DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields with Global-Local Depth Normalization

Jiahe Li[1], Jiawei Zhang[1], Xiao Bai[1]*, Jin Zheng[1], Xin Ning[2], Jun Zhou[3], Lin Gu[4,5]
[1]School of Computer Science and Engineering, State Key Laboratory of Complex & Critical Software Environment, Jiangxi Research Institute, Beihang University
[2]Institute of Semiconductors, Chinese Academy of Sciences
[3]School of Information and Communication Technology, Griffith University
[4]RIKEN AIP [5]The University of Tokyo

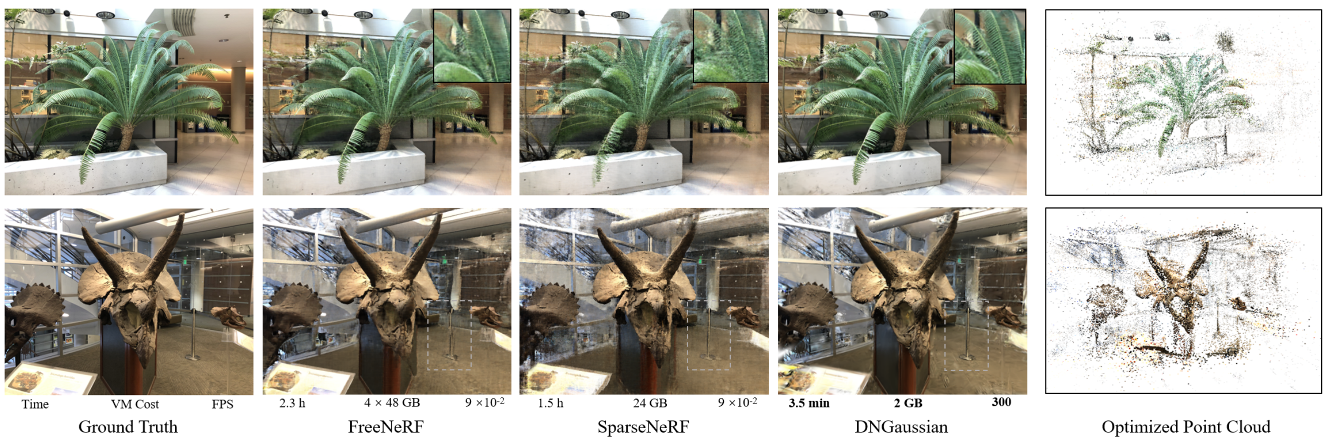

Figure 1. Comparison of the state-of-the-arts FreeNeRF [51] and SparseNeRF [41] with our DNGaussian utilizing three views for training. DNGaussian stands out by delivering comparably high-quality synthesized views and superior details with a remarkable 25× reduction in time and significantly lower memory overhead during training, while attaining the fastest and the only real-time rendering speed of 300 FPS. The point cloud of Gaussians illustrates the detailed and explainable spatial representation learned through our method.

Abstract

Radiance fields have demonstrated impressive performance in synthesizing novel views from sparse input views, yet prevailing methods suffer from high training costs and slow inference speed. This paper introduces DNGaussian, a depth-regularized framework based on 3D Gaussian radiance fields, offering real-time and high-quality few-shot novel view synthesis at low costs. Our motivation stems from the highly efficient representation and surprising quality of the recent 3D Gaussian Splatting, despite it will encounter a geometry degradation when input views decrease. In the Gaussian radiance fields, we find this degradation in scene geometry primarily lined to the positioning of Gaussian primitives and can be mitigated by depth constraint. Consequently, we propose a Hard and Soft Depth Regularization to restore accurate scene geometry under coarse monocular depth supervision while maintaining a fine-grained color appearance. To further refine detailed geometry reshaping, we introduce Global-Local Depth Normalization, enhancing the focus on small local depth changes. Extensive experiments on LLFF, DTU, and Blender datasets demonstrate that DNGaussian outperforms state-of-the-art methods, achieving comparable or better results with significantly reduced memory cost, a 25× *reduction in training time, and over* 3000× *faster rendering speed. Code is available at:* https://github.com/Fictionarry/DNGaussian.

1. Introduction

Novel view synthesis with sparse inputs poses a challenge for radiance fields. Recent advances in neural radiance fields (NeRF) have excelled in reconstructing photorealis-

*Corresponding author: Xiao Bai (baixiao@buaa.edu.cn).

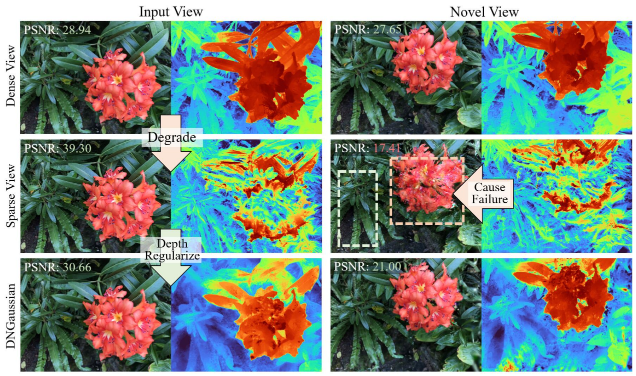

Figure 2. 3D Gaussian Splatting [17] exhibits its potential to reconstruct some fine details (green box) from sparse input views. Nevertheless, the reduced input views would significantly degrade geometry and cause failed reconstruction (orange box). After applying depth regularization, DNGaussian successfully recovers accurate geometry and synthesizes high-quality novel views.

tic appearance and accurate geometry from just a handful of input views [4, 15, 26, 34, 35, 41, 47, 51, 53]. However, most sparse-view NeRFs are implemented with low processing speed and substantial memory consumption, resulting in high time and computational costs that restrict their practical applications. While some methods [35, 37, 47] achieve faster inference speed with grid-based backbones [12, 25, 36], they often suffer from trade-offs, leading to either high training costs or compromised rendering quality.

Recently, 3D Gaussian Splatting [17] has introduced an unstructured 3D Gaussian radiance field, employing a set of 3D Gaussian primitives to achieve remarkable success in rapid, high-quality, and low-cost novel view synthesis, when learned from color dense input views. Even with only sparse inputs, it can still partially retain the surprising ability to reconstruct some clear and detailed local features. Nevertheless, the decrease in view constraints makes a significant portion of scene geometry be incorrectly learned, resulting in failures in novel view synthesis, as illustrated in Figure 2. Inspired by the success of earlier depth-regularized sparse-view NeRFs [35, 41], this paper explores distilling depth information from pre-trained monocular depth estimators to rectify the Gaussian fields of the ill-learned geometry, and introduce the Depth Normalization Regularized Sparse-view 3D Gaussian Radiance Fields (**DNGaussian**) to pursue higher quality and efficiency for few-shot novel view synthesis.

Despite sharing a similar form of depth rendering, the depth regularization for 3D Gaussian radiance fields differs significantly from that employed by NeRF. Firstly, existing depth regularization strategies for NeRFs commonly employ depth to regularize the entire model, which creates a potential geometry conflict in the Gaussian fields that adversely affects quality. Specifically, this practice forces the shape of Gaussians to fit the smooth monocular depth rather than the complex color appearance and thus results in loss of details and blurred appearance. Considering that the basis of scene geometry lies in the position of the Gaussian primitives rather than their shape, we freeze the shape parameters and propose a *Hard and Soft Depth Regularization* to enable spatial reshaping by encouraging movement among the primitives. During regularization, we propose rendering two types of depth to independently adjust the center and opacity of Gaussians without changing their shape, therefore striking a balance between the fitting of complex color appearance and smooth coarse depth.

Moreover, Gaussian radiance fields are more sensitive to small depth errors when compared to NeRF, which may result in a noisy distribution of primitives and failures in regions with complex textures. Existing scale-invariant depth losses often opt to align depth maps to a fixed scale, which leads to the overlook of small losses. To address this issue, we introduce the *Global-Local Depth Normalization* into the depth loss function, thus encouraging the learning of small local depth changes in a scale-invariant way. With the local and global scale normalization, our method guides the loss function to refocus on small local errors while maintaining knowledge on the absolute scale, to enhance the detailed geometry reshaping process for depth regularization.

Integrating the two proposed techniques, DNGaussian synthesizes views with competitive quality and superior details compared to state-of-the-art methods in multiple sparse-view settings on LLFF, Blender, and DTU datasets. This advantage is further enriched by substantially lower memory costs, 25$\times$ reduction of training time, and over 3000$\times$ faster rendering speed. The experiments also demonstrate our method's universal ability to fit complex scenes, wide-ranging views, and multiple materials.

Our main contributions are the following:

- A Hard and Soft Depth Regularization to constrain the geometry of 3D Gaussian radiance fields by encouraging the movement of Gaussians, which enables the coarse-depth regularized space reshaping without compromising fine-grained color performance.
- A Global-Local Depth Normalization that normalizes depth patches on local scales to achieve a refocus on small local depth changes, thereby improving the reconstruction of detail appearance for 3D Gaussian radiance fields.
- A DNGaussian framework for fast and high-quality few-shot novel view synthesis, which combines the above two techniques and achieves competitive quality across multiple benchmarks compared to the state-of-the-art methods, excelling in capturing details with significantly lower training costs and real-time rendering.

To the best of our knowledge, we are the first attempt to analyze and address the depth regularization problem for 3D Gaussian Splatting under coarse depth cues. We hope this paper can inspire more ideas for optimizing radiance fields in under-constrained situations.

2. Related Work

Radiance Fields for Novel View Synthesis. Novel view synthesis aims to generate unseen views of the same object or scene from a set of given images [1, 58]. Neural Radiance Fields (NeRF) [24] uses a large MLP to represent 3D scenes and renders via volume rendering. However, its speed is slow both in training and inference. The following improvements mainly pursue either higher quality [2, 3] or efficiency [5, 11, 14, 20, 25, 36, 52], but hard to achieve both. The most recent unstructured radiance fields [6, 17, 50] utilize a set of primitives to represent scenes. Among them, 3D Gaussian Splatting [17] represents radiance fields by a set of anisotropic 3D Gaussians and renders with a differentiable splatting. This approach achieves great success in fast and high-quality reconstruction for complex real scenes. While this method excels with dense input views and has achieved success in various 3D tasks [22, 38, 46], its reconstruction with sparse view inputs remains an open problem. Also, issues such as how to apply additional constraints for improvement are still unsolved and worthy of discussion.

Few-shot Novel View Synthesis. Few-shot novel view synthesis aims to generate novel views from only a set of sparse input views. Many works address the problem by introducing regularization strategies specified for NeRF [10, 18, 26, 51]. Some pre-trained methods aim to design a generative model and train it on large datasets [4, 8, 19, 53, 60], while others [15, 47] take pre-trained models as a type of loss to regularize the training process with well-learned knowledge. Depth distilling [10, 30, 35, 41] is also a powerful technique for sparse-view neural fields. However, limited by their powerful but slow backbones or the complex pre-trained models, most of these methods are costly in both training and inference. Although some methods [35, 37, 47] have improved inference efficiency via grid-based backbones, they also suffer from trade-offs like higher training costs or lower quality. More recently, some work [21, 27, 31] enable zero-shot novel view synthesis with even one input by diffusion model priors, but can hardly handle complex scenes and with lower efficiency.

Depth Supervision in Sparse-view Neural Fields. As a classic cue in many 3D vision tasks [40, 42, 43, 45, 56, 59], depth information has been widely used to supervise sparse-view neural fields. The first group [10, 30] is to extract accurate but sparse depth values from reliable point clouds, and the second [13, 35, 39, 41, 54] distills depth knowledge from current powerful monocular depth estimators [28, 29]. Considering point clouds are sparse and not available in many sparse-view cases, monocular depth shows its advantage in density and robustness for our tasks. To tackle the scale ambiguity and error of monocular depths, some previous works and concurrent sparse-view 3DGS methods have introduced various scale-invariant losses [9, 35, 48, 54, 61] including depth ranking loss [41, 49], however, all of which are not optimal for us. Firstly, flexible Gaussians are more sensitive to wrong depth cues, requiring extra designs for regularization. Also, these losses align the depth to a certain fixed global scale, which may ignore minor local depth changes. This overlook can lead to a noisy primitive distribution, particularly in regions with intricate textures. Besides, we notice an HDN loss [55] that can preserve details in monocular depth estimation. Nevertheless, it is also unsuitable as its reliance on multi-scale patches would bring long-distance errors and compromise geometric accuracy.

3. Method

3.1. Preliminary for 3D Gaussian Splatting

Representation. 3D Gaussian splatting [17] represents 3D information with a set of 3D Gaussians. It computes pixel-wise color $\mathcal{C}$ with a set of 3D Gaussian primitives θ, view pose P, and the camera parameter involving the center o.

Specifically, a Gaussian primitive can be described with a center $\mu \in \mathbb{R}^3$, a scaling factor $s \in \mathbb{R}^3$, and a rotation quaternion $q \in \mathbb{R}^4$. The basis function of the i-th primitive $\mathcal{G}_i$ is in the form of:

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

where the covariance matrix Σ can be calculated from the scale s and rotation q. For rendering purposes, the Gaussian primitive also retains an opacity value $\alpha \in \mathbb{R}$ and a K-dimensional color feature $f \in \mathbb{R}^K$. Then $\theta_i = \{\mu_i, s_i, q_i, \alpha_i, f_i\}$ is the parameters for the i-th Gaussian.

Rendering. 3D Gaussian Splatting utilizes a point-based rendering to compute the color $\mathcal{C}$ of pixel x_p by blending N ordered Gaussians overlapping the pixel:

$$\mathcal{C}(x_p) = \sum_{i \in N} c_i \widetilde{\alpha}_i \prod_{j=1}^{i-1} (1 - \widetilde{\alpha}_j), \tag{2}$$

where c_i is the decoded color of feature f.

Different from NeRF's ray sampling strategy, the involved N Gaussians are gathered by a well-optimized rasterizer according to x_p, the camera parameter, the view pose P, and a set of pre-defined roles. And the rendering opacity $\widetilde{\alpha}$ of N primitives are calculated by α and their projected 2D Gaussians $\mathcal{G}^{proj}$ on image plane :

$$\widetilde{\alpha}_i = \alpha_i \mathcal{G}_i^{proj}(x_p). \tag{3}$$

Then, similar to NeRF, we can represent the pixel-wise depth $\mathcal{D}$ with the distance to the camera center o:

$$\mathcal{D}(x_p) = \sum_{i \in N} ||\mu_i - o||_2 \times \widetilde{\alpha}_i \prod_{j=1}^{i-1} (1 - \widetilde{\alpha}_j). \tag{4}$$

Optimzation. 3D Gaussian Splatting optimizes the parameters θ for all Gaussians through gradient descent under color supervision. During the optimization process, it identifies and duplicates the most active primitives to better represent intricate textures, simultaneously removing re-

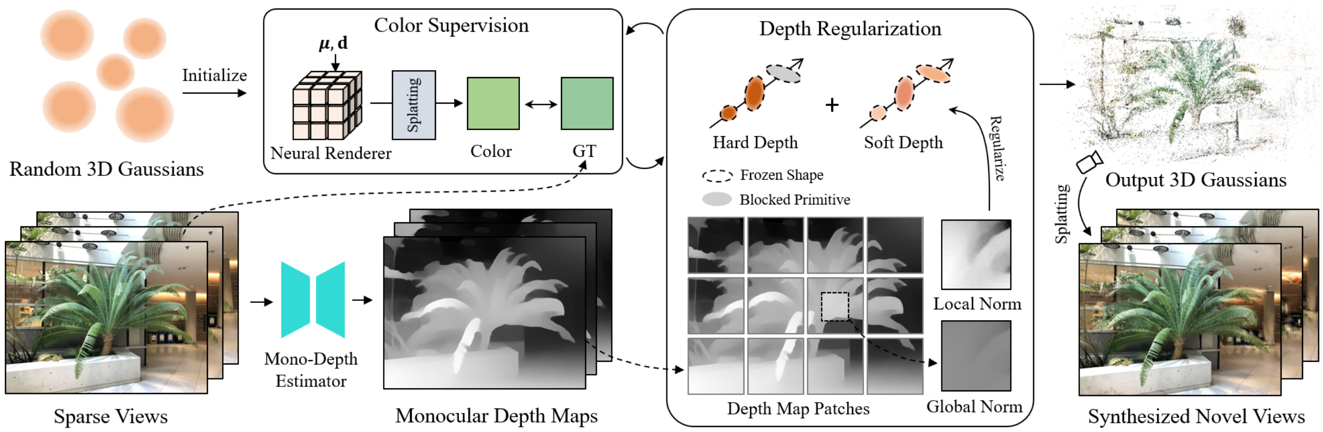

Figure 3. **The Framework of DNGaussian.** Our framework starts from a random initialization and consists of a Color Supervision module and a Depth Regularization module. The optimization process of Color Supervision mainly inherits from 3D Gaussian Splatting [17] except for a Neural Color Renderer. In the depth regularization, we render a Hard Depth and a Soft Depth for the input view, and separately calculate the losses of the pre-generated monocular depth map with the proposed Global-Local Depth Normalization. Finally, the output Gaussian field enables efficient and high-quality novel view synthesis.

dundant primitives. In this work, we inherit these optimization strategies for color supervision.

Initialization. To start from a better geometry, the method suggests utilizing the point cloud from COLMAP [32, 33] or other SfMs to initialize the Gaussians. Instead, considering the instability of point clouds in sparse-view situations, we initialize our method with a random set of Gaussians.

3.2. Depth Regularization for Gaussians

Despite sharing a similar depth computation, existing depth regularization for NeRFs cannot transfer to 3D Gaussian radiance fields due to the huge differences. First, a target conflict between color and depth would occur in the extra parameters. Also, previous regularization for the continuous NeRF only focuses on density, for which it can hardly work well on the discrete and flexible Gaussian primitives.

Shape Freezing. 3D Gaussian radiance fields possess four optimizable parameters $\{\mu, s, q, \alpha\}$ that can directly influence the depth, which is more complex than NeRF. Since the mono-depth is much smoother and easier to fit than color, apply an all-parameter depth regularization on the entire model, which is widely used in previous sparse-view neural fields [9, 13, 41, 49, 54], would lead the shape parameters to overfit the target depth map and cause blurry appearance. Thus, these parameters must be treated differently. Since the scene geometry is mainly represented by the position distribution of Gaussian primitives, we regard the *center* μ and *opacity* α as the most important parameters to regularize, for they separately stand for the position itself and the occupancy of a position. Furthermore, to reduce the negative influence for color reconstruction, we *freeze the scaling* s *and rotation* q in the depth regularization.

Hard Depth Regularization. To achieve the spatial reshaping of the Gaussian fields, we first propose a Hard Depth Regularization that encourages the movement of the nearest Gaussians, which are expected to compose surfaces but often cause noises and artifacts. Considering the predicted depth is rendered with the mixture of multiple Gaussians and reweighted by the cumulative product $\widetilde{\alpha}$, we manually apply a large opacity value τ to all Gaussians. Then, we render a "hard depth" that mainly consists of the nearest Gaussians on the ray shot from camera center o and across the pixel x_p:

$$\mathcal{D}_{hard}(x_p) = \sum_{i \in N} \tau(1-\tau)^{i-1}\mathcal{G}_i^{proj}(x_p)||\mu_i - o||_2. \quad (5)$$

Since now only the center μ is in optimization, Gaussians at wrong positions cannot avoid being regularized by decreasing their opacity or changing shapes, and thus their centers μ move. The regularization is implemented by a similarity loss at the target image area $\mathcal{P}$ to encourage the hard depth $\mathcal{D}_{hard}$ close to the monocular depth $\widetilde{\mathcal{D}}$:

$$\mathcal{R}_{hard}(\mathcal{P}) = \mathcal{L}_{similar}(\mathcal{D}_{hard}(\mathcal{P}), \widetilde{\mathcal{D}}(\mathcal{P})). \quad (6)$$

Soft Depth Regularization. Only regularizing on "hard depth" is insufficient due to the absence of opacity optimization. We also expect to ensure the accuracy of the real rendered "soft depth", otherwise, the surface may become semitransparent and cause hollowness. From this perspective, we additionally *freeze the Gaussian center* μ (denoted by $\breve{\mu}$) to avoid negative influence caused by the center moving, and propose Soft Depth Regularization for the tuning of the opacity α:

$$\begin{gathered}\mathcal{D}_{soft}(x_p) = \sum_{i \in N} ||\breve{\mu}_i - o||_2 \times \widetilde{\alpha}_i \prod_{j=1}^{i-1}(1-\widetilde{\alpha}_j), \\ \mathcal{R}_{soft}(\mathcal{P}) = \mathcal{L}_{similar}(\mathcal{D}_{soft}(\mathcal{P}), \widetilde{\mathcal{D}}(\mathcal{P})).\end{gathered} \quad (7)$$

With both the Hard and Soft Depth Regularization, we constrain the nearest Gaussians to stay in a suitable position with high opacity, therefore composing complete surfaces.

3.3. Global-Local Depth Normalization

Previous depth-supervised neural fields usually build the depth loss on the source scales of the depth maps[9, 13, 35, 41, 54]. This type of alignment measures all losses via a fixed scale based on the statistics of a large area. As a result, it might lead to the overlooking of small errors, particularly when dealing with multiple objectives such as color reconstruction or a wide range of depth variance. This overlook may matter not much in previous NeRF-based works, but can raise heavier problems in the Gaussian radiance fields.

In the Gaussian radiance fields, correcting small depth errors is more challenging because it primarily relies on the movement of Gaussian primitives, a process that happens with a minor learning rate. Also, if the primitives have not been corrected in position during depth regularization, they will become float noises and cause failures, especially in regions with detailed appearance where gathering numerous primitives, as shown in Figure 4.

Local Depth Normalization. To solve this problem, we make the loss function refocus on small errors by introducing a patch-wise local normalization. Specifically, we cut a whole depth map into small patches and normalize the patch $\mathcal{P}$ of predicted depth and monocular depth with the mean value of 0 and standard deviation of near to 1:

$$\mathcal{D}^{LN}(x) = \frac{\mathcal{D}(x) - \text{mean}(\mathcal{D}(\mathcal{P}))}{\text{std}(\mathcal{D}(\mathcal{P})) + \epsilon}, \quad \text{s.t. } x \in \mathcal{P}, \tag{8}$$

where ϵ is a value for numerical stability. Since then, all patches have been normalized on a local scale and the loss can be calculated inside. Later, we apply the Local Depth Normalization to the Hard and Soft Depth Regularization to help with geometry reshaping.

Global Depth Normalization. In contrast to focusing on small local losses, we also need a global view to learn an overall shape. To fill the lack of global scale, we further add a Global Depth Normalization in the depth regularization. This makes the depth loss aware of the global scale while preserving local relevance. Similar to the local one, we apply a patch-wise normalization to free the depth from the source scale and focus on local changes. The only difference is here we use a global standard deviation of the whole image depth $\mathcal{D}_{\mathcal{I}}$ of image $\mathcal{I}$ to replace that of the patch:

$$\mathcal{D}^{GN}(x) = \frac{\mathcal{D}(x) - \text{mean}(\mathcal{D}(\mathcal{P}))}{\text{std}(\mathcal{D}_I)}, \quad \text{s.t. } x \in \mathcal{P}, \mathcal{P} \subseteq \mathcal{I}. \tag{9}$$

In addition, our patch-wise normalization can also avoid long-distance errors in the monocular depth by driving the learning of locally relative depth, which serves a similar effect as depth rank distillation [41, 49]. But differently, for geometry reshaping purposes, we also encourage the model to learn the absolute depth change rather than ignoring it.

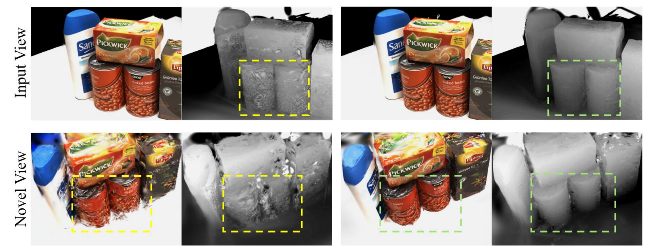

Figure 4. A fixed global scale pays little attention to the small depth errors even under L1 loss, which leads to noisy primitives and causes failures in novel view (yellow box). Our Global-Local Depth Normalization refocuses on small errors via local scale and helps reconstruct a more accurate appearance (green box).

3.4. Training Details

Loss Function The loss function consists of three parts: color reconstruction loss $\mathcal{L}_{color}$, hard depth regularization $\mathcal{R}_{hard}$ and soft depth regularization $\mathcal{R}_{soft}$. Following 3D Gaussian Splatting, the color reconstruction loss is a combination of L1 reconstruction loss and a D-SSIM term of the rendering image $\hat{\mathcal{I}}$ and ground-truth $\mathcal{I}$:

$$\mathcal{L}_{color} = \mathcal{L}_1(\hat{\mathcal{I}}, \mathcal{I}) + \lambda \mathcal{L}_{\text{D-SSIM}}(\hat{\mathcal{I}}, \mathcal{I}). \tag{10}$$

The depth regularization $\mathcal{R}_{hard}$ and $\mathcal{R}_{soft}$ all include a local and a global term separately from our two kinds of depth normalization. We take the L2 loss to measure the similarity. Both of the regularizations are in the form of:

$$\mathcal{R}_T = \mathcal{L}_2(\mathcal{D}_T^{GN}, \widetilde{\mathcal{D}}^{GN}) + \gamma \mathcal{L}_2(\mathcal{D}_T^{LN}, \widetilde{\mathcal{D}}^{LN}), \tag{11}$$

where T stands for $hard$ or $soft$. In practice, we reserve an error tolerance for the L2 loss to relax the constraint. The full loss function is formulated by:

$$\mathcal{L} = \mathcal{L}_{color} + \mathcal{R}_{hard} + \mathcal{R}_{soft}. \tag{12}$$

Neural Color Renderer. 3D Gaussian Splatting stores color via spherical harmonics, however, it is easy to overfit with only sparse views. To relieve this problem, we take a grid encoder and an MLP as the Neural Color Renderer to predict color for each primitive (Figure 3). During inference, we store the intermediate result and only calculate the last MLP layers to merge view direction for acceleration.

4. Experiments

4.1. Setups

Datasets. we conduct our experiment on three datasets: the NeRF Blender Synthetic dataset (Blender) [24], the DTU dataset [16], and the LLFF dataset [23]. We follow the setting used in previous works [26, 41, 51] with the same split of DTU and LLFF to train the model on 3 views and test on another set of images. To erase the noises in the background and focus on the target object, we apply the same

	Setting	LLFF				DTU			
		PSNR ↑	LPIPS ↓	SSIM ↑	AVGE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	AVGE ↓
SRF [7]	Trained on DTU	12.34	0.591	0.250	0.313	15.32	0.304	0.671	0.171
PixelNeRF [53]	Trained on DTU	7.93	0.682	0.272	0.461	16.82	0.270	0.695	0.147
MVSNeRF [4]	Trained on DTU	17.25	0.356	0.557	0.171	18.63	0.197	0.769	0.113
SRF ft [7]	Trained on DTU Fine-tuned per Scene	17.07	0.529	0.436	0.203	15.68	0.281	0.698	0.162
PixelNeRF ft [53]	Trained on DTU Fine-tuned per Scene	16.17	0.512	0.438	0.217	18.95	0.269	0.710	0.125
MVSNeRF ft [4]	Trained on DTU Fine-tuned per Scene	17.88	0.327	0.584	0.157	18.54	0.197	0.769	0.113
Mip-NeRF [2]	Optimized per Scene	14.62	0.495	0.351	0.246	8.68	0.353	0.571	0.323
DietNeRF [15]	Optimized per Scene	14.94	0.496	0.370	0.240	11.85	0.314	0.633	0.243
RegNeRF [26]	Optimized per Scene	19.08	0.336	0.587	0.149	18.89	0.190	0.745	0.112
FreeNeRF [51]	Optimized per Scene	19.63	0.308	0.612	0.134	19.92	0.182	0.787	0.098
SparseNeRF [41]	Optimized per Scene	19.86	0.328	0.624	0.127	19.55	0.201	0.769	0.102
3DGS [17]	Optimized per Scene	15.52	0.405	0.408	0.209	10.99	0.313	0.585	0.252
3DGS†	Optimized per Scene	16.46	0.401	0.440	0.192	14.74	0.249	0.672	0.169
DNGaussian (Ours)	Optimized per Scene	19.12	0.294	0.591	0.132	18.91	0.176	0.790	0.102

† with the same hyperparameters and the neural color renderer as DNGaussian .

Table 1. **Quantitative Comparison on LLFF and DTU for 3 input views.** The best, second-best, and third-best entries are marked in red, orange, and yellow, respectively. Notably, the Gaussian-based methods directly show the background color on the meaningless invisible places, which would cause lower metrics, especially in PSNR.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [24]	14.934	0.687	0.318
NeRF (Simplified) [15]	20.092	0.822	0.179
DietNeRF [15]	23.147	0.866	0.109
DietNeRF + ft [15]	23.591	0.874	0.097
FreeNeRF [51]	24.259	0.883	0.098
SparseNeRF [41]	22.410	0.861	0.119
3DGS [17]	22.226	0.858	0.114
DNGaussian (Ours)	24.305	0.886	0.088

Table 2. **Quantitative Comparison on Blender for 8 input views.** The best, second-best, and third-best entries are marked in red, orange, and yellow, respectively.

object masks as previous works [26] for DTU at evaluation. For Blender, we follow DietNeRF [15] and FreeNeRF [51] to train with the same 8 views and test on 25 unseen images. Aligned with the baselines, downsampling rates of 8, 4, and 2 are applied to LLFF, DTU, and Blender. Following previous sparse-view settings, the camera poses are assumed to be known via calibration or other ways.

Evaluation Metrics. We report PSNR, SSIM [44], and LPIPS [57] scores to evaluate the reconstruction performance quantitatively. Also, an Average Error (AVGE) [26] is reported by the geometric mean of MSE $= 10^{-\text{PSNR}/10}$, $\sqrt{1-\text{SSIM}}$, and LPIPS.

Baselines. Following the previous sparse-view neural fields [15, 26, 41, 51], We take current SOTA methods SRF [7], PixelNeRF [53], MVSNeRF [4], Mip-NeRF [2], DietNeRF [15], RegNeRF [26], FreeNeRF [51], and SparseNeRF [41] as our baselines. For most NeRF-based methods, we directly report their best quantitative results in corresponding published papers for comparisons. The results of raw 3D Gaussian Splatting (3DGS) [17] are also reported.

Implementations. We build our models on the official PyTorch 3D Gaussian Splatting codebase. We train the model with $6,000$ iterations for all datasets, and the soft depth regularization is applied after $1,000$ iterations for stability. We set $\gamma = 0.1, \tau = 0.95$ in loss functions for all experiments. The neural renderer consists of a hash encoder [25] with 16 levels in a resolution range of 16 to 512 and a max size of 2^{19}, and a 5 layer MLP with the hidden dim of 64. We use DPT [28] to predict monocular depth maps for all input views. The models of 3DGS and DNGaussian are randomly initialized with a uniform distribution.

4.2. Comparison

LLFF. The qualitative results and visualizations on the LLFF dataset from 3 input views are reported in Table 1 and Figure 5. Notably, since the NeRF-based baselines would interpolate colors to those invisible areas from input views while the discrete Gaussian radiance fields directly expose the black background on these empty spaces, the 3DGS-based methods natively have a weakness in the reconstruction metrics from these meaningless invisible areas. Despite that, our approach still outperforms all baselines in the LPIPS score, and achieves comparable PSNR, SSIM, and Average Error to the best methods. From both the quantitative and qualitative results, we can see that our DNGaussian predicts more fine details and precise geometry. FreeNeRF tends to synthesize smooth views that lack high-frequency details, also the geometry is not as accurate as the depth-supervised SparseNeRF and Our DNGaussian. Although regularized by same depth maps, SparseNeRF performs more weak in details and geometry completeness. DNGaussian also has huge improvements in both image geometry quality compared to the well-tuned 3DGS.

DTU. The quantitative results on the DTU 3-view setting reported in Table 1 show that our method achieves the best in LPIPS and SSIM, and the second best in Average Error.

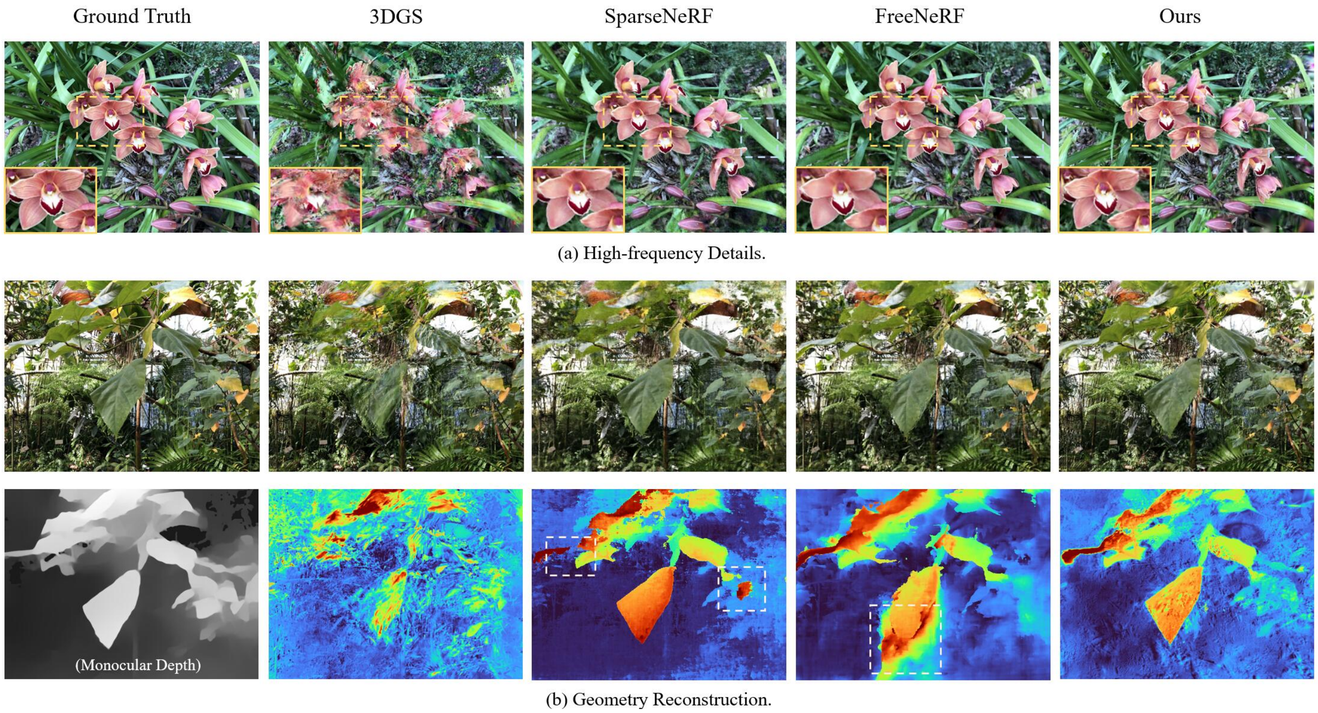

Figure 5. **Qualitative Comparison on LLFF.** 3DGS [17] fails to synthesize accurate novel views under sparse inputs. The rendering views from FreeNeRF [51] and SparseNeRF [41] are both smooth but with too many details lost. FreeNeRF further learns a wrong geometry in complete scenes. Our method learns more complete foreground geometry and renders high-quality novel views with fine details.

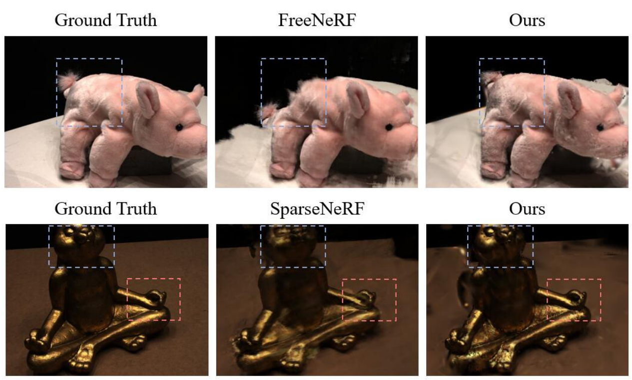

Figure 6. **Qualitative Comparison on DTU.** Our method excels both in geometry and rendering qualities in difficult areas.

However, we got a lower score in PSNR, which is mainly due to scale variance and the noise occlusion coming from the textureless board and background in the scene. In the qualitative examples in Figure 6, It can be observed that our method can learn a more correct and complete geometry compared with both FreeNeRF and the depth-supervised SparseNeRF, and produces high-quality details even on difficult plush and reflective areas.

Blender. To test the fitting ability from surrounding views, we make an evaluation of the Blender dataset under 8 input views. The scores are reported in Table 2, in which some data come from FreeNeRF [51] and DietNeRF [15]. Our method has got the best scores in all PSNR, SSIM and LPIPS. From the qualitative results in Figure 7, it can be seen that our method synthesizes views with correct geometry and fewer floaters compared to the vanilla 3DGS, and has a better performance in detail compared to the second-best FreeNeRF. The results demonstrate that DNGaussian can not only handle looking-forward scenes like LLFF and DTU, but also a whole reconstruction of complex objects with transparent and reflective materials.

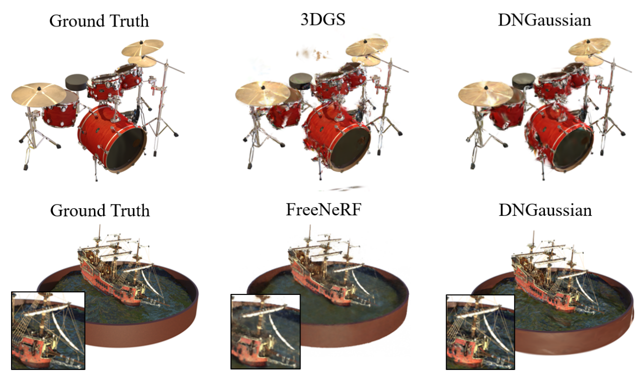

Figure 7. **Qualitative Comparison on Blender.** The results demonstrate the strong fitting ability from surrounding views and reconstruct detailed and complex scenes.

Efficiency. We further conduct an efficiency study on the LLFF 3-view setting with RTX 3090 Ti GPUs to explore the performance of current SOTA baselines [41, 51] with limited GPU memories of 24GB/12GB, and training time

Method	FPS	Time	GPU Mem	PSNR↑	LPIPS↓	SSIM↑
FreeNeRF [51]	9×10^{-2}	2.3 h	4×48 GB	19.63	0.308	0.612
		2.3 h	24 GB	19.71	0.322	0.603
		1.0 h	24 GB	19.66	0.357	0.574
SparseNeRF [41]	9×10^{-2}	1.5 h	32 GB	19.86	0.328	**0.624**
		1.5 h	12 GB	**19.95**	0.334	0.598
		0.5 h	12 GB	19.94	0.341	0.585
Ours	**300**	**3.5 min**	**2 GB**	19.12	**0.294**	0.591

Table 3. **Efficiency Comparison with Limited Resources.** Our method achieves efficient training and the fastest real-time rendering while synthesizing competitive high-quality novel views.

Regularization		Normalization		PSNR ↑	LPIPS ↓	SSIM ↑
AP	Hard Soft	Local	Global			
✓				18.14	0.354	0.538
	✓			17.90	0.351	0.525
✓			✓	18.31	0.339	0.552
✓		✓	✓	18.68	0.331	0.565
	✓		✓	18.55	0.324	0.562
	✓	✓	✓	19.12	0.294	0.591

Table 4. **Ablation Study.** We ablate our method on the LLFF 3-view setting. The results show the effect of our contributions.

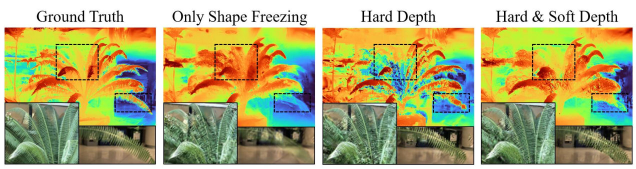

Figure 8. **Visualization Results of Depth Regularization.** Our Hard Depth Regularization significantly improves the high-frequency details but causes hollows. This drawback can be solved via the Soft Depth to synthesize fine details. We take the depth from dense views as the ground truth for comparison.

of 1.0h/0.5h, as shown in Table 3. The top row of each group represents the default setting of the corresponding baseline, where the training time is measured by us with the same number of iterations on a single GPU. While both FreeNeRF and SparseNeRF perform worse under strict resource limitations, our method shows huge advantages in efficiency, which achieves remarkable accelerations of 25× on training time and over 3000× on FPS, while synthesizing competitive quality novel views. Given the necessity for per-scene optimization and rapid visualization, our high efficiency holds significant value for practical applications.

4.3. Ablation Study

We ablate our method on the LLFF 3-view setting. The quantitative results are reported in Table 4 and 5.

Depth Regularization. We ablate our Hard and Soft Depth Regularization with a plain all-parameter (AP) L2 reconstruction loss term. To better separately illustrate the effect of our two types of depth and exclude the influence of shape freezing, we further visualize a comparison to the situation only with shape freezing in Figure 8. It has been shown that the plain depth regularization can not effectively reshape the scene geometry, which proves the necessity of our method. Both the qualitative and quantitative results demonstrate our

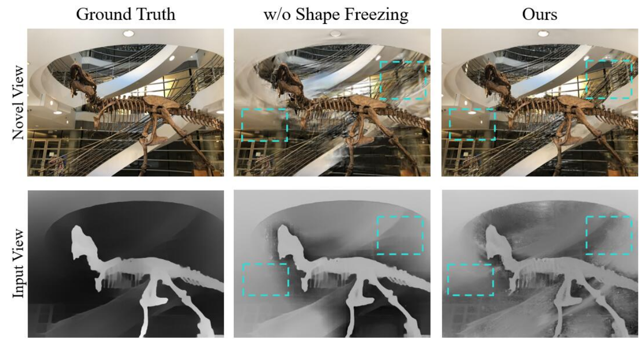

Figure 9. **Visualization Results of Shape Freezing.** The synthesized novel view without shape freezing is filled with blurry areas, which is mainly caused by the over-smooth geometry learned from the monocular depth (left bottom).

Setting	PSNR ↑	LPIPS ↓	SSIM ↑	AVGE ↓
w/o Shape Freezing	17.96	0.363	0.547	0.160
w/o Center Freezing	18.87	0.300	0.584	0.140
All	**19.12**	**0.294**	**0.591**	**0.132**

Table 5. **Ablation Study on Parameter Freezing.** The results demonstrate the necessity of our parameter freezing strategy.

effect on geometry quality and high-frequency details.

Global-Local Depth Normalization. From the result, we can observe that only adding a global normalization can also help fitting, which is mainly due to the local patch-wise loss computation. After the attendance of local normalization, the rendering quality improves especially in detail. These improvements are much more obvious when applied to our proposed regularization than the all-parameter regularization that is unsuitable for the fields. The results correspond to our design and show the effectiveness of our Global-Local Depth Normalization.

Parameter Freezing. To illustrate the effect of our parameter-freezing strategy, we also ablate the shape freezing in regularization and center freezing in soft depth calculation. The results are shown in Table 5 and Figure 9. The visualization illustrates the problem of the depth-color conflict in Sec.3.2. In the situation without center freezing, some primitives may move to unexpected places to compensate for the depth loss, which causes lower quality. By introducing the proposed parameter freezing, we successfully relieve the problems and achieve the best results.

5. Conclusion

This paper presents the DNGaussian framework that introduces 3DGS into the few-shot novel view synthesis task by depth regularization. Due to the space limitation, we have put more discussions in the supplementary material.

Acknowledgements. In this work, we are supported by the National Natural Science Foundation of China 62276016, 62372029. Lin Gu is supported by JST Moonshot R&D Grant Number JPMJMS2011 Japan.

References

[1] 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. 3

[2] 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. 3, 6

[3] 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. 3

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2, 3, 6

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 3

[6] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4182–4194, 2023. 3

[7] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 6

[8] Wenyan Cong, Hanxue Liang, Peihao Wang, Zhiwen Fan, Tianlong Chen, Mukund Varma, Yi Wang, and Zhangyang Wang. Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3193–3204, 2023. 3

[9] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20637–20647, 2023. 3, 4, 5

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3

[11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3

[12] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2

[13] Shoukang Hu, Kaichen Zhou, Kaiyu Li, Longhui Yu, Lanqing Hong, Tianyang Hu, Zhenguo Li, Gim Hee Lee, and Ziwei Liu. Consistentnerf: Enhancing neural radiance fields with 3d consistency for sparse view synthesis. *arXiv preprint arXiv:2305.11031*, 2023. 3, 4, 5

[14] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19774–19783, 2023. 3

[15] 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*, pages 5885–5894, 2021. 2, 3, 6, 7

[16] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 5

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2, 3, 4, 6, 7

[18] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022. 3

[19] Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[20] 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. 3

[21] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 3

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

[23] 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. 5

[24] 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. 3, 5, 6

[25] 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. 2, 3, 6

[26] 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, 3, 5, 6

[27] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 3

[28] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 3, 6

[29] 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*, 44(3), 2022. 3

[30] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 3

[31] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 3

[32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[33] 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. 4

[34] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22883–22893, 2023. 2

[35] Jiuhn Song, Seonghoon Park, Honggyu An, Seokju Cho, Min-Seop Kwak, Sungjin Cho, and Seungryong Kim. Därf: Boosting radiance fields from sparse inputs with monocular depth adaptation, 2023. 2, 3, 5

[36] 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, 3

[37] Jiakai Sun, Zhanjie Zhang, Jiafu Chen, Guangyuan Li, Boyan Ji, Lei Zhao, and Wei Xing. Vgos: Voxel grid optimization for view synthesis from sparse inputs. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1414–1422. International Joint Conferences on Artificial Intelligence Organization, 2023. Main Track. 2, 3

[38] 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. 3

[39] Mikaela Angelina Uy, Ricardo Martin-Brualla, Leonidas Guibas, and Ke Li. Scade: Nerfs from space carving with ambiguity-aware depth estimates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16518–16527, 2023. 3

[40] Chen Wang, Xiang Wang, Jiawei Zhang, Liang Zhang, Xiao Bai, Xin Ning, Jun Zhou, and Edwin Hancock. Uncertainty estimation for stereo matching based on evidential deep learning. *Pattern Recognition*, 124:108498, 2022. 3

[41] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9065–9076, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[42] Xiang Wang, Chen Wang, Bing Liu, Xiaoqing Zhou, Liang Zhang, Jin Zheng, and Xiao Bai. Multi-view stereo in the deep learning era: A comprehensive review. *Displays*, 70: 102102, 2021. 3

[43] Xiang Wang, Haonan Luo, Zihang Wang, Jin Zheng, and Xiao Bai. Robust training for multi-view stereo networks with noisy labels. *Displays*, 81:102604, 2024. 3

[44] 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

[45] Zihang Wang, Haonan Luo, Xiang Wang, Jin Zheng, Xin Ning, and Xiao Bai. A contrastive learning based unsupervised multi-view stereo with multi-stage self-training strategy. *Displays*, page 102672, 2024. 3

[46] 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. 3

[47] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4180–4189, 2023. 2, 3

[48] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Realtime 360° sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*, 2023. 3

[49] 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 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4479–4489, 2023. 3, 4, 5

[50] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 3

[51] 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. 1, 2, 3, 5, 6, 7, 8

[52] 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. 3

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2, 3, 6

[54] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 3, 4, 5

[55] Chi Zhang, Wei Yin, Billzb Wang, Gang Yu, Bin Fu, and Chunhua Shen. Hierarchical normalization for robust monocular depth estimation. *Advances in Neural Information Processing Systems*, 35:14128–14139, 2022. 3

[56] Jiawei Zhang, Xiang Wang, Xiao Bai, Chen Wang, Lei Huang, Yimin Chen, Lin Gu, Jun Zhou, Tatsuya Harada, and Edwin R Hancock. Revisiting domain generalized stereo matching networks from a feature consistency perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13001–13011, 2022. 3

[57] 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

[58] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 286–301. Springer, 2016. 3

[59] Xiaoqing Zhou, Xiang Wang, Jin Zheng, and Xiao Bai. Adaptive spatial sparsification for efficient multi-view stereo matching. *Acta Electronica Sinica*, 51(11):3079–3091, 2023. 3

[60] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023. 3

[61] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00451*, 2023. 3