalignment between different viewpoints. (c) Aligning monocular depth with rendered depth using Eq. (6) can partially alleviate this issue. (d) We ensemble the more reliable regions from (a) and (c) with Eq. (9). (e) We perform depth erosion on the ensembled depth to reduce floaters.

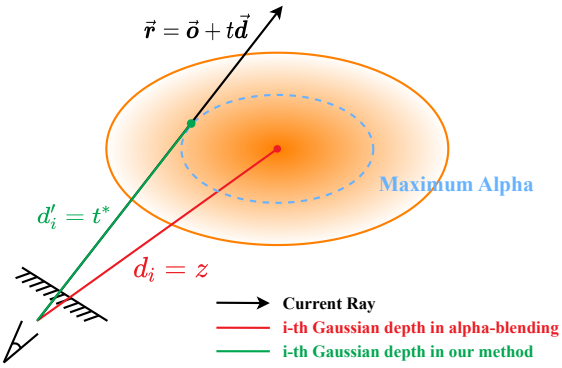


Figure 4. **Unbiased Depth Rendering.** Illustration of depth rendering with the alpha-blending method and our unbiased method.

This implies that the resulting coarse Gaussian model offers a rough estimate of the scene's depth. As illustrated in Fig. 3(a), our experiments demonstrate that the depth rendered by this pre-trained model is more accurate in high-frequency regions (foreground bicycle), but less reliable in low-texture areas (background sky), such as plain walls, floors, and skies.

Previous works [9, 14, 43] typically utilize alpha-blending to compute the depth at pixel $\boldsymbol{u}$ as Eq. (3). However, since we are employing an under-optimized model in which the Gaussians have not yet fully densified, their centers may deviate significantly from the current ray. Consequently, approximating the Gaussian depth using the center depth can result in a point cloud with significant floaters when reprojected back into the world space, as depicted in Fig. 5.

Our proposed solution is to replace d_i in Eq. (3) with a more accurate approximation $d_i' = t^*$: the point at which the opacity contribution α reaches its maximum as the ray passes through the current Gaussian ellipsoid (the green depth depicted in Fig. 4). Since α is proportional to function $\mathcal{G}_i$, this problem is equivalent to finding the parameter t for ray $\boldsymbol{r} = \boldsymbol{o} + t\boldsymbol{d}$ where $\mathcal{G}_i(\boldsymbol{r})$ reaches its maximum, *i.e.*

$$t^* = \arg\max_t \ \mathcal{G}_i(\boldsymbol{o} + t\boldsymbol{d}). \tag{4}$$

For the detailed derivation, please refer to the supplementary material Sec. 6.1.

We skip Gaussians that do not intersect with the current ray, which is indicated by the discriminant of Eq. (4) $\Delta < 0$. Above all, the rendered depth of pixel u is then given by:

$$D_{\text{render}}(u) = \sum_{i,\Delta_i>=0} d_i' \alpha_i(u) \prod_{j=1,\Delta_j>=0}^{i-1} (1 - \alpha_j(u)). \tag{5}$$

(a) Ours **(b) Alpha-Blending**

Figure 5. Visual comparison of depth maps and reprojected points with the standard alpha-blending method and our unbiased alpha-blending method.

Depth Alignment with Monocular Prior. Many previous works [9, 37, 49] have successfully utilized off-the-shelf monocular depth estimation models [19, 38, 45] to obtain prior geometric information. They address scale ambiguity by comparing estimated depths with sparse SfM points based on the scale-consistency assumption, which is not always as reliable as expected [8, 19]. This limitation is particularly evident in scenarios with significant scale variations or complex geometries, leading to the initialization of 3D points in incorrect positions, as illustrated in Fig. 3(b). To address this issue, we learn a separate affine transformation for each object in each image, which includes an independent scale s_i and shift t_i:

$$D_{\text{align}}(\boldsymbol{p}_i) = s_i \cdot \hat{D}_{\text{est}}(\boldsymbol{p}_i) + t_i, \tag{6}$$

where $\hat{D}_{\text{est}}$ represents the normalized estimated depth from [19], and $\boldsymbol{p}_i$ is the mask of the i-th object from [48]. Since we do not have SfM points as ground truth, the rendered depths from the coarse Gaussian Model serve as references.

Specifically, taking the current view I along with its rendered image I_{render} and depth map D_{render} from Eq. (5) into consideration, we leverage off-the-shelf detectors [36] to detect 2D correspondences between I and I_{render}. Let $\mathcal{K} = \{k_j\}$ and $\mathcal{K}' = \{k'_j\}$ denote the 2D points set for I and I_{render}, respectively. We solve for the scale s_i and shift t_i using the closed-form linear regression solution:

$$s_i, t_i = \arg\min \sum_{\boldsymbol{p}_i(k_j)=1} ||D_{\text{align}}(k_j) - D_{\text{render}}(k'_j)||_2^2, \quad (7)$$

where $D_{\text{align}}(k_j)$ and $D_{\text{render}}(k'_j)$ represent the depth value of matched points. An example from the *bicycle* dataset is depicted in Fig. 3(c).

Depth Ensemble and Edge-aware Erosion. We ensemble the aligned depth with rendered depth to obtain more reliable initial 3D points. Given a pixel, we denote the aligned depth as d_{align}, the depth obtained from our unbiased rendering (Eq. (5)) as the rendered depth d_{render}, and the depth of the Gaussian with the largest contribution weight $w_i = \alpha_i \prod_{j=1,\Delta_j>=0}^{i-1}(1-\alpha_j)$ as the maximum-weight depth d_{max}. Ideally, a ray passing through the surface of a solid object should satisfy $d_{\text{max}} = d_{\text{render}}$. Therefore, the selection of pixel depth is performed as follows, where τ_d is a pre-defined threshold.

$$d = \begin{cases} d_{\text{render}}, & (d_{\text{max}} - d_{\text{render}})/d_{\text{render}} < \tau_d \\ d_{\text{align}}, & \text{else} \end{cases}, \quad (8)$$

However, since the rendered depth is derived from an under-optimized model, this per-pixel depth ensemble tends to disrupt the depth continuity of each object. Therefore, we introduce a per-object depth ensemble method.

$$D(\boldsymbol{p}_i) = \begin{cases} D_{\text{render}}(\boldsymbol{p}_i), & \delta(\frac{(D_{\text{max}}(\boldsymbol{p}_i) - D_{\text{render}}(\boldsymbol{p}_i))}{D_{\text{render}}(\boldsymbol{p}_i)}) < \tau_d \\ D_{\text{align}}(\boldsymbol{p}_i), & \text{else} \end{cases}, \quad (9)$$

where $\delta(\cdot)$ represents the mean value of variable $(\cdot)$.

Due to significant errors existing at object edges in both rendered and aligned depth maps (Fig. 3(a) and (c)), similar problems arise in the ensembled depth, resulting in numerous floaters after back-projection (Fig. 3(d)). To mitigate this, we propose the following edge-aware depth erosion method. Consider a pixel u in the ensembled depth map D, $\mathcal{F}(u,r) = \{f_i = |D(u) - D(u_i)| \mid u_i \in P(u,r)\}$ denotes the depth differences between u and pixels within the patch $P(u,r)$ of radius r. If the number of $f_i < \tau_f$ is below n_f, $D(u)$ is considered to cause floater and excluded from initialization. τ_f is a pre-defined threshold proportional to the median of $\{\min(\mathcal{F}(u,r))\}$ and n_f is set to 3 in our experiments.

3.3. Progressive Segmented Initialization

For a sequence of n consecutively captured RGB images $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ with their corresponding ensembled depths $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$, a naive approach to forming initial points involves back-projecting all image pixels based on their ensembled depths, as proposed in [14]. However, this leads to redundant points in well-observed regions, creating an excessive number of initial Gaussians and severely affecting the subsequent optimization and densification. While point redundancy can be reduced through random selection [11] or voxel grid downsampling, the former requires manual tuning and risks missing details and the latter depends on prior knowledge of the scene scale to set the voxel size appropriately. To address this issue, we propose a progressive segmented initialization method to reduce the number of back-projected points, as illustrated in the middle of Fig. 2.

Segmented Initialization. To reduce the redundancy of pixels from the same region, we select m keyframes $\mathcal{I}_k = \{I_{k_1}, ..., I_{k_m}\}$ to separate the training images into several segments, as shown in the bottom left of Fig. 2. Instead of back-projecting every single image, we perform a segmented initialization by fully projecting the keyframes and partially the rest.

We select keyframes with the following procedure. Given that $i-1$ keyframes have been selected, we sequentially consider the rest images from $\{I_{k_{i-1}+1}, \ldots, I_n\}$. Using detectors mentioned in Sec. 3.2, we detect 2D correspondences with $I_{k_{i-1}}$ and select the first image satisfying $M/N < \tau_k$ as the next keyframe I_{k_i}, where M represents the number of matching points with high confidence score and N the total number of pixels. τ_k is a predefined threshold representing the size of the overlap area between two images relative to the entire image.

Although using keyframes to initialize points can mostly cover the scene, some areas are still out of observation, causing incomplete reconstruction for some views. To tackle this issue, we use images between adjacent keyframes, denoted as *local frames*, to fill in the missing regions by solely back-projecting the points from under-reconstruction areas. Besides, we downsample the training images to further reduce redundant 3D Gaussians.

Progressive Initialization. Although we have significantly improved the quality of the depth maps in Sec. 3.2, there still exists a considerable amount of noise. Inspired by but different from [14], we address this issue by learning a global 3DGS $\mathcal{G}_g$ incrementally to further refine the depth maps, during which only the centers of Gaussians are updated, while other attributes remain unchanged. This process starts from the first keyframe $I_{k_1} = I_1$, where all pixels are back-projected into world space to initialize $\mathcal{G}_g$. Assuming that previous $i-1$ keyframes have already been incorporated, the next keyframe I_{k_i} is then considered. First,

we optimize the photometric loss to refine the ensembled depths $\{D_1, D_2, \ldots, D_{k_i}\}$ for all previous views. Subsequently, all pixels in I_{k_i} are back-projected based on the optimized fused depth D_{k_i}.

After that, for current $\textit{segment}_i = \{I_{k_{i-1}+1}, \ldots, I_{k_i-1}\}$, local 3DGS $\mathcal{G}_{l_i}$ is introduced to enrich the global 3DGS $\mathcal{G}_g$, which is initialized with two adjacent keyframes $I_{k_{i-1}}$ and I_{k_i}. Local frames from $\textit{segment}_i$ are then incrementally incorporated. Different from global 3DGS, only points back-projected from the under-reconstruction areas of the rendered images from $\mathcal{G}_{l_i}$ will be added. Finally, the newly added points in $\mathcal{G}_{l_i}$ are then merged into the global 3DGS.

Importance Resampling. Since the scene is thoroughly observed, even with segmented initialization, the process aforementioned still generates a large number of 3D points, significantly impacting the subsequent optimization in 3DGS. Therefore, we apply importance resampling to streamline the initialized 3D points. We uniformly sample 10% of the total back-projected 3D points and train a coarse 3DGS model for 1000 steps without performing densification. The center of Gaussian with the highest contribution weight along each ray is retained as the initialization 3D point for refinement.

4. Experiments

4.1. Experimental Setup

Datasets. To validate the effectiveness of our method, extensive qualitative and quantitative comparison experiments are conducted on three real-world datasets, including two benchmark datasets (**Mip-NeRF360** [5] and **Tanks and Temples** [22]) and an in-the-wild dataset (**OMMO** [25]). Following the guidelines in Mip-NeRF360 [5], a train/test split is constructed by selecting every 8-th image for testing in each scene. We use the same image resolution as 3DGS [20] and report the standard evaluation metrics, including PSNR, SSIM [39], and LPIPS [55]. During the initialization process, it is ensured that there is an overlap between adjacent images, therefore the training set is reordered for initialization if this condition is not met. In the refinement process, the training set is randomly shuffled, as done in 3DGS.

Implementation Details. Our method is primarily implemented based on 3DGS [20]. During the pre-training process, sparse-large-variance (SLV) initialization is utilized, as suggested by RAIN-GS [18]. The number of pre-training steps is set to 5000 for indoor datasets and 10000 for outdoor datasets to achieve a roughly accurate scene structure. In the estimated depth alignment process, only the depths of matching points are optimized, leading to rapid convergence, therefore the number of fitting steps is set to 100 based on experimental results. The depth selection threshold τ_d is set to 0.1, while the proportionality coefficient for the depth ensemble threshold τ_f is set to 8. The interval for adding new keyframes during the initialization process matches the densification interval of 3DGS [20] to facilitate incremental scene growth. Our optimization parameters remain consistent with those listed in the 3DGS [20] configuration during the refinement process. All experiments are conducted on a single RTX4090 GPU.

4.2. Comparison

In this section, we compare our method with 3DGS [20] using SfM points for initialization alongside other methods that do not utilize SfM points: 3DGS with random points initialization and RAIN-GS [18]. The comparisons are conducted on two benchmark datasets (**Mip-NeRF360** dataset [5] and **Tanks and Temples** [22]) and an in-the-wild dataset **OMMO** [25].

Quantitative Evaluation. The quantitative results, including PSNR, SSIM, and LPIPS, are represented in Tab. 1. The *Colmap-Free 3DGS** is trained with fixed ground-truth poses. Additionally, since [14] does not account for scale when utilizing monocular depth to estimate camera poses and generate 3D points, directly using the ground-truth poses during training may lead to significant failures. To align the estimated monocular depth with the scene scale, we conduct the same rescaling process as outlined in [32] for [14]. Our method demonstrates substantial improvements across all three metrics compared to all other methods, even outperforming 3DGS initialized with SfM point clouds. This quantitatively validates that our approach achieves superior rendering and geometry results even without additional high-quality point clouds.

Qualitative Evaluation. We also present the qualitative results in Fig. 6. While 3DGS [20] initialized with SfM recovers fine details in regions with sufficient 3D points (*e.g.*, the nearby buildings in *Scene15*, the upper part of the bookshelf in *room*), areas lacking these points often miss critical details (*e.g.*, the distant city in *Scene10*, the underside of the chair in *bicycle*). 3DGS with random initialization suffers from more artifacts and geometric inaccuracies due to significant errors in the random initial points. Similarly, RAIN-GS [18], despite its new initialization strategy, struggles to address detail loss from insufficient initial points and produces lower geometric quality compared to SfM-initialized 3DGS (*e.g.*, the lower part of the bookshelf in *room*, the ground in *bicycle*), as reflected in the LPIPS score. The experimental results demonstrate that neither the random initialization approach (*e.g.*, 3DGS with random points and RAIN-GS) nor the solely depth-estimation-based approach (*e.g.*, [14] with ground-truth poses) achieves the rendering quality of SfM points-initialized 3DGS. In contrast, our method incorporates monocular depth estimation as an auxiliary prior, enhanced by effective depth alignment, to improve the initial 3D points quality. Furthermore, our

Table 1. **Quatitative Comparison on Mip-NeRF360 [5], Tanks and Temples [22] Datasets** and OMMO [25] Datasets. Colmap-Free 3DGS* indicates the model trained with ground-truth poses and rescaled estimated depths. The first, second, and third best performances are highlighted in red, orange, and yellow, respectively. Our method demonstrates superior performance compared to existing point-free methods and the original 3DGS across all metrics.

Methods	SfM Points	Mip-NeRF360 [5]			Tanks and Temples [22]			OMMO [25]		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS [20]	✓	27.21	0.815	0.214	23.14	0.841	0.183	27.79	0.885	0.185
3DGS [20] (Random)	×	22.19	0.704	0.313	20.48	0.765	0.252	25.86	0.841	0.229
Colmap-Free 3DGS [14]*	×	21.46	0.604	0.344	20.71	0.711	0.330	-	-	-
RAIN-GS [18]	×	27.23	0.807	0.229	23.13	0.826	0.207	26.93	0.864	0.218
Ours	×	27.59	0.822	0.187	23.59	0.848	0.140	28.01	0.891	0.157

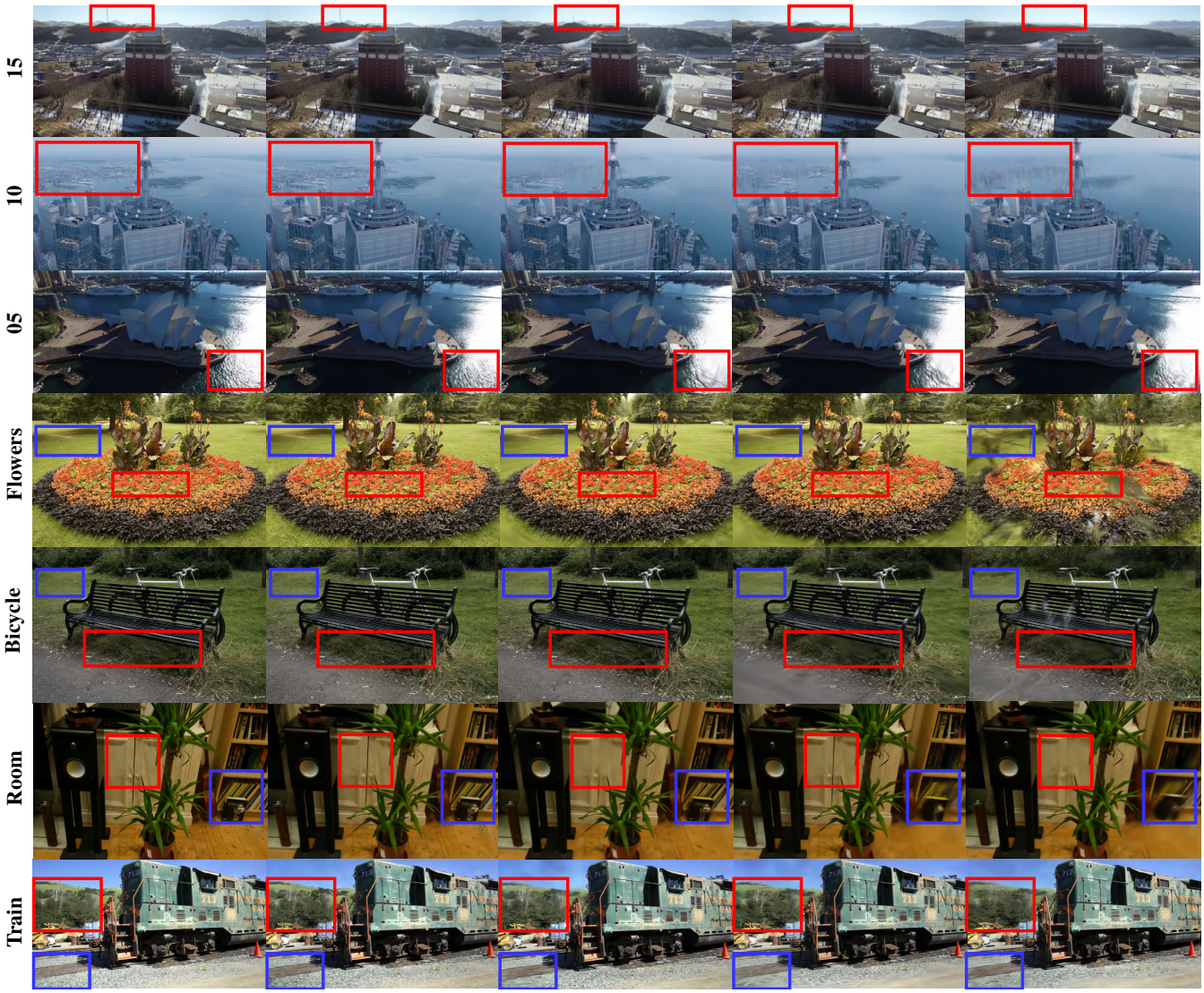


Figure 6. **Qualitative Comparisons of different methods on Mip-NeRF360 [5], Tanks and Temples [22] and OMMO [25] Datasets**. We conduct comparisons with original SfM-initialized 3DGS, random-initialized 3DGS (*3DGS(Random)*), and RAIN-GS. Our method significantly outperforms other methods, producing visually reliable results with sharper details. Note that our method surpasses the original 3DGS not only in areas with insufficient SfM points but also in regions with abundant points. Please zoom in for more details.

Table 2. **Applying our initialization method to different 3D Gaussian Splatting Models on OMMO [25] dataset.**

Init Methods	Mini-Splatting [11]			3DGS-MCMC [21]			3DGS [20]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Random	25.55	0.831	0.243	27.75	0.887	0.180	25.86	0.841	0.229
SfM Points	26.87	0.862	0.205	28.34	0.898	0.163	27.79	0.885	0.185
Ours	**27.29**	**0.877**	**0.174**	**28.50**	**0.904**	**0.153**	**28.01**	**0.891**	**0.157**

Table 3. Ablation for proposed components in our framework on Mip-NeRF360 [5] dataset.

Model	PSNR↑	SSIM↑	LPIPS↓
Base Model	27.079	0.808	0.194
+ Depth Alignment	27.285	0.812	0.190
+ Local Gaussian	**27.588**	**0.822**	**0.187**

Table 4. Ablation for different depths used for initialization on Mip-NeRF360 [5] dataset.

Depth	PSNR↑	SSIM↑	LPIPS↓
Ensembled Depth	**27.588**	**0.822**	**0.187**
Aligned Depth	27.524	0.818	0.189
Estimated Depth	27.390	0.816	0.191
Rendered Depth	26.596	0.708	0.201

segmented dense initialization effectively overcomes the limitations of SfM in providing initialization points under challenging conditions. As illustrated in Fig. 6, our method recovers more details in scenes where depth estimation confidence is high (*e.g.*, indoor scenes) and effectively mitigates artifacts and reconstruction errors caused by depth inaccuracies in scenes with lower estimation confidence (*e.g.*, outdoor scenes).

4.3. Ablation Study

We first evaluate our initialization method on the OMMO dataset [25] using two other advanced Gaussian frameworks with improved optimization processes: mini-splatting [11] and 3DGS-MCMC [21]. Our initialization does not interfere with subsequent optimization. Both frameworks enhance rendering quality by refining the Gaussian optimization process, with the latter further reducing sensitivity to initialization. Tab. 2 presents the results of both models under three different initialization conditions, demonstrating that the impact of initialization points on Gaussian rendering is universal and that the effectiveness of our method extends beyond the original Gaussian model.

Additionally, we conduct a series of experiments on the Mip-NeRF360 dataset [5] to validate the effectiveness of key components, namely depth alignment and local Gaussian, by incrementally introducing them into the base model. Results are presented in Tab. 3. As shown in the second row, depth alignment enhances scene geometry and visual details by leveraging a more accurate depth prior, resulting in a PSNR improvement of 0.206. The third row confirms that incorporating local frames during the initialization phase enriches depth priors for previously unseen areas, thus enhancing modeling quality.

Table 5. Comparison of runtime between our pipeline and COLMAP on *Scene03* (300 images, 1237×658 resolution) from OMMO [25] dataset.

Step	Ours		3DGS
Preprocess	Pose	2.27 min	-
	Depth	0.36 min	
	Segmentation	7.07 min	
Point Initialization	Depth Alignment	3.33 min	∼3.5 h (COLMAP)
	Initialization	43.03 min	
Refinement	42.62 min		33.74 min
Total Time	**98.68 min**		**∼4.1 h**

Besides, we perform ablation studies to assess the impact of different depth priors on model quality, as shown in Tab. 4. The edge-aware depth erosion strategy is applied across all experiments. Here, the estimated depth is rescaled based on the scale-consistency assumption, while aligned depth is with Eq. (6). The results demonstrate that separate affine transformation provides a depth prior that is more consistent with the scene scale, while the ensembled depth yields an additional improvement.

Finally, we conduct a runtime analysis on *Scene03*, which contains 300 images at a resolution of 1237×658, covering the entire pipeline. Benefiting from recent advances in foundation 2D/3D models, the preprocessing stages are highly efficient. As shown in Tab. 5, our initialization takes only 46 minutes, significantly faster than COLMAP's full SfM reconstruction, which requires several hours. As our method does not modify the refinement stage, its runtime remains comparable to the original pipeline.

5. Conclusion

In this work, we propose Librated-GS, a novel approach that removes the dependence on accurate initial point clouds in 3DGS for novel view synthesis from a sequence of images. We first fuse monocular depth estimates with coarse rendering depths to resolve scale ambiguity and establish geometric priors. Then, we introduce a progressive segmented initialization process that leverages both local and global Gaussians to construct a coarse solution. Extensive experimental results validate the effectiveness of our approach and its compatibility with other advanced 3D Gaussian models, extending the applicability of 3DGS to challenging scenarios where accurate point clouds are difficult to obtain.

Acknowledgment: This work was partially supported by Key R&D Program of Zhejiang Province (No.2023C01039) and NSF of China (No. 62425209).

References

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2): 4606–4613, 2022. 1

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 2

[3] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 1034–1040. IEEE, 1997. 1

[4] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 1, 2

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 6, 7, 8

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 1, 2

[7] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 2

[8] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3. 1–a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 4

[9] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2024. 4, 3

[10] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2024. 2

[11] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024. 5, 8, 4

[12] Yalda Foroutan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Evaluating alternatives to sfm point cloud initialization for gaussian splatting. *arXiv preprint arXiv:2404.12547*, 2024. 2

[13] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *European Conference on Computer Vision*, pages 368–381, 2010. 2

[14] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting, 2024. 2, 4, 5, 6, 7

[15] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021. 2

[16] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jiaolong Yang, Seungryong Kim, and Chong Luo. Unifying correspondence pose and nerf for generalized pose-free novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20196–20206, 2024. 2

[17] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[18] Jaewoo Jung, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park, and Seungryong Kim. Relaxing accurate initialization constraint for 3d gaussian splatting. *arXiv preprint arXiv:2403.09413*, 2024. 2, 3, 6, 7

[19] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 4, 5, 6

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 6, 7, 8

[21] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2025. 2, 8

[22] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 6, 7

[23] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. 2

[24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[25] Chongshan Lu, Fukun Yin, Xin Chen, Wen Liu, Tao Chen, Gang Yu, and Jiayuan Fan. A large-scale outdoor multimodal dataset and benchmark for novel view synthesis

and implicit scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7557–7567, 2023. 6, 7, 8

[26] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2

[27] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 1

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1, 2

[30] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[31] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. *arXiv preprint arXiv:2403.13806*, 2024. 2

[32] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 6, 3

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

[34] Jiuhn Song, Seonghoon Park, Honggyu An, Seokju Cho, Min-Seop Kwak, Sungjin Cho, and Seungryong Kim. Därf: Boosting radiance fields from sparse input views with monocular depth adaptation. *Advances in Neural Information Processing Systems*, 36:68458–68470, 2023. 2

[35] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5459–5469, 2022. 2

[36] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *ICLR*, 2022. 5

[37] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024. 4

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

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[40] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*, 2023. 2

[41] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2

[42] Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, et al. Vr-nerf: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. 1

[43] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 4

[44] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023. 2

[45] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 4

[46] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[47] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36:32215–32234, 2023. 2

[48] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36:32215–32234, 2023. 5

[49] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocu-

lar geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 4

[50] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. NIS-SLAM: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, 30(11):7129–7139, 2024. 2

[51] Hongjia Zhai, Hai Li, Zhenzhe Li, Xiaokun Pan, Yijia He, and Guofeng Zhang. PanoGS: Gaussian-based panoptic segmentation for 3d open vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14114–14124, 2025. 2

[52] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. SplatLoc: 3d gaussian splatting-based visual localization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 31(5):3591–3601, 2025. 1

[53] Hongjia Zhai, Boming Zhao, Hai Li, Xiaokun Pan, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. NeuraLoc: Visual localization in neural implicit map with dual complementary features. In *IEEE International Conference on Robotics and Automation*, 2025. 2

[54] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision*, pages 592–611. Springer, 2022. 2

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[56] Xiaoyu Zhang, Weihong Pan, Chong Bao, Xiyu Zhang, Xiaojun Xiang, Hanqing Jiang, and Hujun Bao. Lookcloser: Frequency-aware radiance field for tiny-detail scene. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16122–16132, 2025. 2

[57] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. *arXiv preprint arXiv:2403.15530*, 2024. 2

[58] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. 1

[59] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025. 2