

MCGS: Multiview Consistency Enhancement for Sparse-View 3D Gaussian Radiance Fields

Yuru Xiao, Deming Zhai, *Member, IEEE*, Wenbo Zhao, *Member, IEEE*, Kui Jiang, *Member, IEEE*, Junjun Jiang, *Senior Member, IEEE*, Xianming Liu, *Member, IEEE*

Abstract—Radiance fields represented by 3D Gaussians excel at synthesizing novel views, offering both high training efficiency and fast rendering. However, with sparse input views, the lack of multi-view consistency constraints results in poorly initialized point clouds and unreliable heuristics for optimization and densification, leading to suboptimal performance. Existing methods often incorporate depth priors from dense estimation networks but overlook the inherent multi-view consistency in input images. Additionally, they rely on multi-view stereo (MVS)-based initialization, which limits the efficiency of scene representation. To overcome these challenges, we propose a view synthesis framework based on 3D Gaussian Splatting, named MCGS, enabling photorealistic scene reconstruction from sparse input views. The key innovations of MCGS in enhancing multi-view consistency are as follows: i) We introduce an initialization method by leveraging a sparse matcher combined with a random filling strategy, yielding a compact yet sufficient set of initial points. This approach enhances the initial geometry prior, promoting efficient scene representation. ii) We develop a multi-view consistency-guided progressive pruning strategy to refine the Gaussian field by strengthening consistency and eliminating low-contribution Gaussians. These modular, plug-and-play strategies enhance robustness to sparse input views, accelerate rendering, and reduce memory consumption, making MCGS a practical and efficient framework for 3D Gaussian Splatting.

Index Terms—3DGS, Novel View Synthesis, Sparse-View.

I. INTRODUCTION

NOVEL view synthesis (NVS) plays a pivotal role in a wide range of real-world applications, such as VR/AR, robotics, and autonomous driving. Neural Radiance Fields (NeRFs) [1] achieve realistic novel view synthesis from a set of dense input views. However, their performance significantly drops when only sparse views are available, a challenge known as the few-shot neural rendering problem [2]–[4]. Recent studies have made significant strides in addressing this issue, achieving photorealistic novel view synthesis through various techniques such as depth regularization [5], depth distillation [6], semantic consistency regularization [7], and frequency annealing [3]. Nonetheless, these methods still face challenges, including costly training and slow rendering speeds due to the limitations of implicit representation. While recent grid-based methods [8] have substantially accelerated training, rendering speeds remain a bottleneck.

3D Gaussian Splatting (3DGS) [9] has employed 3D Gaussian primitives to model the radiance field, achieving

Y. Xiao, D. Zhai, W. Zhao, K. Jiang, J. Jiang, and X. Liu are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, E-mail: xiaoyuru.30@gmail.com; {zhaidebing, wbzhao, jiangkui, jiangjunjun, csxm}@hit.edu.cn.

success in high-fidelity and real-time novel view synthesis with high training efficiency. However, it still suffers from a significant performance decline when only sparse input views are available [10]–[12]. This drop is primarily due to its reliance on multi-view constraint based optimization and densification, which are less effective under sparse view conditions. The weakened multi-view consistency constraints result in unstable training, especially when the initialized point cloud is extremely sparse. To remedy this limitation, existing methods integrate explicit geometry priors to enhance multi-view consistency. These strategies focus specifically on densely building scenes with strong geometry priors, such as depth distillation [11] or depth regularization [10] from pre-trained dense estimation networks. However, their performance heavily depends on dense initialization provided by a multi-view stereo (MVS) algorithm, resulting in inefficient scene representation and high memory costs. More recently, CoR-GS [12] introduced a co-pruning strategy to address inconsistencies in Gaussian representations that cannot be resolved through standard optimization methods. This approach establishes paired Gaussian fields to identify Gaussians that are inaccurately positioned. However, training two parallel fields significantly doubles the training time. Therefore, developing an efficient method to identify inconsistent Gaussians and improve overall multi-view consistency remains a significant challenge.

In this paper, we propose novel strategies for initialization and optimization to improve multi-view consistency, ensuring high rendering speed, short training duration, and low memory consumption. First, we develop a sparse initializer that incorporates a sparse matcher and a random filling strategy to achieve efficient yet sufficient scene representation. Additionally, we introduce a multi-view consistency guided progressive pruning strategy. During the initial training stage, we compute a prune mask using high-level general visual features and gradually integrate low-level features in subsequent iterations. Our progressive pruning strategy targets Gaussians where the fine-grained color consistency constraint is weak and cannot be optimized or pruned effectively. This approach shifts the training focus toward the geometric entities while mitigating perturbations caused by randomly filled Gaussians in empty space, thereby reducing “floater” artifacts common in sparse-view scenarios. Together, the sparse initializer and pruning strategy enhance geometric consistency, contributing to both memory and rendering efficiency. Extensive experiments with the LLFF, Blender, and DTU datasets reveal that our method significantly improves performance over the vanilla 3DGS

architecture in sparse-view scenarios, with fewer Gaussian primitives and higher rendering speed, making it a practical pipeline for sparse-view 3D Gaussian radiance fields.

Our main contributions are summarized as follows:

- We propose a novel sparse initializer that produces sparse yet sufficient initial points, which efficiently embeds geometric prior in the initialization process.
- We propose a multi-view consistency-guided progressive pruning strategy to shift the training focus to the geometric entity, leading to enhanced overall consistency and representation efficiency of the optimized field.
- The proposed MCGS serves as a plug-and-play module for 3DGS-based architectures that enhances their robustness to view sparsity, while also improving memory and rendering efficiency.

II. RALATED WORK

Radiance Fields. Neural Radiance Fields (NeRFs) [1] have emerged as outstanding 3D representation methods for novel view synthesis. NeRF employs a large Multi-Layer Perceptron (MLP) network to map 3D coordinates to object attributes, such as radiance and color. Coupled with volume rendering, it produces remarkably photo-realistic novel views. Recent research has focused on improving rendering quality [13], [14], enhancing efficiency [15], and increasing robustness to sparse views [3]. However, NeRF's ray-marching process requires querying the MLP network for numerous sampling points, limiting its rendering speed. A recent alternative, 3D Gaussian Splatting (3DGS) [9], represents radiance fields using explicit Gaussian primitives and renders scenes through differentiable rasterization. This approach offers high-quality reconstruction, efficient training, and real-time rendering, making it well-suited for practical applications. Despite these advantages, 3DGS relies on dense input views. Sparse input views pose challenges, as they result in insufficient Gaussian initialization from structure-from-motion (SFM) and reduce multi-view consistency, leading to geometry distortion and degraded novel view quality. Improving 3DGS performance under sparse-view conditions remains an open research problem.

Novel View Synthesis Using Sparse Views. Many methods have been proposed for NeRF to achieve realistic novel view synthesis from sparse input views. Some methods integrate additional 3D supervision, such as sparse point clouds [5] or estimated depth maps [6], [16], [17], to regularize the geometry. In addition to utilizing extra 3D information, some approaches apply random patch-based semantic consistency constraints [7], [18] or geometry regularization [2]. Depth distillation [6] is a skillful technique for extracting depth priors from a depth estimation network, though it suffers from smoothness inherited from imprecisely estimated depth. GeCoNeRF [19] introduces multi-view consistency regularization for few-shot novel view synthesis. Generative methods [20]–[25] learn geometric priors from large multi-view datasets and require additional fine-tuning. Another set of methods [26], [27] performs geometry regularization on estimated correspondences extracted from dense matchers. However, the training efficiency and rendering speed of these methods are still limited

by their slow backbones. Although recent methods have shifted focus to designing strategies for grid-based backbones [8], [28], they still face challenges with low rendering speed and a trade-off between training efficiency and rendering quality.

Recent methods [10], [11] have been developed to improve the performance of 3D Gaussian Splatting (3DGS) while ensuring efficient training and inference. These approaches employ techniques such as depth distillation and depth regularization to refine the geometry of a densely initialized Gaussian field generated from multi-view stereo (MVS), achieving significant results in sparse-view settings. CoR-GS [12] introduces a co-pruning strategy to remove inconsistent Gaussians and a co-regularization method to correct geometry, establishing a novel pruning and refinement paradigm for sparse-view 3DGS.

III. METHOD

A. Preliminaries: 3D Gaussian Splatting

Representation and Rendering. 3D Gaussian Splatting (3DGS) [29] represents a 3D scene as a set of anisotropic Gaussian primitives. Each primitive is defined by differentiable parameters: a position vector $\mu \in \mathbb{R}^3$, a rotation quaternion $q \in \mathbb{R}^4$, and a scaling vector $s \in \mathbb{R}^3$. These parameters collectively define the primitive's distribution in 3D space:

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

where the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$ can be computed from the scale vector s and the rotation quaternion q . These Gaussians are then rendered with their differentiable attributes, such as opacity $\alpha \in \mathbb{R}$ and color, which is represented by spherical harmonic coefficients.

3DGS utilizes the volume rendering function to render the Gaussians in order of their depth as

$$\mathbf{c}(\mathbf{p}) = \sum_{i \in N} \mathbf{c}_i \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j), \quad (2)$$

where \mathbf{p} is the overlapped pixel, \mathbf{c}_i is the color of the i -th Gaussian $G_i(\mathbf{x})$ computed from spherical harmonic coefficients, and $\hat{\alpha}_i$ can be evaluated from the projected 2D Gaussian $G_i^{2D}(\mathbf{p})$:

$$\hat{\alpha}_i = \alpha_i G_i^{2D}(\mathbf{p}). \quad (3)$$

The pixel-wise depth d can also be computed using the volume rendering function:

$$d(\mathbf{p}) = \sum_{i \in N} d_i \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j), \quad (4)$$

where $d_i = |\boldsymbol{\mu}_i - \mathbf{o}|_2$ is the depth of the i -th Gaussian to the camera center \mathbf{o} .

B. Sparse Initializer

Current 3DGS techniques typically use point clouds derived from COLMAP for initializing 3D Gaussian representations. However, the point clouds generated by COLMAP are often sparse, leading to suboptimal performance, especially in scenarios with limited input views. To address this, some

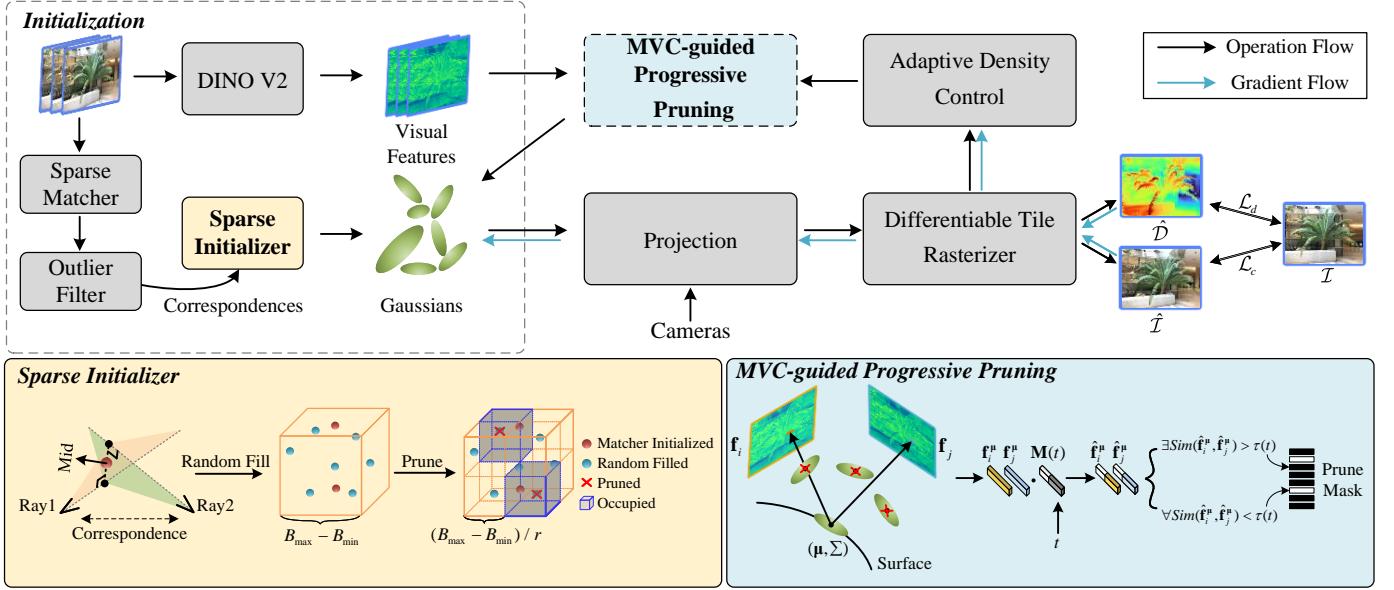


Fig. 1. **An overview of the complete framework.** Our proposed pipeline includes a sparse initializer to provide a sparse yet sufficient initial point cloud and a multi-view consistency-guided progressive pruning strategy to exclude noisy points from the initializer and enhance the overall multi-view consistency during training.

approaches incorporate multi-view stereo (MVS) methods to produce a denser point cloud during initialization. Despite this, the initial Gaussian primitives tend to exhibit high-frequency characteristics, introducing high-frequency artifacts [3] and resulting in inefficient scene representation.

In contrast, we utilize the end-to-end sparse matching network LightGlue [30], pre-trained on large-scale multi-view datasets, to obtain initial correspondences $\langle \mathbf{r}_s, \mathbf{r}_t \rangle_N$ for each image \mathcal{I}_s and its paired view \mathcal{I}_t , where $\mathbf{r}_i = (\mathbf{o}_i, \mathbf{d}_i)$, $i \in s, t$, represents the ray with origin \mathbf{o}_i and direction \mathbf{d}_i . We compute the midpoint \mathbf{p}_{mid} and its color \mathbf{c}_{mid} for each correspondence pair $\langle \mathbf{r}_s, \mathbf{r}_t \rangle$ as: $\mathbf{p}_{mid} = \frac{\mathbf{x}_s + \mathbf{x}_t}{2}$ and $\mathbf{c}_{mid} = \frac{\mathbf{c}_s + \mathbf{c}_t}{2}$, where \mathbf{x}_s and \mathbf{x}_t are the nearest points on the two rays, and \mathbf{c}_s and \mathbf{c}_t are their respective colors. This results in an initialized point cloud $\langle \mathbf{p}_{mid}, \mathbf{c}_{mid} \rangle_N$. Optionally, for datasets with significant symmetry that may cause visual ambiguity, we apply an outlier filter following CorresNeRF [26].

Random Filling. The sparse feature matching process encounters substantial challenges in low-texture areas or regions with repetitive structures [30], [31], often resulting in insufficiently initialized Gaussians. To mitigate this issue, we propose a random filling strategy. Initially, we randomly initialize N points within the bounding box defined by (B_{min}, B_{max}) . To prevent interference with points initialized by the sparse matching process, we exclude any randomly generated points that fall within voxels already occupied by the points derived from the sparse matcher. The voxel size is calculated as $\frac{B_{max} - B_{min}}{r}$, where r represents the predefined resolution. This random filling approach enhances the overall completeness of the initialized geometry while minimizing disruption to points containing multi-view consistency priors. However, some randomly filled points may reside in empty spaces with minimal contribution to geometric reconstruction, which can result in suboptimal artifacts. These extraneous points are subsequently

addressed through a newly designed pruning strategy.

C. Multi-view Consistency Guided Progressive Pruning

Current 3DGS methods often utilize multi-view consistency constraints based on color properties to optimize and densify Gaussians. However, such fine-grained constraints struggle with inconsistent Gaussians, particularly in low-texture regions where spatial similarity in color leads to ambiguity, especially when input views are sparse. Recently, general features extracted from DINOv2 [32], embedded with global context, have demonstrated robust performance in identifying slightly inconsistent structures. Despite this, integrating these visual features effectively into the 3DGS architecture to handle inconsistent Gaussians remains a challenge. Additionally, the initial random points filled in empty space contribute minimally to reconstruction and must be addressed.

To overcome these challenges, we propose a multi-view consistency-guided (MVC-guided) progressive pruning strategy, utilizing the powerful DINOv2 visual encoder to exclude inconsistent Gaussians. We begin by extracting visual features $\{\mathbf{f}_i, i = 1, \dots, N\}$ from N input images $\{\mathcal{I}_i, i = 1, \dots, N\}$. Each feature $\mathbf{f}_i \in \mathbb{R}^{K \times H \times W}$ contains K -dimensional representations, concatenated from components of dimensions $\{K_l, l = 1, \dots, L\}$ extracted from various levels l of DINOv2. We then upsample these features to align them with pixel scale. For Gaussians with central positions $\{\mu_j, j = 1, \dots, M\}$, we project the 3D points onto the input images using intrinsic and extrinsic parameters \mathbf{K}_i and $(\mathbf{R}_i, \mathbf{T}_i)$:

$$\mu_j^{2D} = \frac{1}{z} \mathbf{K}_i(\mathbf{R}_i \mu_j + \mathbf{T}_i), \quad (5)$$

where z is the distance to the camera origin along the z axis in the camera coordinate system. We query the feature $\mathbf{f}_i^{\mu_j} \in \mathbb{R}^K$ at position μ_j^{2D} from the feature map \mathbf{f}_i using

bilinear interpolation. For each Gaussian, we compute the cosine similarity s between features $\langle \mathbf{f}_m^{\mu_j}, \mathbf{f}_n^{\mu_j} \rangle$ queried from pairs of images, with $m \neq n$, as $s_{mn}^{\mu_j} = \frac{\mathbf{f}_m^{\mu_j} \cdot \mathbf{f}_n^{\mu_j}}{\|\mathbf{f}_m^{\mu_j}\|_2 \|\mathbf{f}_n^{\mu_j}\|_2}$. We apply the mask \mathbf{M} to prune Gaussians as

$$M_j = \begin{cases} 1, & \text{if } s_{mn}^{\mu_j} < \tau, \forall m, n \in [N], m \neq n \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

where $[N]$ represents the set of integers from 1 to the number of input views N , and M_j is the j -th element in \mathbf{M} and the j -th Gaussian. Directly applying fixed levels of features and a threshold τ leads to insufficient Gaussians during the initial training phase. As a result, we relax the constraint and design a progressive pruning strategy by applying masks to the features to exclude low-level features and gradually incorporate these low-level features during training. Additionally, we increase the threshold τ from a small initial value. The mask represented by Eq. 6 can be adapted as

$$M_j = \begin{cases} 1, & \text{if } s_{mn}^{\mu_j}(t) < \tau(t), \forall m, n \in [N], m \neq n \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

We apply the mask $\hat{\mathbf{M}}(t)$ to the features to compute similarity with a progressively increased pruning step $t \in [T]$:

$$s_{mn}^{\mu_j}(t) = \frac{(\hat{\mathbf{M}}(t)\mathbf{f}_m^{\mu_j}) \cdot (\hat{\mathbf{M}}(t)\mathbf{f}_n^{\mu_j})}{\|\hat{\mathbf{M}}(t)\mathbf{f}_m^{\mu_j}\|_2 \|\hat{\mathbf{M}}(t)\mathbf{f}_n^{\mu_j}\|_2} \quad (8)$$

with

$$\hat{M}_k(t) = \begin{cases} 1, & k > \sum_{l=1}^{L-t} K_l, t < L \\ 1, & t \geq L \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where $\hat{M}_k(t)$ is the k -th element in the mask $\hat{\mathbf{M}}(t)$, and $\tau(t)$ is set as a hyperparameter. The progressive pruning strategy excludes Gaussians in empty space that have low contributions to the rendered results, thereby focusing training and densification more on Gaussians on the surface.

D. Edge-Aware Depth Regularization

The proposed method improves the multi-view consistency of Gaussian representations by utilizing general visual features. However, the newly introduced pruning strategy encounters challenges with geometric voids (see Fig. 2). These voids arise from either an insufficient number of initial Gaussians or pruning errors based on opacity, especially in low-texture regions. To address these issues, we build on the techniques presented in [33], [34], incorporating edge-aware depth regularization (EADR) to correct geometric distortions. After the final MVC-guided progressive pruning step T , we apply EADR to enhance depth continuity and minimize artifacts along object boundaries.

$$\mathcal{L}_d = \frac{1}{N} \omega_d \sum_{i,j} \left(|\partial_x D_{i,j}| e^{-\beta |\partial_x \mathcal{I}_{i,j}|} + |\partial_y D_{i,j}| e^{-\beta |\partial_y \mathcal{I}_{i,j}|} \right), \quad (10)$$

where $D_{i,j}$ represents the (i, j) -th pixel of the depth map \mathbf{D} , and $\mathcal{I}_{i,j}$ denotes the corresponding pixel in the input image. The parameter N is the total number of pixels, and β —fixed

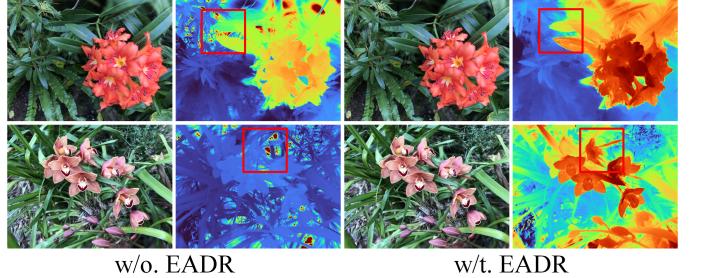


Fig. 2. We present the RGB and depth results generated by our method, comparing scenarios with and without edge-aware depth regularization (EADR). Geometric voids that cannot be resolved by our pruning strategy are effectively addressed using the EADR approach.

at 2 throughout all experiments—acts as a hyperparameter controlling edge sharpness in the depth map. During the initial training phase, the weight ω_d is set to 0. After the final pruning step, it is increased to 1.

$$\omega_d = \begin{cases} 0, & i < (T-1)i_{step} \\ 1, & \text{otherwise} \end{cases}, \quad (11)$$

where i is the current iteration count, T denotes the total number of pruning steps, and i_{step} specifies the number of iterations between consecutive pruning steps. The refinement process aims to minimize geometry errors in low-texture regions while smoothing the previously established geometry.

IV. EXPERIMENTS

A. Setups

Datasets and Metrics. We evaluate our method using three datasets: the synthetic Blender dataset [1], the forward-facing LLFF dataset [35], and the DTU dataset [36]. For the Blender dataset, we follow the protocols of FreeNeRF [3] and Diet-NeRF [7], training our model on 8 specific views (IDs 26, 86, 2, 55, 75, 93, 16, and 73), and perform evaluations on 25 images uniformly selected from the test set, downsampled by a factor of 2. For the LLFF dataset, we adhere to FreeNeRF's procedure, training our model on the first 3 remaining views and evaluating every eighth image, with all images downsampled by a factor of 8. For the DTU dataset, we follow FreeNeRF's experimental setup, conducting experiments with 3 input views and an 8× downsampling rate. Evaluations of the DTU dataset are performed on images masked with ground truth object masks. We report PSNR, SSIM, and LPIPS metrics, averaging the results across all scenes in accordance with community standards.

Implementation Details. We implement MCGS using the 3DGS base architecture. The initialized point cloud is obtained through our sparse initializer, which relies solely on the training views. We utilize calibrated ground truth poses in the datasets, adhering to community standards. We set the total number of iterations to 10,000, with the position learning rate decreasing from 0.0016 to 0.000016. The densification interval and the opacity reset interval are set to 300 and

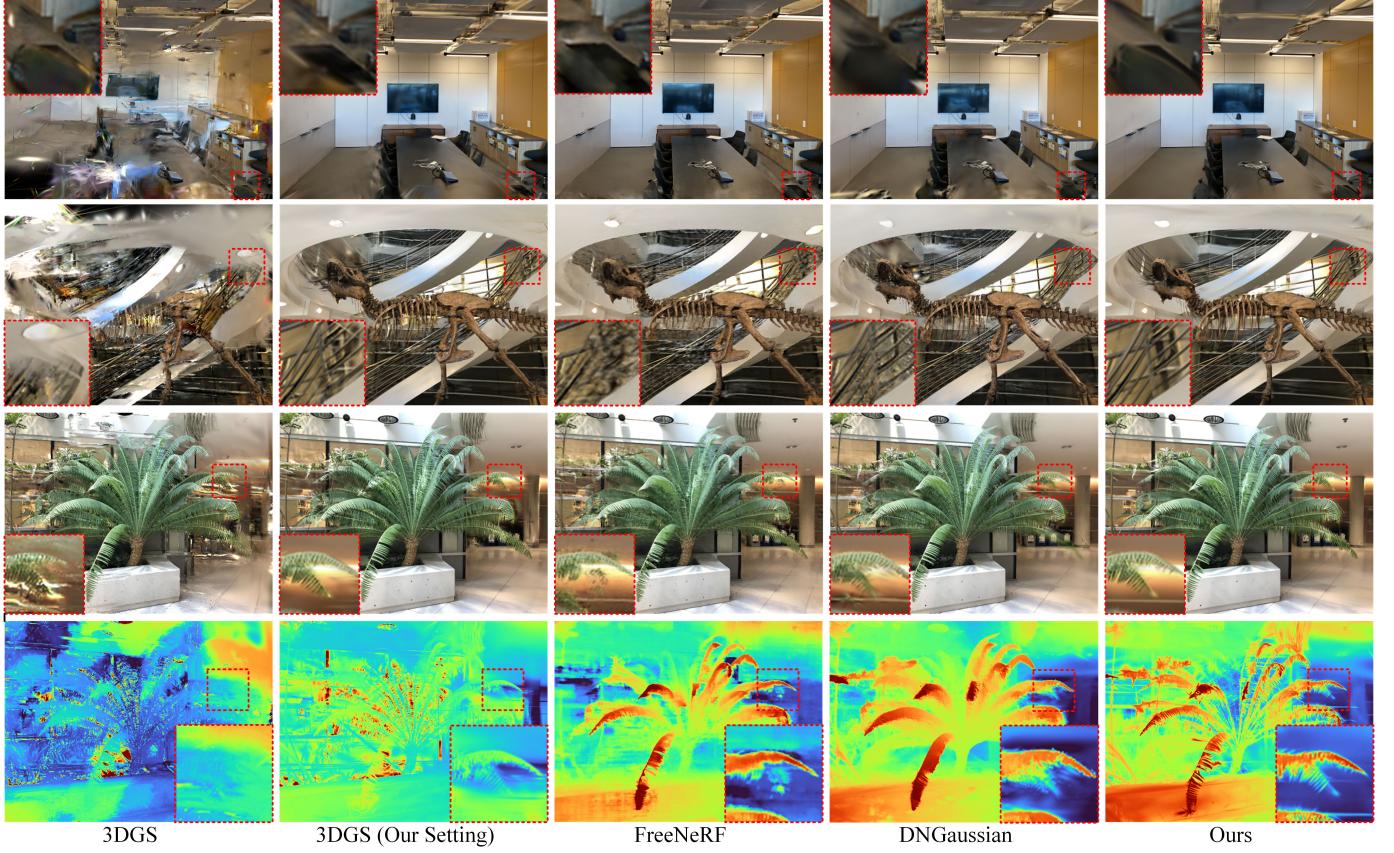


Fig. 3. **Qualitative Comparisons on LLFF.** We present novel views of the scenes “room”, “trex”, and “fern” (first, second, and third rows) rendered by our method as well as the baselines: 3DGS, 3DGS with our setting, FreeNeRF, and DNGaussian. Our method achieves the most realistic novel views, especially in the areas highlighted by red boxes. Please enlarge the PDF to see more details.

1,000, respectively. The density gradient threshold is fixed at 0.0005. For our proposed method, we configure the total pruning steps T to 3 or 4, with the pruning step increasing by 1 every 3,000 iterations for the LLFF or Blender dataset, respectively. For feature similarity computation, we extract features from DINO-ResNet-50 [32] with feature dimensions $K_l \in \{64, 64, 128, 256\}$. The pruning threshold $\tau(t)$ is represented by the lists $[0.75, 0.8, 0.85, 0.85]$ and $[0.6, 0.65, 0.7, 0.8]$ for the LLFF and Blender datasets, respectively. For the Blender dataset, we apply an outlier filter to remove inconsistent correspondences according to CorresNeRF and use background occlusion regularization following FreeNeRF. All experiments are conducted on a single NVIDIA 3090 GPU.

B. Evaluation on LLFF Dataset

We present comprehensive qualitative and quantitative evaluations on the LLFF dataset, as shown in Fig. 3 and Tab. I. For qualitative results, our method demonstrates significant improvement compared to both 3DGS and 3DGS optimized with our setting. The results rendered by the base 3DGS architecture exhibit significant distortion, whereas our method achieves highly realistic rendering with detailed RGB images. Notably, the well-known baseline FreeNeRF fails to capture the detailed areas highlighted by the red boxes due to the exclusion of high-frequency information through frequency regularization. The recent few-shot NVS baseline, DNGaussian,

which is based on 3DGS, produces blurred RGB images due to coarse-grained depth regularization using a monocular depth estimation network. In contrast, our method neither excludes high-frequency information nor relies on a pre-trained depth estimation network, achieving more detailed rendering.

TABLE I
QUANTITATIVE COMPARISONS ON LLFF. \dagger INDICATES THAT THE METHOD IS IMPLEMENTED BASED ON OUR SETTING. THE BEST AND SECOND-BEST QUANTITATIVE RESULTS ARE MARKED IN RED AND GREY, RESPECTIVELY.

Method	Venue	LLFF		
		PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
DSNeRF	CVPR-22	18.94	0.582	0.362
RegNeRF	CVPR-22	19.08	0.587	0.336
FreeNeRF	CVPR-23	19.63	0.612	0.308
SparseNeRF	ICCV-23	19.86	0.624	0.328
SPARF	CVPR-23	20.20	0.630	0.327
3DGS	TOG-23	14.94	0.468	0.379
3DGS †	TOG-23	19.09	0.645	0.242
MipSplatting	CVPR-24	15.27	0.479	0.371
MipSplatting †	CVPR-24	19.17	0.646	0.247
DNGaussian	CVPR-24	20.02	0.682	0.229
Ours w/3DGS	–	20.33	0.699	0.226
Ours w/MipSplatting	–	20.37	0.696	0.234

The quantitative results further verify the effectiveness of our method. Our method surpasses the base 3DGS by more than 5 dB in terms of PSNR with its original setting, and by 1.2 dB with our setting. Both qualitative and quantitative

results indicate that our method significantly enhances the multi-view consistency of 3D Gaussians. We also integrate our method into the recent MipSplatting architecture [37] to verify its effectiveness across different architectures. Our method achieves a 1.2 dB PSNR improvement compared to MipSplatting with our setting.

C. Evaluation on Blender Dataset

Fig. 4 and Tab. II show the qualitative and quantitative results, respectively. For qualitative results, we compare our method with the base 3DGS, the well-known baseline FreeNeRF, and DNGaussian. Our method shows comparable results to FreeNeRF and DNGaussian, while performing better in areas highlighted by red boxes. For quantitative results, our method outperforms the base architecture 3DGS by 1.3 dB and DNGaussian by 0.1 dB in terms of PSNR. These qualitative and quantitative results further verify the effectiveness of our method on the panoramic dataset.

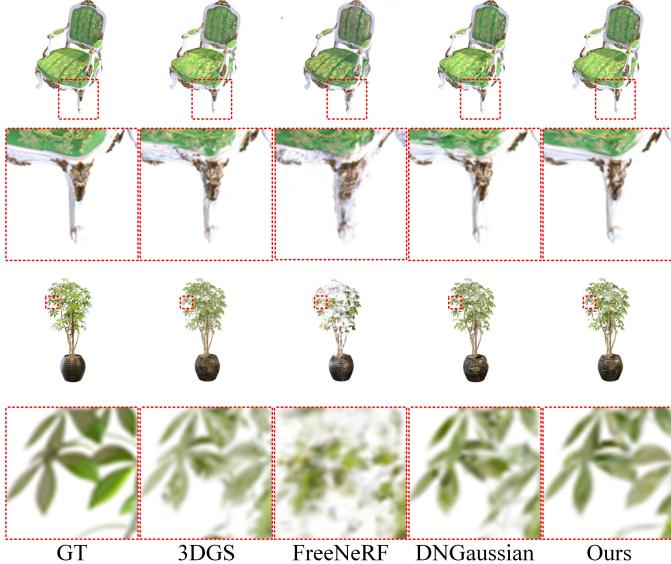


Fig. 4. Qualitative Comparison on Blender Using 8 Input Views. We present novel views and zoomed-in areas (highlighted by red boxes) rendered by our method and the baselines.

TABLE II

QUANTITATIVE COMPARISONS ON BLENDER. † INDICATES THAT THE METHOD IS IMPLEMENTED BASED ON OUR SETTING. THE BEST AND SECOND-BEST QUANTITATIVE RESULTS ARE MARKED IN RED AND GREY, RESPECTIVELY.

Method	Venue	PSNR↑	SSIM↑	LPIPS↓
DietNeRF	ICCV-21	23.15	0.866	0.109
DietNeRF, \mathcal{L}_{MSE} ft	ICCV-21	23.59	0.874	0.097
InfoNeRF	CVPR-22	22.01	0.852	0.133
MixNeRF	CVPR-23	23.84	0.878	0.103
VGOS	IJCAI-23	21.32	0.861	0.168
FreeNeRF	CVPR-23	24.26	0.883	0.098
3DGS†	TOG-23	22.77	0.865	0.112
DNGaussian	CVPR-24	23.90	0.880	0.089
Ours	–	24.06	0.887	0.089

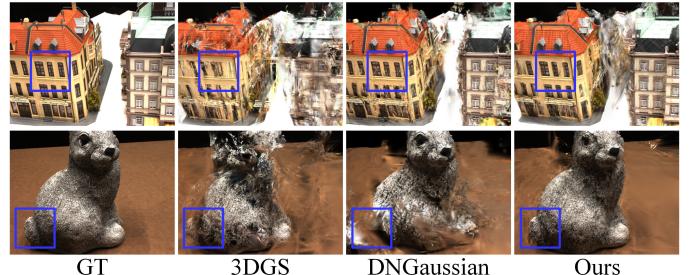


Fig. 5. Qualitative Comparisons on DTU. We show novel views rendered by our method and baseline methods in a 3-view setting. The detailed results, highlighted in blue boxes, demonstrate significant improvement.

D. Evaluation on DTU Dataset

Fig. 5 and Tab. III present the qualitative and quantitative results on the DTU dataset using 3 input views, respectively. For quantitative results, our method shows significant improvement compared to the base 3DGS architecture (surpassing 3DGS by 3 dB in terms of PSNR). Moreover, our method outperforms the recent 3DGS-based few-shot novel view synthesis method, DNGaussian. The qualitative results further confirm the improvement, with our method producing more detailed rendered results with less distortion, as highlighted in the blue box in Fig. 5.

TABLE III
QUANTITATIVE COMPARISONS ON DTU. † INDICATES THAT THE METHOD IS IMPLEMENTED BASED ON OUR SETTING. THE BEST AND SECOND-BEST QUANTITATIVE RESULTS ARE MARKED IN RED AND GREY, RESPECTIVELY.

Method	Venue	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF ft	CVPR-21	18.95	0.710	0.269
MVSNeRF ft	ICCV-21	18.54	0.769	0.197
DietNeRF	ICCV-21	11.85	0.633	0.314
RegNeRF	CVPR-22	18.89	0.745	0.190
FreeNeRF	CVPR-23	19.92	0.787	0.182
3DGS†	TOG-23	14.92	0.698	0.248
DNGaussian	CVPR-24	18.91	0.790	0.176
Ours	–	19.02	0.810	0.154

E. Method Analysis & Ablation Study

Efficiency Assessment. In this study, we propose a novel sparse initializer and a pruning strategy to enhance multi-view consistency during initialization and optimization. Our method not only significantly improves the overall consistency of the field but also represents the scene more efficiently with fewer Gaussians, leading to higher memory efficiency and faster rendering speeds. Tab. IV presents the assessment of training time (Time), FPS, number of Gaussians (#GS(k)), and visual quantitative results (PSNR, SSIM, LPIPS). The results indicate that our method achieves the best visual performance with far fewer Gaussians and offers faster training ($7\times$ faster than FSGS) and rendering speeds compared to the FSGS method. Notably, our method surpasses the base 3DGS architecture, implemented with our setting, by 1.2 dB in PSNR and achieves nearly $1.5\times$ faster rendering speed with minimal training efficiency loss (less than 20 s), which is due to the additional pruning step.

TABLE IV

EFFICIENCY COMPARISONS. OUR METHOD ACHIEVES EFFICIENT TRAINING, REAL-TIME RENDERING, AND OPTIMAL RESULTS WITH A MINIMAL NUMBER OF GAUSSIAN PRIMITIVES. THE BEST AND SECOND-BEST QUANTITATIVE RESULTS ARE MARKED IN RED AND GREY, RESPECTIVELY.

Method	Time↓	FPS↑	#GS(k)	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	3 h	0.11	-	19.63	0.612	0.308
3DGS	1.4 min	183	47	19.09	0.645	0.242
FSGS	12 min	228	194	20.37	0.696	0.205
DNGaussian	3.5 min	240	78	20.02	0.682	0.229
Ours	1.7 min	277	36	20.33	0.699	0.226

TABLE V

ABLATION OF VARIOUS INITIALIZATION METHODS. THE BEST AND SECOND-BEST QUANTITATIVE RESULTS ARE MARKED IN RED AND GREY, RESPECTIVELY.

Method	FPS↑	#GS(k)	PSNR↑	SSIM↑	LPIPS↓
3DGS w/SFM	183	47	19.09	0.645	0.242
MCGS w/SFM	194	39	20.07	0.691	0.230
3DGS w/MVS	165	50	20.04	0.694	0.209
MCGS w/MVS	246	47	20.42	0.708	0.216
3DGS w/SI	232	43	19.88	0.677	0.221
MCGS w/SI	277	36	20.33	0.699	0.226

Ablation of Different Initializer. To thoroughly assess the effectiveness of our sparse initializer (SI), we replace it with the commonly used SFM and MVS algorithms performed by COLMAP software for extremely sparse and dense initialization. We then evaluate our method (MCGS) compared to the base 3DGS architecture using these various initializers. The evaluation results are shown in Tab. V. We found that our sparse initializer significantly outperforms the COLMAP-based sparse SFM initializer on both 3DGS and our method (surpassing 0.8 dB and 0.25 dB in terms of PSNR, respectively), highlighting the advantages of our initializer. Notably, while COLMAP-based MVS initialization shows only slightly higher quantitative results, it suffers from memory inefficiency and slower rendering speeds. In contrast, the sparse initializer proposed by our method achieves a good balance between performance and efficiency. Additionally, the quantitative results demonstrate the effectiveness of our MVC-guided pruning strategy across various initialization methods, with improvements of 0.9 dB, 0.4 dB, and 0.5 dB in PSNR compared to 3DGS alone.

Notably, while COLMAP-based MVS initialization shows only slightly higher quantitative results, it suffers from high-frequency floaters (refer to Fig. 6) that our MVC-guided pruning strategy cannot effectively address. Furthermore, MVS-based initialization with more optimized Gaussian primitives leads to memory inefficiency and slower rendering speeds. In contrast, the sparse initializer proposed by our method achieves a good balance between performance and efficiency.

Ablation Study. We ablate our method on the LLFF dataset using three input views. The qualitative and quantitative results of the ablation study are presented in Fig. 7 and Tab. VI. The quantitative analysis shows that each component of our method—sparse initializer (A), random filling (B), multi-view consistency-guided progressive pruning (C), and edge-aware depth regularization (D)—contributes to improved

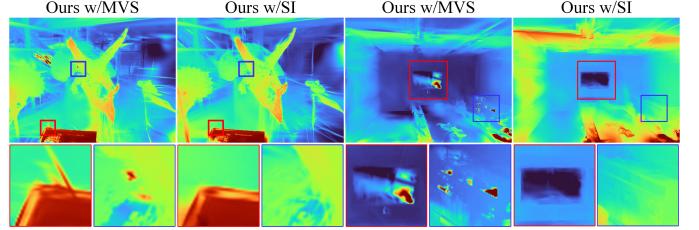


Fig. 6. The MVS-based dense initialization introduces high-frequency artifacts. In contrast, our sparse initializer effectively mitigates these issues.

performance. Specifically, the sparse initializer (A) enhances the PSNR by 0.6 dB compared to the baseline 3DGS with the COLMAP initializer. The random filling strategy (B) further refines the initial geometry, yielding an additional 0.2 dB improvement. Multi-view consistency-guided progressive pruning (C) removes inconsistent Gaussians, resulting in a 0.4 dB increase. The photometric and depth results in Fig. 7 visually demonstrate the impact of each component. Although the edge-aware depth regularization (D) provides only minor quantitative gains, it smooths depth maps, fills geometric voids, and stabilizes the overall pipeline.

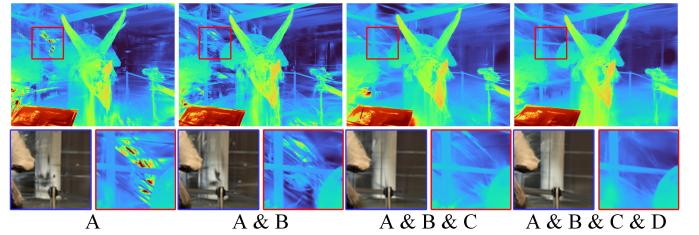


Fig. 7. **Qualitative Ablation on LLFF.** We present a qualitative ablation on the LLFF dataset with 3 input views.

TABLE VI
QUANTITATIVE ABLATION ON LLFF.

	A	B	C	D	PSNR↑	SSIM↑	LPIPS↓
3DGS	✗	✗	✗	✗	19.09	0.645	0.242
Ours	✓	✗	✗	✗	19.61	0.667	0.228
	✓	✓	✗	✗	19.88	0.677	0.221
	✓	✓	✓	✗	20.29	0.699	0.226
	✓	✓	✓	✓	20.33	0.699	0.226

V. CONCLUSION

In this paper, we propose two novel strategies for incorporating multi-view consistency priors into 3DGS during both the initialization and optimization processes. For initialization, we introduce a sparse initializer that combines a pre-trained sparse matching network with a random filling strategy to provide sparse yet sufficient initial Gaussians. For optimization, we propose a multi-view consistency-guided progressive pruning strategy to enhance the consistency of the Gaussian field and eliminate Gaussians in empty spaces. Our method achieves state-of-the-art performance while offering faster rendering speeds and minimal memory costs.

REFERENCES

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [2] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. 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*, 2022, pp. 5480–5490.
- [3] J. Yang, M. Pavone, and Y. Wang, “Freenerf: Improving few-shot neural rendering with free frequency regularization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8254–8263.
- [4] M. Kim, S. Seo, and B. Han, “Infonerf: Ray entropy minimization for few-shot neural volume rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12912–12921.
- [5] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised nerf: Fewer views and faster training for free,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12882–12891.
- [6] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, “Sparsenerf: Distilling depth ranking for few-shot novel view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9065–9076.
- [7] A. Jain, M. Tancik, and P. Abbeel, “Putting nerf on a diet: Semantically consistent few-shot view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5885–5894.
- [8] J. Sun, Z. Zhang, J. Chen, G. Li, B. Ji, L. Zhao, and W. Xing, “Vgos: voxel grid optimization for view synthesis from sparse inputs,” in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 1414–1422.
- [9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.
- [10] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, “Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20775–20785.
- [11] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, “Fsgs: Real-time few-shot view synthesis using gaussian splatting,” *arXiv preprint arXiv:2312.00451*, 2023.
- [12] J. Zhang, J. Li, X. Yu, L. Huang, L. Gu, J. Zheng, and X. Bai, “Cor-gs: Sparse-view 3d gaussian splatting via co-regularization,” *arXiv preprint arXiv:2405.12110*, 2024.
- [13] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [14] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.
- [15] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [16] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, “Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5610–5619.
- [17] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. 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*, 2022, pp. 12892–12901.
- [18] S. Seo, D. Han, Y. Chang, and N. Kwak, “Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20659–20668.
- [19] M.-S. Kwak, J. Song, and S. Kim, “Geconerf: few-shot neural radiance fields via geometric consistency,” in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 18023–18036.
- [20] K.-E. Lin, Y.-C. Lin, W.-S. Lai, T.-Y. Lin, Y.-C. Shih, and R. Ramamoorthi, “Vision transformer for nerf-based view synthesis from a single input image,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 806–815.
- [21] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.
- [22] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, “Ibrnet: Learning multi-view image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.
- [23] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang, “Sparseneus: Fast generalizable neural surface reconstruction from sparse views,” in *European Conference on Computer Vision*. Springer, 2022, pp. 210–227.
- [24] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, “Stereo radiance fields (srif): Learning view synthesis for sparse views of novel scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7911–7920.
- [25] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, “Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14124–14133.
- [26] Y. Lao, X. Xu, X. Liu, H. Zhao *et al.*, “Corresnerf: Image correspondence priors for neural radiance fields,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [27] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, “Sparf: Neural radiance fields from sparse and noisy poses,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4190–4200.
- [28] J. Wynn and D. Turmukhambetov, “Diffusionerf: Regularizing neural radiance fields with denoising diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4180–4189.
- [29] J. R. Shue, E. R. Chan, R. Po, Z. Ankner, J. Wu, and G. Wetzstein, “3d neural field generation using triplane diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20875–20886.
- [30] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, “Lightglue: Local feature matching at light speed,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17627–17638.
- [31] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.
- [32] M. Oquab, T. Darcey, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, “Dinov2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2023.
- [33] P. Heise, S. Klose, B. Jensen, and A. Knoll, “Pm-huber: Patchmatch with huber regularization for stereo matching,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2360–2367.
- [34] M. Comi, A. Tonioni, M. Yang, J. Tremblay, V. Blukis, Y. Lin, N. F. Lepora, and L. Aitchison, “Snap-it, tap-it, splat-it: Tactile-informed 3d gaussian splatting for reconstructing challenging surfaces,” *arXiv preprint arXiv:2403.20275*, 2024.
- [35] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.
- [36] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, “Large scale multi-view stereopsis evaluation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 406–413.
- [37] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19447–19456.