SparseGS: Sparse View Synthesis using 3D Gaussian Splatting

Haolin Xiong[1,3,†,*] Sairisheek Muttukuru[1,*] Hanyuan Xiao[2,3] Rishi Upadhyay[1]
Pradyumna Chari[1] Yajie Zhao[2,3] Achuta Kadambi[1]

[1]University of California, Los Angeles [2]University of Southern California
[3]USC Institute for Creative Technologies

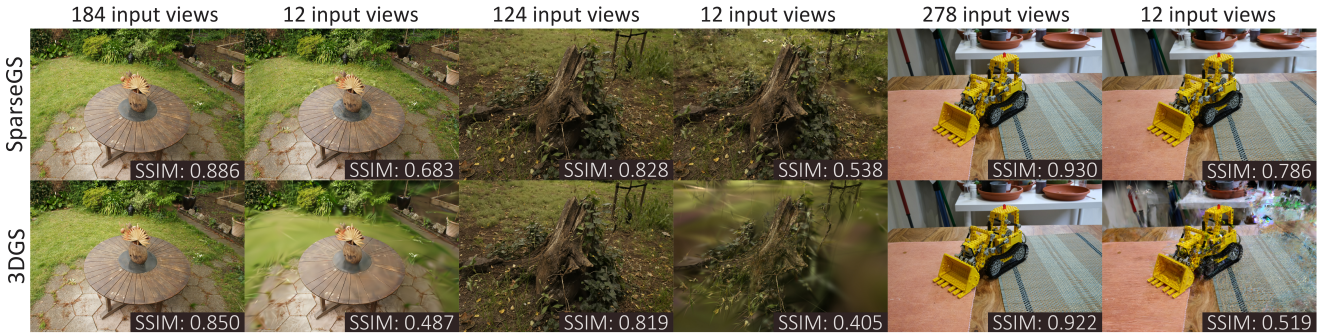

Figure 1. **The quality of 3DGS [11] degrades as the number of input views decreases**, particularly in unbounded scenes. SparseGS significantly improves novel view synthesis quality in sparse-input settings while maintaining fast training and real-time rendering.

Abstract

3D Gaussian Splatting (3DGS) has recently enabled real-time rendering of unbounded 3D scenes for novel view synthesis. However, this technique requires dense training views to accurately reconstruct 3D geometry. A limited number of input views will significantly degrade reconstruction quality, resulting in artifacts such as "floaters" and "background collapse" at unseen viewpoints. In this work, we introduce SparseGS, an efficient training pipeline designed to address the limitations of 3DGS in scenarios with sparse training views. SparseGS incorporates depth priors, novel depth rendering techniques, and a pruning heuristic to mitigate floater artifacts, alongside an Unseen Viewpoint Regularization module to alleviate background collapses. Our extensive evaluations on the Mip-NeRF360, LLFF, and DTU datasets demonstrate that SparseGS achieves high-quality reconstruction in both unbounded and forward-facing scenarios, with as few as 12 and 3 input images, respectively, while maintaining fast training and real-time rendering capabilities.

1. Introduction

The challenge of learning 3D representations from 2D images has been a longstanding area of interest, but achieving a balance between efficiency and fidelity remains a persistent challenge. While Neural Radiance Fields (NeRFs) excel in high-quality rendering and effectively represent anisotropic effects in view interpolation, they suffer from long training times, and blurriness if only sparse views are provided as input. The recent development of 3D Gaussian Splatting (3DGS) has substantially reduced the training cost by introducing a more compact, explicit 3D representation coupled with a real-time rendering pipeline. However, 3DGS still suffers from artifacts caused by the inherent ambiguity in projection from 3D to 2D posed by sparse input views.

Artifacts in 3DGS, such as "floaters" (high-density floating regions due to misplaced Gaussians) and "background collapse" (caused by Gaussians being misplaced at incorrect depths, resulting in background Gaussians appearing in the foreground) tend to be more pronounced than those in NeRFs. These issues are further exacerbated when the training set lacks substantial scene coverage, such as in multi-view unbounded scenes [2] (referred as 360-degree scenes in the rest of this paper). Extensions of 3DGS, such as [13, 47], have attempted to incorporate depth priors as geometry supervisions or regularizers, but they still fail to

∗ Equal contribution.
† Corresponding author. {xiongh@ucla.edu}

resolve the problem of floaters, particularly in unbounded scenes. We observe that naively applying the same alpha-blending equation for rendering depth can cause gradients to propagate to the wrong Gaussians, adversely affecting quality. Additionally, only providing extra guidance from the training views does not mitigate the problem of overfitting, thus failing to address background collapse issues in sparse-input settings.

Our objective is to accurately reconstruct 360-degree unbounded 3D scenes using as few as 12 input images. To address the aforementioned "floater" and "background collapse" issues in 3DGS representations, we introduce three key modules. First, we propose two novel depth rendering techniques (softmax-scaling depth and mode-selection depth) that go beyond the widely used alpha-blending depth to more effectively manage floaters. Next, we introduce a module designed to tackle background collapse by leveraging a 2D generative diffusion prior [15, 25] and depth warping [21, 42]. Finally, capitalizing on the explicit nature of 3DGS representations, which allows for direct scene manipulation, we present a floater pruning procedure to identify and eliminate undesired Gaussians. Combined, our pipeline achieves state-of-the-art (SOTA) performance in sparse-input novel view synthesis (NVS) problems, not only on forward-facing datasets but also on 360-degree unbounded scenes, a scenario that most current few-shot techniques [13, 23, 38] struggle to handle effectively.

In summary, our contributions are:

1. We propose a novel framework, **SparseGS**, for training coherent and robust 3D Gaussian representations from limited inputs, outperforming SOTA methods in sparse view synthesis.
2. SparseGS addresses the "floater" issue by introducing mode-selection and softmax-scaling depth rendering techniques.
3. SparseGS introduces an Unseen Viewpoint Regularization module, which mitigates overfitting by regularizing 3DGS training at viewpoints different from the input views. Empirically, the module reduces "background collapse" in sparse-input settings.
4. SparseGS also introduces an explicit and adaptive operator to further prune undesirable floating artifacts in 3DGS.

2. Related Work

We focus on neural representations of 3D scenes, particularly radiance fields, considering their offered detail level and scalability in novel view synthesis Additionally, we explore related works aimed at enhancing rendering quality and addressing artifacts when only sparse view images are provided as input.

Radiance Fields. Neural radiance field (NeRFs) as a 3D scene representation was first introduced by [19]. NeRF learns a continuous field of density and color that naturally interpolates high-dimensional appearance features in an differentiable way. The volumetric rendering equation is

$$C = \sum_{i=1}^{N} T_i \sigma_i c_i, \tag{1}$$

$$\text{with} \quad \alpha_i = (1 - \exp(-\sigma_i \delta_i)), T_i = \prod_{j=1}^{i-1} (1 - \alpha_i), \tag{2}$$

where C is the aggregated color along a ray, T_i is the transmittance of a sampled point i along the ray, σ_i is the density of i, and c_i is the color of i. A flurry group of extension works introduce more challenging datasets [12, 20], improve training speed [22] or quality [1], and apply NeRF to other tasks [40, 45]. Among them, Mip-NeRF [1] introduced the concept of anti-aliasing to NeRFs by rendering conical frustums rather than rays. As a result, Mip-NeRF is able to anti-alias renderings at consistently high quality at focal lengths different from training views. Mip-NeRF 360 [2] followed up Mip-NeRF that specifically tackled the bottleneck in 360° unbounded scene representation. In this work, we evaluate on both forward-facing scenes and 360° unbounded scenes in challenging sparse-view input task.

3D Gaussian Splatting (3DGS) [11] introduced a point-based differentiable 3D representation that achieves competitive rendering quality, training time and also real-time rendering in high resolution. The points with volume (or splats) are parameterized by their position, rotation, scaling, opacity, and a set of spherical harmonics coefficients for view-dependent color. Specifically, the geometry of splats is defined by a full 3D covariance matrix that is composed by a scaling matrix and a rotation matrix. The differentiable settings enable training with simple color supervision same as NeRFs. 3DGS advances NeRF in scalability of the representation, where NeRF cannot encode large scenes with limited number of parameters in a MLP. In addition, 3D Gaussian representations are explicit rather than the implicit representations of NeRFs, which allows for more direct editing and easier interpretability. We leverage this property for our technique which identifies and directly deletes floaters.

Recent works have built on top of 3D Gaussian Splatting to perform a variety of downstream tasks including text-to-3D generation [36, 44], dynamic scene representation [16, 41], and animating humans [48].

Few Shot Novel View Synthesis. The problem of novel view synthesis from few images has received significant interest as many view synthesis techniques can require a prohibitively high number of views for real-world usage. Early techniques [32, 37, 46] leverage multi-plane images (MPIs), which represent images by sub-images at different

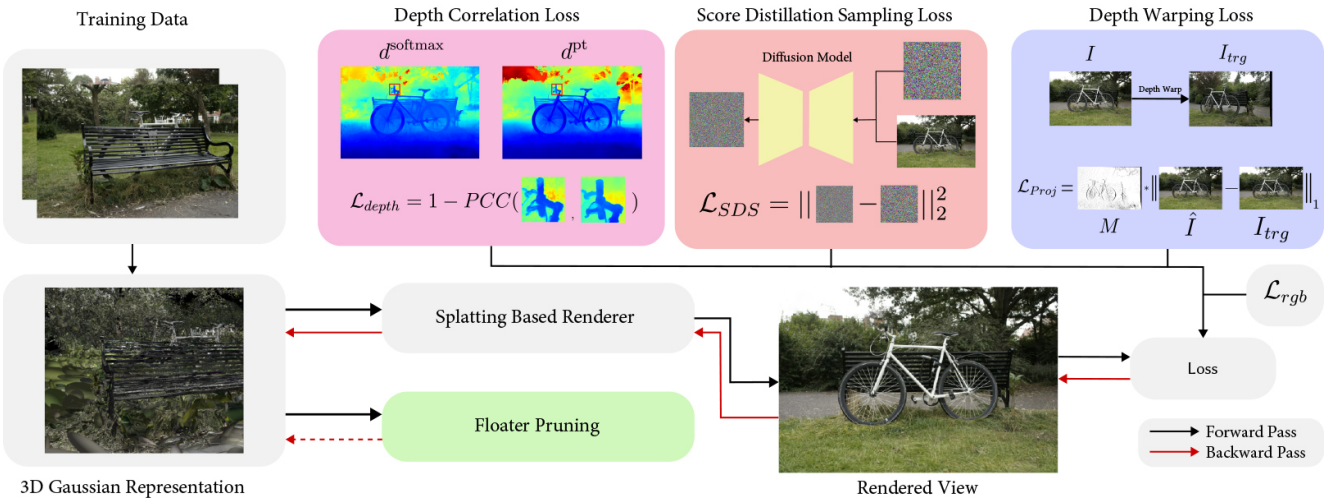

Figure 2. **Our proposed pipeline incorporates depth priors, diffusion constraints, and a floater pruning technique to improve few-shot novel view synthesis performance.** During training, we render the softmax depth and use Pearson correlation to encourage it to align with d^{pt} (Sec. 3.2). We also generate novel views using the procedure (Sec. 3.3.1) and incorporate a Score Distillation Sampling loss. At pre-set intervals, we prune floaters according to the Advanced Floater Removal procedure described in Sec. 3.4.

depths, to re-render depth and color from novel view points using traditional transformations. More recent techniques are built on top of the NeRF framework and tend to tackle the problem in one of two ways: The first set of methods introduce constraints on the variation between views. An early example of this type of method was DietNeRF [7] which added constraints to ensure that high-level semantic features remained the same from different views since they contained the same object. Another example is RegNeRF [23] which applies both color and depth consistency losses to the outputs at novel views. The second set of methods approach the problem by adding depth priors to novel views to regularize outputs. SparseNeRF [38] falls into this category and uses a pre-trained depth estimation model to get pseudo-ground truth depth maps which are then used for a local depth ranking loss. They additionally apply a depth smoothness loss to encourage rendered depth maps to be piecewise-smooth. DSNeRF [3] also uses additional depth supervision, but uses the outputs from a Structure-From-Motion(SFM) pipeline (typically COLMAP [30]) instead of a pre-trained model. Other than the techniques above, Neo 360 [6] uses tri-planes to represent a bounded scene more efficiently. [13] tackles the sparse-view 3DGS reconstruction with depth regularization and monocular depth estimation. Concurrent work [47] introduces a more advanced densification strategy tailored to 3DGS and leverages a similar zero-shot depth estimator [26, 27]. To the best of our knowledge, [6, 47] are the only existing methods that explicitly tackle the problem of 360° few-shot novel-view synthesis.

3. Methods

Overview. Our method consists of three key components designed to function cohesively to improve view consistency and depth accuracy in novel view synthesis: a depth correlation loss, an Unseen Viewpoint Regularization (UVR) module, and a floater pruning operation. In the following section, we first discuss three different ways of rendering depth from 3DGS scenes, followed by a patch-based depth loss. Then, we dissect the UVR module into two parts: a Score Distillation Sampling (SDS) loss and a depth warping loss, which are designed for regularizing viewpoints distant and close to training cameras, respectively. Finally, we introduce a floater pruning procedure, which utilizes depth to identify and remove misplaced Gaussians ("floaters"). Fig. 2 showcases a high-level architecture of our pipeline.

3.1. Mode-selection & Softmax-scaling Depth Rendering

Alpha-blending is a widely used technique in NeRFs [19] for rendering depth maps. Applying the same method to 3DGS, the alpha-blending depth at pixel (x, y), denotes as $d^{\text{alpha}}_{x,y}$, is calculated as:

$$d^{\text{alpha}}_{x,y} = \sum_{i=1}^{N} T_i \alpha_i d_i \tag{3}$$

where T_i is the accumulated transmittance for the i-th Gaussian, α_i is the alpha-compositing weight, and d_i is the depth of the Gaussian. In this rendering approach, the depth value of a pixel is influenced by all Gaussians along their corresponding ray. As a result, the model might adjust the trans-

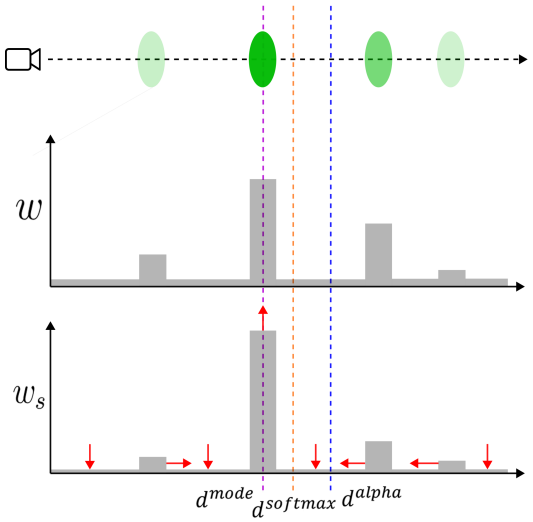

Figure 3. **A Demonstration of the Three Kinds of Depth.** The weights w_i are shown at the top, with the weights after applying softmax displayed as w_s below. Although a single Gaussian may have the highest weight, other nearby low-weight Gaussians still influence the depth calculation. By applying softmax-scaling, depth accumulation is biased toward Gaussians with higher weights. The Depth Pearson Correlation is less likely to distort the actual depth by manipulating the opacity of floaters. By the end of training, our model consolidates the low-weight Gaussians into a single Gaussian point at the correct depth, ensuring that all depth variations align.

mittance values of incorrectly placed Gaussians instead of altering their positions, making the 2D depth map appear accurate from the training angles despite the incorrect underlying 3D geometries. So, we introduce two different ways of rendering depth from 3DGS, which are used in later sections.

Mode-selection. In mode-selection, the depth of the Gaussian with the largest contributing $w_i = T_i\alpha_i$ represents the depth of that pixel. The mode-selected depth of a pixel can be written as:

$$d^{\text{mode}}_{x,y} = d_{\arg\max_i(w_i)}. \tag{4}$$

Intuitively, the mode-selected depth chooses the highest-opacity Gaussian, while the alpha-blended depth takes into account all Gaussians along the imaginary ray from the camera. A crucial insight for building our pruning operator is that d^{mode} and d^{alpha} should ideally be the same. Consider the toy setting on the left of Fig. 3. In this scenario, d^{mode} corresponds to the depth of the second Gaussian from the left, as it has the highest w_i. However, d^{alpha} appears slightly behind this Gaussian due to the influence of low-weight Gaussians with greater depths. This discrepancy can create ambiguity regarding the true depth. We interpret this ambiguity as a measure of the uncertainty of 3DGS at a given point and leverage it to guide our pruning operator introduced later.

Softmax-scaling. While rendering depth using the mode identifies the most significant Gaussians contributing to the depth, the $\arg\max$ operator restricts the gradient during backpropagation flow to only the Gaussian with the highest w_i. This approach can be problematic when a Gaussian closer to the viewer is translucent, as the weights of farther Gaussians should be non-zero. In order to overcome this limitation, we further introduce a softmax-scaling to modify alpha-blending:

$$d^{\text{softmax}}_{x,y} = \log\left(\frac{\sum_{i=1}^{N} w_i e^{\beta w_i} d_i}{\sum_{i=1}^{N} w_i e^{\beta w_i}}\right), \tag{5}$$

where the β parameter allows us to modulate the softmax temperature and thereby, to choose a desired amplification of highly weighted Gaussians. Note that:

$$\lim_{\beta\to 0} d^{\text{softmax}}_{x,y} = \log(d^{\text{alpha}}_{x,y}), \tag{6}$$

$$\lim_{\beta\to \infty} d^{\text{softmax}}_{x,y} = \log(d^{\text{mode}}_{x,y}). \tag{7}$$

With the softmax-scaling add-on, we can approximate the mode depth while still propagating gradients to Gaussians off the mode.

3.2. Patch-based Depth Correlation Loss

We compute pseudo-ground truth depth maps using pre-trained depth estimation models on the training views. We opt for the Marigold monocular depth estimator [10], though any pre-trained depth estimation model [17, 26, 27, 33] could work. We refer to the depth from this pre-trained model as d^{pt}.

Since the monocular estimation model predicts relative depth, while alpha-blending, softmax-scaling, and mode-selection depths are COLMAP-anchored, directly applying an L2 loss, such as mean squared error (MSE), would be ineffective. One option is to estimate scale and shift parameters to align the two depth maps in metric space. However, this transformation is not guaranteed to be constant across all pixels, and a naive alignment could introduce additional unwanted distortion. Instead, we adopt Pearson correlation across image patches to compute a similarity metric between depth maps. This approach is derived from a similar intuition as the depth ranking losses proposed by prior work [38], in that they both leverage relative depth to ascertain global depth. However, rather than comparing the ranks of two selected pixels per iteration, we compare between entire patches, allowing the model to influence larger portions of the image and learn more local structures. The Pearson correlation coefficient encourages patches at the same location in both depth maps to have high cross-correlation,

regardless of variations in depth value ranges. At each iteration, we randomly sample N non-overlapping patches to compute the depth correlation loss as:

$$\mathcal{L}_{\text{depth}} = \frac{1}{N}\sum_{i}^{N} 1 - \text{PCC}(p_i^{\text{softmax}}, p_i^{d^{\text{pt}}}) \tag{8}$$

where $p_i^{\text{softmax}} \in \mathbb{R}^{S^2}$ denotes the i-th patch of $d^{\text{softmax}}$and $p_i^{\text{pt}} \in \mathbb{R}^{S^2}$ denotes the i-th patch of d^{pt}, with the patch size S being a hyper-parameter empirically chosen to be 1/100 of the image resolution. Additionally, a global Pearson depth loss should be applied to prevent depth discontinuities from appearing at patch edges. Intuitively, this loss works to align the softmax-scaling depth maps of the Gaussian representation with d^{pt} while avoiding the problem of inconsistent scale and shift.

3.3. Unseen Viewpoints Regularization (UVR)

In this section, we propose two regularization methods to improve reconstruction from novel viewpoints:
1). Score Distillation Sampling (SDS) Loss: This loss leverages a large vision model [28] to guide training from viewpoints that are distant from the original training cameras, ensuring that the renders appear more natural.
2). Depth Warping Loss: This loss utilizes monocular-estimated depth to warp training images to nearby angles, effectively generating additional pseudo-training cameras from the re-projected views.

3.3.1 Score Distillation Sampling Loss

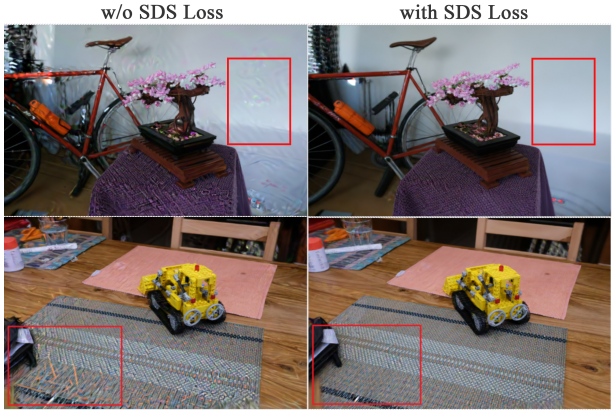

Figure 4. **Illustration of benefits from the SDS loss.** While the scene structure is well preserved, the high-frequency noise in both geometry and texture is significantly reduced (red box).

In the sparse-view setting, Gaussians that are well constrained under input viewpoints often appear as small fragmentation rendered from other sampled viewpoints. This issue is caused by misplaced Gaussians or loosely constrained high-dimensional color representations. Simple smoothing techniques, such as Laplacian smoothing, lead to loss of sharpness in detail. Inspired by recent diffusion models [4, 8, 24, 25, 29, 43] and Score Distillation Sampling (SDS) [36] for zero-shot 3D reconstruction [5, 14, 15, 34], we propose using a pre-trained 2D diffusion prior to refine 3DGS reconstruction. The diffusion prior is expected to preserve meaningful image details (in-distribution modeled by diffusion model) while removing non-photorealistic artifacts (out-of-distribution modeled by diffusion model).

Specifically, we implement a strategy that samples random viewpoints around the center of scene estimated from input cameras. Then, the renderings at the sampled viewpoints are encoded and decoded by the diffusion model, where the predicted noise is then supervised with our SDS loss, formulated as:

$$\hat{I} = \mathcal{N}(\sqrt{\hat{\alpha}}I', (1-\hat{\alpha})\mathbf{I}), \tag{9}$$

$$\mathcal{L}_{\text{SDS}} = \nabla_G \mathbb{E}[(\epsilon_\phi(\hat{I}_p; \tilde{I}_p) - \epsilon)\frac{\partial \hat{I}_p}{\partial G}], \tag{10}$$

where G represents the parameters of our Gaussian representation, $\hat{a}$ represents the cumulative product of one minus the variance schedule, ϵ is the sampled noise, and $\epsilon_\phi(\cdot)$ is the predicted noise by the diffusion model. $\hat{I}_p$ denotes the rendered image at camera pose p with added noise, encoded by the diffusion model, and $\tilde{I}_p$ is the denoised image. We compare rendering results with and without our SDS loss in Fig. 4, where the module successfully removes high-frequency artifacts while leaving the scene structure untouched.

3.3.2 Depth Warping Loss

3DGS operates by fitting to the training views, with the expectation that it will learn the underlying geometry of the scene. However, in sparse-input settings, the model tends to excessively overfit to the limited data, leading to significant background collapse even with slight changes in viewing angles. Regularizing the training views with depth prior helps but is not enough in extreme sparse-view scenarios. To address this, we employ image re-projection, utilizing established depth warping techniques [21, 42] to augment the training data by re-projecting images to nearby viewpoints.

Unlike previous methods [3] that rely on depth maps generated from COLMAP, which often suffer from noise due to spurious correspondences, noisy camera parameters, or poor COLMAP optimization, we have discovered that scaling the relative range of monocular-estimated depth maps to match the range of rendered alpha-blending depth maps is sufficient for depth warping. Consequently, the quality of the warping heavily depends on the quality of $\min_i d_i^{\text{alpha}}$, $\max_i d_i^{\text{alpha}}$, and d^{pt}. However, in practice, with

the current monocular estimation model [10], we have observed that most areas in the warped results are stable, provided that the aforementioned Pearson depth loss has converged to a reasonable level.

Mathematically, we define our image re-projection as follows: For pixel $p_i(x_i, y_i)$ in training image I_{src}, the warping to the corresponding pixel $p_j(x_j, y_j)$ at an unseen viewpoint I_{trg} can be formulated as:

$$p_j = K_{\text{trg}} T (K_{\text{src}}^{-1} Z_i p_i), \tag{11}$$

where Z_i represents the monocular-estimated depth map scaled to the range of $\min_i d_i^{\text{alpha}}$ and $\max_i d_i^{\text{alpha}}$. K_{trg} and K_{src} are the intrinsic matrices of the source and the target camera, and T refers to the transformation between camera poses from viewpoint I_{src} to I_{trg}. A warp mask M indicating which pixels are validly warped is generated along the process. Finally, the Depth Warping loss $\mathcal{L}_{\text{Proj}}$ is formulated as:

$$\mathcal{L}_{\text{Proj}} = M * \mathcal{L}_1(\hat{I}, I_{\text{trg}}). \tag{12}$$

3.4. Advanced Floater Pruning

Because the softmax depth loss is a soft constraint, there may exist regions where d^{mode} and d^{alpha} do not align. As a result, some floaters may remain along the rays of the input views. Therefore, we propose a novel pruning operator to remove the Gaussians at false modes at the end of training.

We generate a mask F_i for each training view i by calculating Δ_i , which represents the per-pixel relative difference between the depth obtained from mode selection and the depth from alpha blending. Upon examining the distribution of Δ_i, we observed that images containing numerous floaters typically exhibit bimodal histograms. This is because floaters are usually positioned significantly away from the true depth. Conversely, images without floaters tend to have more unimodal histograms. This difference is illustrated in Fig. 5(c). Based on this observation, we apply a threshold to Δ_i to make the distribution more unimodal, thereby minimizing the presence of floaters. An example of a floater mask is depicted in Fig. 5(d). After generating the mask, we proceed to identify and prune all Gaussians, including the mode Gaussian. The impact of this pruning is demonstrated in Fig. 5(b). The complete pruning algorithm is detailed in the supplementary materials.

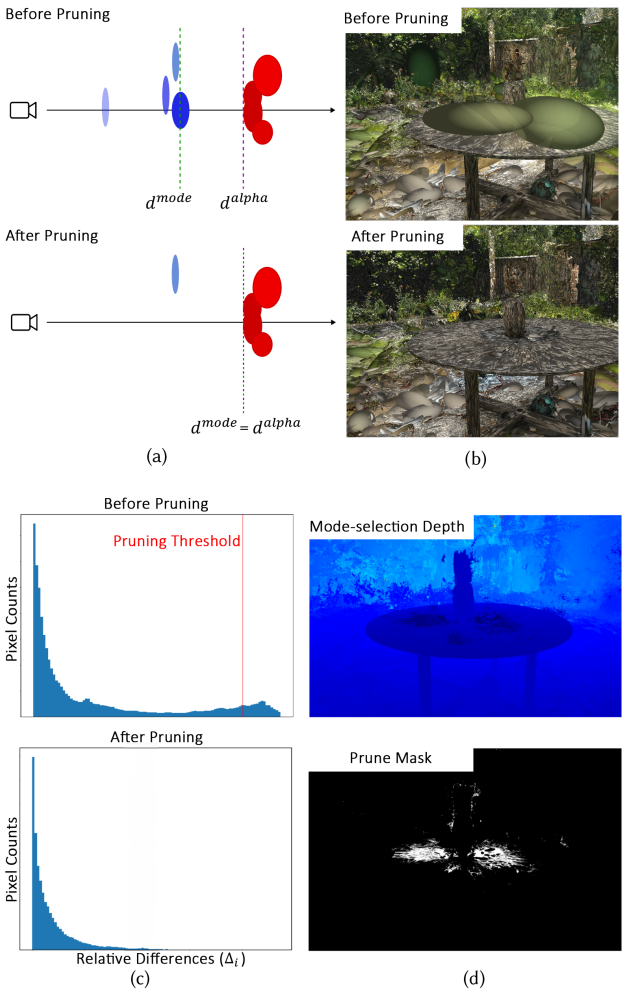

Figure 5. **The proposed floater pruning technique removes Gaussians at inaccurate depths.** An example (a) demonstrates our pruning method: before pruning, there are floaters (blue) in front of the Gaussians at the object surface (red) and therefore, d^{mode} d^{alpha} are not aligned. The pruning method removes all Gaussians on that pixel before the mode and as a result, $d^{\text{mode}} = d^{\text{alpha}}$. Applied to the garden scene (b), the pruning method removes large floaters at the center and left of the scene. We see the relation between the bimodality of the histogram of relative differences between d^{mode} and d^{mode}, before and after the pruning operation in (c), with the appropriate pruning cutoff points indicated by the red vertical line. (d) The prune mask F_i (bottom) is obtained from thresholding the relative differencing of the mode depth d^{mode} (top) at the cutoff points.

4. Experiments

4.1. Experimental Settings

Datasets. We conduct our experiments on three datasets, categorized into two settings: 1) the Mip-NeRF360 [2] dataset, which features seven challenging 360° scenes; 2) the LLFF [18] and DTU [9] datasets, which contain forward-facing scenes. For the Mip-NeRF360 dataset, we use every eighth image as testing view and evenly sample 12 or 24 views from the remaining views as the training set. For the LLFF and DTU datasets, we follow the training and evaluation protocols established in RegNeRF [23]. Following conventions, all input images are downscaled to 1/4 of the original height and width.

Implementation Details. We obtain the camera calibration of the input views and the initial point cloud for 3DGS us-

Models	12-view PSNR ↑	12-view LPIPS ↓	12-view SSIM ↑	24-view PSNR ↑	24-view LPIPS ↓	24-view SSIM ↑
Mip-NeRF 360	17.73	0.520	0.432	19.78	0.431	0.530
RegNeRF	18.84	0.544	0.437	20.55	0.398	0.546
SparseNeRF	17.44	0.609	0.395	21.13	0.389	0.600
3DGS	17.49	0.431	0.499	19.93	0.401	0.588
DNGaussian	16.28	0.549	0.432	19.26	0.440	0.550
FSGS (w/o MVS)	18.15	0.485	0.504	21.76	0.396	0.626
SparseGS (w/o MVS)	18.46	0.476	0.513	22.03	0.381	0.631
FSGS	19.31	0.413	0.574	22.82	0.293	0.693
SparseGS (Ours)	19.37	0.398	0.577	23.02	0.290	0.713

Table 1. **Quantitative comparison on Mip-NeRF360 dataset for 12/24 input-view settings.**

ing COLMAP [30, 31]. Specifically, the initial point cloud is output from the multi-view stereo (MVS) step. We train 10k iterations in 12-view settings and 30k iterations in 24-view settings. The depth correlation loss is applied with a patch size of 64×64 pixels for the MipNeRF360 dataset and 32×32 pixels for other datasets. The β for the softmax depth is set to 5. Floater pruning is applied once after training completes. In general, the depth correlation loss converges after 3k iterations. Meanwhile, alpha-blending depth maps are rendered for image re-projection. For each training view, we generate four extra warping viewpoints by rotating the camera around the center of scene.

We use Stable-Diffusion version 2.1 for our diffusion model in Sec. 3.3.1. The SDS loss is enabled after two-thirds of the total iterations, when the majority of scene structure has been stable.

We set the depth warping loss weight to 0.05, the SDS loss weight to 5×10^{-4}, the local and global Pearson loss weights both to 0.15.

4.2. Comparison

Unbounded Dataset. We use the Mip-NeRF360 dataset to evaluate 3D reconstruction of unbounded 360° scenes. The dataset presents significant challenges due to inherent complexity of scenes and strong occlusions under sparse-view settings. Moreover, the initial point cloud from COLMAP can be very limited even with the MVS step. In order to prove robustness of our method, we also evaluate performance with even sparser point clouds output by Structure From Motion (i.e., without the MVS step) as initialization. In quantitative evaluation from now on, we compute peak signal-to-noise ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS) and structural similarity index measure (SSIM) metrics. As shown in Tab. 1, SparseGS significantly outperforms previous NeRF-based methods and concurrent works, FSGS and DNGaussian, in both 12-view and 24-view settings.

Forward-facing Datasets. We also provide evaluations on the forward-facing datasets (LLFF and DTU) to demonstrate robustness of our pipeline. The LLFF dataset comprises eight complex forward-facing real scenes, while the DTU dataset includes object-centric scenes with foreground masks. It is important to note that in under-reconstructed regions, where input coverage is insufficient, NeRF-based methods often produce overly smooth appearances. This limitation actually prompted the introduction of positional encoding [19, 35]. In contrast, 3DGS-based representations gravitate towards placing isolated Gaussians, appearing as high-frequency artifacts. PSNR tends to tolerate overly smooth images and heavily penalizes sharp artifacts, while perceptual metrics like LPIPS emphasize the opposite [39]. This can lead to minor discrepancies between the two metrics on certain test cases in Tab. 2. In general, SparseGS is competitive against previous and concurrent SOTA methods in both datasets.

Models	LLFF PSNR ↑	LLFF LPIPS ↓	LLFF SSIM ↑	DTU PSNR ↑	DTU LPIPS ↓	DTU SSIM ↑
Mip-NeRF	15.22	0.540	0.351	16.71	0.239	0.757
RegNeRF	18.06	0.411	0.535	18.89	0.190	0.745
SparseNeRF	19.07	0.401	0.564	19.55	0.201	0.769
3DGS	16.94	0.402	0.488	14.18	0.301	0.628
FSGS	19.88	0.340	0.612	18.36	0.232	0.707
DNGaussian	19.12	0.294	0.591	18.91	0.176	0.790
SparseGS (Ours)	19.86	0.322	0.668	18.89	0.178	0.834

Table 2. **Quantitative comparison on forward-facing datasets.**

Depth PCC: Alpha-blending	Depth PCC: Softmax-scaling	UVR	Pruning	PSNR ↑	LPIPS ↓	SSIM ↑
				17.49	0.431	0.499
✓				18.45	0.410	0.549
	✓			18.86	0.401	0.561
	✓	✓		19.09	0.400	0.565
	✓	✓	✓	**19.37**	**0.398**	**0.577**

Table 3. **Ablation Studies.** We ablate our components on the Mip-NeRF360 dataset under 12-view setting.

4.3. Ablation Studies

We ablate our method on the Mip-NeRF360 dataset 12-view setting. Quantitative results are reported in Tab. 3. Qualitative figures are shown in Fig. 7.

Depth Correlation Loss. The depth Pearson Correlation loss introduces depth knowledge into the 3DGS representation. We show the improvements by applying this loss to both alpha-blending and softmax-scaling depth. As indicated in Tab. 3, our softmax-scaling depth rendering method performs better, improving PSNR by 1.37dB compared to 3DGS and significantly enhance the quality of the rendered depth map Fig. 7.

Unseen Viewpoints Regularization. The UVR module adds regularization from viewpoints away from the training viewpoints, reducing the problem of excessive overfitting in sparse-input settings. It effectively reduces high-frequency artifacts, improving PSNR by an average 0.33dB.

Floater Pruning. The floater pruning heuristic further cleans up the scene by deleting misplaced low-opacity Gaussians, making both the image and the depth map sharper. This module further boosts PSNR by 0.28dB.

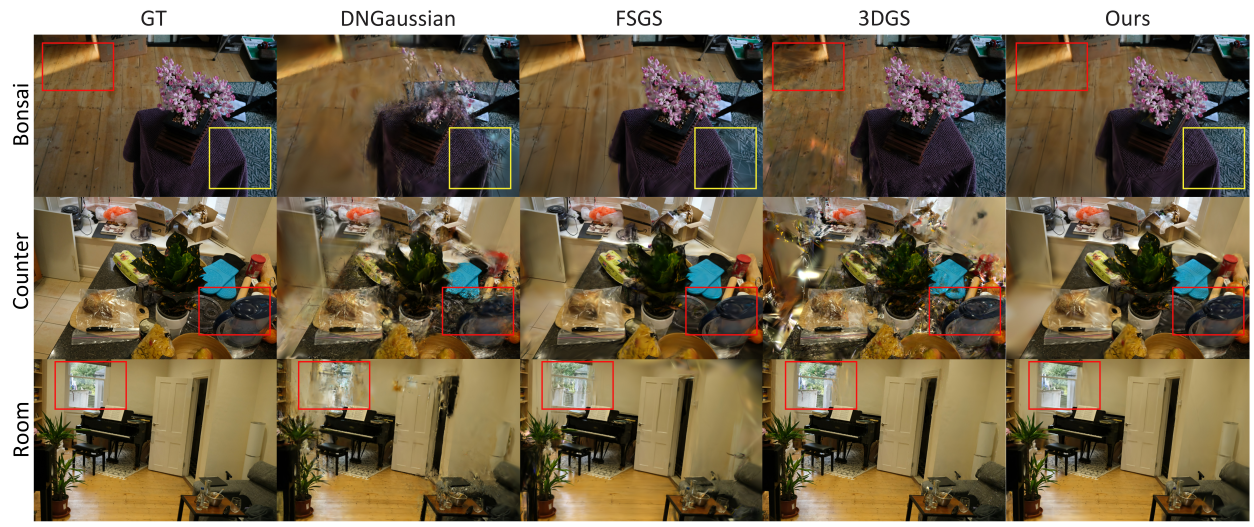

Figure 6. **Qualitative evaluation on the Mip-NeRF 360 dataset.** Our model effectively reconstructs high-frequency geometry, preserving sharp boundaries between the subject and the background. In contrast, FSGS excels in preserving fine details due to its densification technique but fails to reconstruct background geometry. On the other hand, DNGaussian can render a coherent background, but foreground objects are incorrectly pruned out.

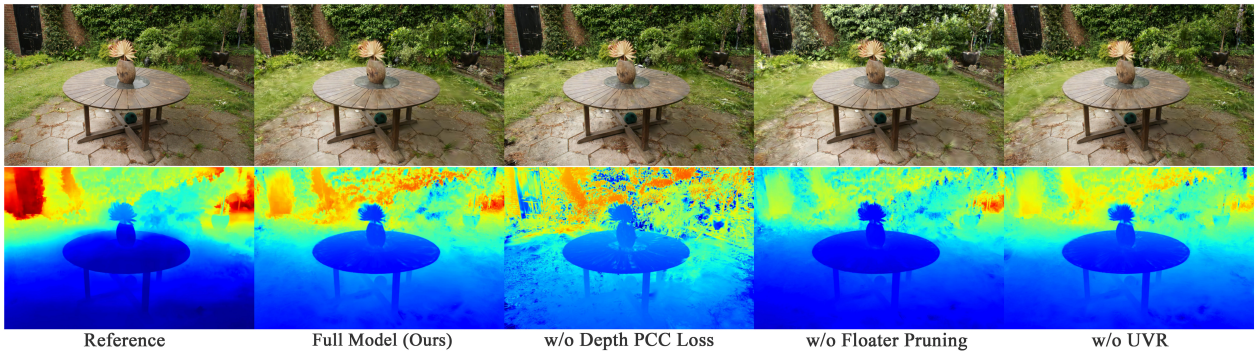

Figure 7. **Ablation studies.** The reference depth map is produced by a monocular depth estimation model. Our complete model outputs a cleaner and more consistent scene structure with less noise.

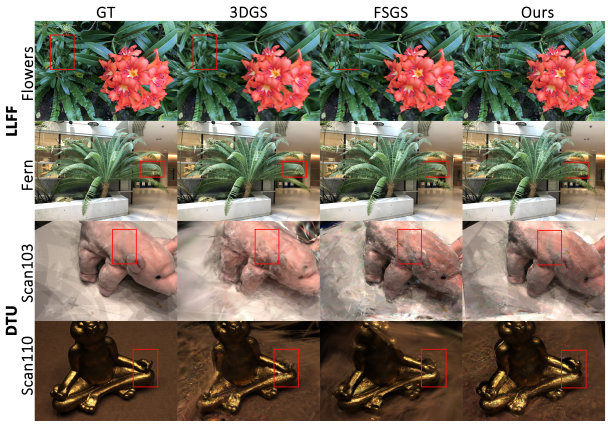

Figure 8. **Qualitative evaluation on the forward-facing datasets.** Our method is able to reconstruct high-frequency geometry more accurately than state-of-the-art [47].

5. Conclusion

In this paper, we propose a method using 3D Gaussian Splatting (3DGS) representation to tackle sparse-view 3D reconstruction task. We observe that the alpha blending depth rendering in original 3DGS often results in misplaced Gaussians (known as "floaters"). Therefore, we propose to constrain depth convergence with softmax-scaling and mode-selection bias that significantly reduce such floaters. In regions with little coverage by input views, we leverage Score Distillation Sampling (SDS) and Depth Warping to reduce collapse in geometry and noise in texture while preserving fine details. Lastly, we propose a novel floater pruning process to identify and remove low-opacity floaters. In evaluation, we show that our method outperforms the state-of-the-art methods by outputting a much cleaner and more coherent scene under even more challenging 12-view setting on Mip-NeRF360 dataset.

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. *ICCV*, 2021. 2

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 1, 2, 6

[3] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 5

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[5] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 5

[6] Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Vitor Guizilini, Thomas Kollar, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Neo 360: Neural fields for sparse view synthesis of outdoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9187–9198, 2023. 3

[7] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, 2021. 3

[8] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. 2022. 5

[9] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 6

[10] 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 (CVPR)*, 2024. 4, 6

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[12] Aviad Levis, Pratul P Srinivasan, Andrew A Chael, Ren Ng, and Katherine L Bouman. Gravitationally lensed black hole emission tomography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19841–19850, 2022. 2

[13] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. *arXiv preprint arXiv:2403.06912*, 2024. 1, 2, 3

[14] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 5

[15] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2, 5

[16] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 2

[17] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proc. CVPR*, 2021. 4

[18] 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)*, 2019. 6

[19] 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 *ECCV*, 2020. 2, 3, 7

[20] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. *CVPR*, 2022. 2

[21] Yuji Mori, Norishige Fukushima, Toshiaki Fujii, and Masayuki Tanimoto. View generation with 3d warping using depth information for ftv. In *2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 229–232, 2008. 2, 5

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2

[23] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 6

[24] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023. 5

[25] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 5

[26] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 3, 4

[27] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 3, 4

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 5

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 7

[31] 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. 7

[32] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. *CVPR*, 2019. 2

[33] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, et al. Ldm3d: Latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*, 2023. 4

[34] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[35] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 7

[36] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2, 5

[37] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2

[38] Guangcong Wang, 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, 3, 4

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 7

[40] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, 2022. 2

[41] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2

[42] 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. 2022. 2, 5

[43] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360° views. 2022. 5

[44] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*, 2023. 2

[45] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2

[46] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2

[47] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting, 2023. 1, 3, 8

[48] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. *arXiv preprint arXiv:2311.08581*, 2023. 2