Initialize to Generalize: A Stronger Initialization Pipeline for Sparse-View 3DGS

Feng Zhou*, Wenkai Guo*, Pu Cao, Zhicheng Zhang†, Jianqin Yin†

***Abstract*—Sparse-view 3D Gaussian Splatting (3DGS) often overfits to the training views, leading to artifacts like blurring in novel view rendering. Prior work addresses it either by enhancing the initialization (*i.e.*, the point cloud from Structure-from-Motion (SfM)) or by adding training-time constraints (regularization) to the 3DGS optimization. Yet our controlled ablations reveal that initialization is the decisive factor: it determines the attainable performance band in sparse-view 3DGS, while training-time constraints act as complementary refinements whose effectiveness depends on the initial geometric prior. Given initialization's primacy, we focus our design there. Although SfM performs poorly under sparse views due to its reliance on feature matching, it still provides reliable seed points. Thus, building on SfM, our effort aims to supplement the regions it fails to cover as comprehensively as possible. Specifically, we design: (i) frequency-aware SfM that improves low-texture coverage via low-frequency view augmentation and relaxed multi-view correspondences; (ii) 3DGS self-initialization that lifts photometric supervision into additional points, compensating SfM-sparse regions with learned Gaussian centers; and (iii) point-cloud regularization that enforces multi-view consistency and uniform spatial coverage through simple geometric/visibility priors, yielding a clean and reliable point cloud. Our experiments on LLFF and Mip-NeRF360 demonstrate consistent gains in sparse-view settings, establishing our approach as a stronger initialization strategy.**

***Index Terms*—Novel View Synthesis, Sparse-view 3DGS, Generalization**

I. Introduction

NOVEL view synthesis [1]–[4] plays a central role in applications such as virtual and augmented reality [5], free-viewpoint video [6], and digital content creation [7], [8], where the goal is to generate realistic novel views from captured imagery. To enable such capabilities, recent advances in 3D neural scene representations [1], [9] have demonstrated remarkable performance, with 3D Gaussian Splatting (3DGS) [10] emerging as a highly efficient alternative to neural radiance fields [1]. In practice, however, data acquisition is often limited to a small number of viewpoints due to constraints in hardware, cost, or capture conditions. Under such sparse-view inputs, vanilla 3DGS is prone to severe overfitting to the training views [11]–[13], resulting in noticeable artifacts and persistent floaters when rendering unseen views.

To address this challenge, existing solutions broadly fall into two categories. The first focuses on enhancing the initial point cloud of 3DGS, which is typically obtained from Structure-from-Motion (SfM) [14]. For instance, CoMapGS [15] leverages a depth estimator together with a robust visual encoder to produce a stronger initialization. The second line of work constrains the growth or optimization of Gaussian primitives, often by introducing regularization losses during training. A representative example is CoR-GS [16], which prunes outlier primitives by enforcing consistency between two parallel Gaussian fields. While such methods have shown effectiveness in practice, our evidence suggests that training-time regularization mainly serves as a patch for poor initialization rather than addressing the fundamental bottleneck in sparse-view 3DGS.

We begin with ablation studies comparing the effectiveness of two strategies for mitigating overfitting. In Section III, we evaluate several state-of-the-art training-time constraints across a range of initialization strengths. These initialization strengths are explicitly controlled ideally (rather than being determined solely by the input images). The results show that initialization quality sets the attainable performance band, whereas training-time constraints mainly refine the optimization within the band induced by the initial geometry and can further improve performance when paired with a strong initialization. The findings indicate that initialization is the more influential lever, which motivates focusing on developing a stronger initialization pipeline.

We start from the vanilla SfM algorithm since it provides reliable initial seed points, though it produces only a few in sparse-view scenarios due to its feature-matching nature. Our work is dedicated to accurately supplementing those undercovered regions by fully exploiting the visual information contained in the available images, step by step. First, inspired by EAP-GS [12], we modify the standard SfM pipeline by relaxing the minimum track matching numbers from three to two, yielding a denser initial point set. Moreover, distinct from prior work, we pre-mask high-frequency regions and perform SfM on the resulting augmented, doubled image set, which encourages a more balanced and richer point distribution. Secondly, to further enrich the point set, we propose 3DGS Self-Initialization, which lifts per-pixel photometric constraints into additional points by leveraging the learning signal of 3DGS. Specifically, after training a first-pass lightweight 3DGS on the input views and reusing all primitive centers as a point cloud, we are able to compensate for regions with insufficient image features. Finally, we introduce further regularization techniques to refine the final point cloud by removing noise and unreliable points: (1) discarding points observed from only a single view due to depth ambiguity, (2) applying

Feng Zhou, Wenkai Guo, Pu Cao, Zhicheng Zhang, and Jianqin Yin are with the Beijing University of Posts and Telecommunications.

*These authors contributed equally to this work and should be considered co-first authors. Email:zhoufeng@bupt.edu.cn

†Jianqin Yin and Zhicheng Zhang are corresponding authors. Email: jqyin@bupt.edu.cn

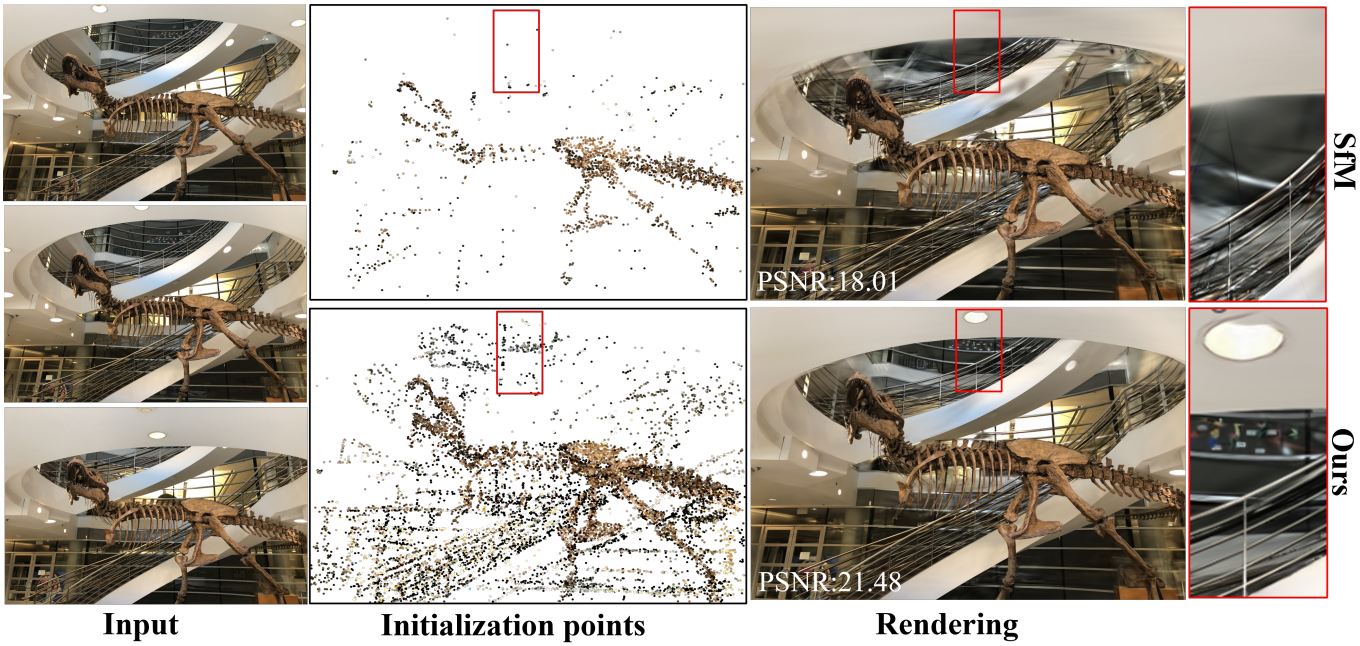


Fig. 1: A Stronger Initialization for Sparse-View 3DGS. Standard SfM (top row) generates a sparse point cloud (middle) under sparse inputs, leading to artifacts and blurriness in the 3DGS rendering (right). Our proposed initialization pipeline (bottom row) yields a significantly denser and more reliable point cloud (middle), which dramatically improves the final rendering quality (right), especially in challenging low-texture regions, thus setting a stronger foundation for high-quality novel view synthesis under sparse views.

clustering-based noise reduction, and (3) enforcing normal-based consistency filtering.

To summarize, our contributions are as follows:

- Our controlled ablations with explicitly set initialization strengths show that sparse-view performance in 3DGS is initialization-limited, and training-time constraints provide only modest improvement.
- We design a rich and reliable three-stage initialization: (i) a low-frequency-aware SfM variant that relaxes the minimum track length from three to two and pre-masks high-frequency content to improve low-texture coverage; (ii) 3DGS Self-Initialization that trains a first-pass 3DGS and reuses all primitive centers as the initial point cloud, turning photometric cues into additional points; and (iii) point-cloud regularization for removing noise or unreliable points, including removal of single-view points, clustering-based denoising, and a simple geometric prior for consistency and coverage.
- Experiments demonstrate that our proposed method achieves state-of-the-art performance on LLFF and Mip-NeRF360 datasets and offers a superior initialization choice for sparse-view 3DGS.

II. Related Work

A. Novel View Synthesis

Novel view synthesis (NVS) aims to generate photorealistic images from novel viewpoints given only a finite set of calibrated observations of a typically static scene [17], [18]. Among the different strategies, optimization-based methods reconstruct a full 3D scene representation, from which novel views can be rendered at arbitrary camera poses. Neural scene representations [19] have recently emerged as the dominant paradigm: NeRF, for example, encodes view-dependent color and density and renders images through volumetric integration [1]. Beyond such implicit fields, explicit formulations like 3D Gaussian Splatting (3DGS) represent scenes as anisotropic Gaussians and achieve real-time rendering with competitive quality [10], [20]–[22]. The explicit and discrete nature of the Gaussian primitives makes 3DGS highly sensitive to the initial geometry, contrasting with the robustness of volumetric methods like NeRF. This high sensitivity makes the initialization of the point cloud a critical determinant of final performance in 3DGS. In parallel, generative approaches that exploit large-scale priors have also been developed [3], [23]–[27]. These methods provide extremely fast inference and reduced reliance on precise camera pose annotations, but they remain limited in maintaining multi-view consistency and geometric fidelity.

3D Reconstruction with Sparse Views. Reconstructing 3D scenes from sparse views is inherently under-constrained: limited viewpoints lead to incomplete surface coverage, ambiguous depth estimation, and frequent artifacts such as floaters and texture misalignment. The standard initialization for 3DGS relies on Structure-from-Motion (SfM) algorithms like COLMAP, which performs poorly under sparse-view conditions. SfM relies heavily on texture-rich areas for feature matching, leading to sparse and incomplete initial point clouds that often lack coverage in smooth or low-texture regions (e.g., walls, skies). To address these challenges, some NVS works

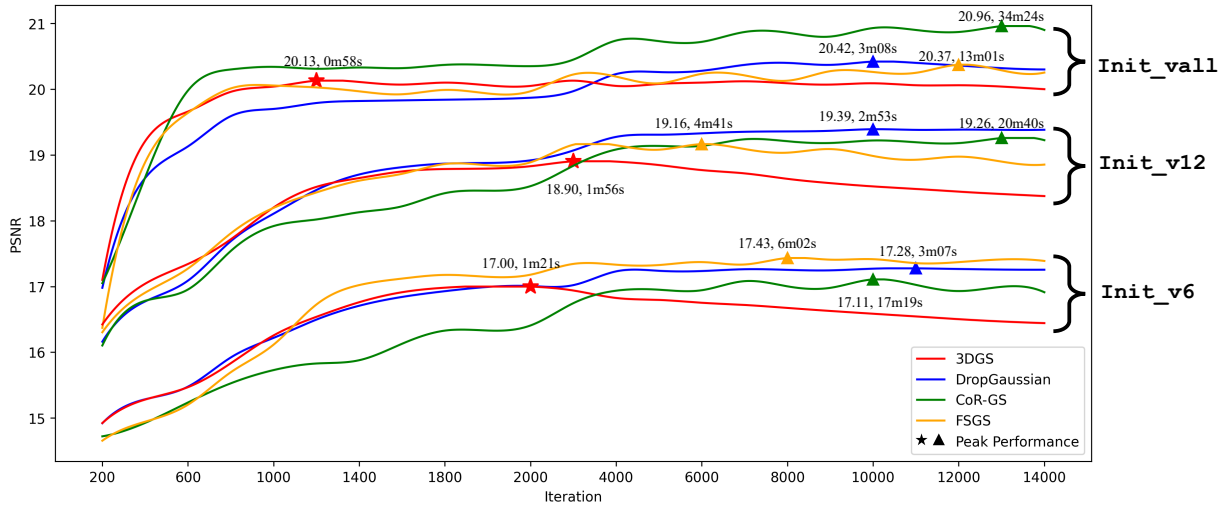


Fig. 2: Ablation study on Mip-NeRF360 under the 12-view sparse-view protocol PSNR on average, comparing three optimization constraints (DropGaussian, CoR-GS, and FSGS) against vanilla 3DGS across different initialization strengths (`Init_v6`, `Init_v12`, `Init_vall`). Results show that while regularization methods modestly delay overfitting and improve performance within each initialization band, the final reconstruction quality is primarily governed by initialization strength.

incorporate additional geometric and photometric priors, introducing supervision such as depth [28], [29] to guide the optimization process. Another line leverages controllable 2D generative models to synthesize novel training views, effectively densifying the supervision and mitigating the scarcity of input observations [30]–[35].

B. Overfitting in Sparse-view 3DGS

Under sparse views, vanilla 3DGS optimized from photometric supervision tends to overfit the limited inputs, degrading markedly on unseen view rendering. Efforts to mitigate overfitting generally take two forms, the first being to enhance the initialization of the 3DGS point cloud [15], [36], [37]. Intuitively, initialization provides the spatial and geometric bias for subsequent optimization; with a better-distributed seed point cloud, the system is guided toward plausible geometry rather than overfitting to limited photometric cues. For instance, LoopSparseGS [36] introduces a progressive Gaussian initialization strategy that iteratively densifies the seed point cloud using pseudo-views, significantly improving coverage and stability under sparse inputs. EAP-GS [12] augments the SfM point cloud by relaxing track constraints and adaptively densifying underrepresented regions, providing a stronger initialization for few-shot 3DGS. These initialization-focused approaches directly address the root cause of sparse-view failure: insufficient geometric prior.

The second form constrains the optimization or growth of Gaussian primitives, typically by introducing regularization losses or pruning strategies to suppress overfitting artifacts [37]–[41]. For instance, CoR-GS [16] introduces a co-regularization framework that enforces consistency between two parallel Gaussian fields, effectively pruning outliers and improving generalization under sparse views. FSGS [11] imposes frequency-based structural regularization, suppressing high-frequency artifacts and stabilizing training when input views are highly limited. DropGaussian [38] proposes a dropout-style scheme that randomly removes Gaussians during training, enhancing gradient flow to the remaining primitives and mitigating overfitting without extra priors.

As visually and quantitatively illustrated in Figure 2, we argue that existing optimization constraints—such as DropGaussian and CoR-GS—provide valuable incremental refinements but ultimately operate within performance bounds dictated by the quality of the initial point cloud. This distinction is crucial: regularization acts as an optimization-level constraint designed to mitigate the symptoms of overfitting (e.g., elongated Gaussians), whereas robust initialization addresses the fundamental geometric prior. Our work is motivated by the observation that a weak, sparse initial geometry sets a critically low performance floor that training-time fixes struggle to overcome. By proposing an initial point cloud that is denser, cleaner, and inherently more geometrically consistent than standard SfM, we fundamentally elevate the performance landscape. This stronger geometric prior not only achieves superior results in isolation but also provides a robust foundation that significantly amplifies the efficacy of subsequent optimization constraints, setting a new, higher performance level for sparse-view 3DGS.

III. Initialization vs. Regularization: An Empirical Study

A. Preliminary and Definition

Structure-from-Motion (SfM) [14]. Given images $\{I_i\}_{i=1}^{N}$ with intrinsics $\{K_i\}$, SfM estimates camera poses $P_i = (R_i, t_i)$ and a sparse 3D point set $\mathcal{X} = \{X_m \in \mathbb{R}^3\}$ from multi-view feature tracks. Tracks link projections of the same 3D point across views. In standard SfM system (*i.e.*, COLMAP [42]), tracks observed in at least three images are retained for

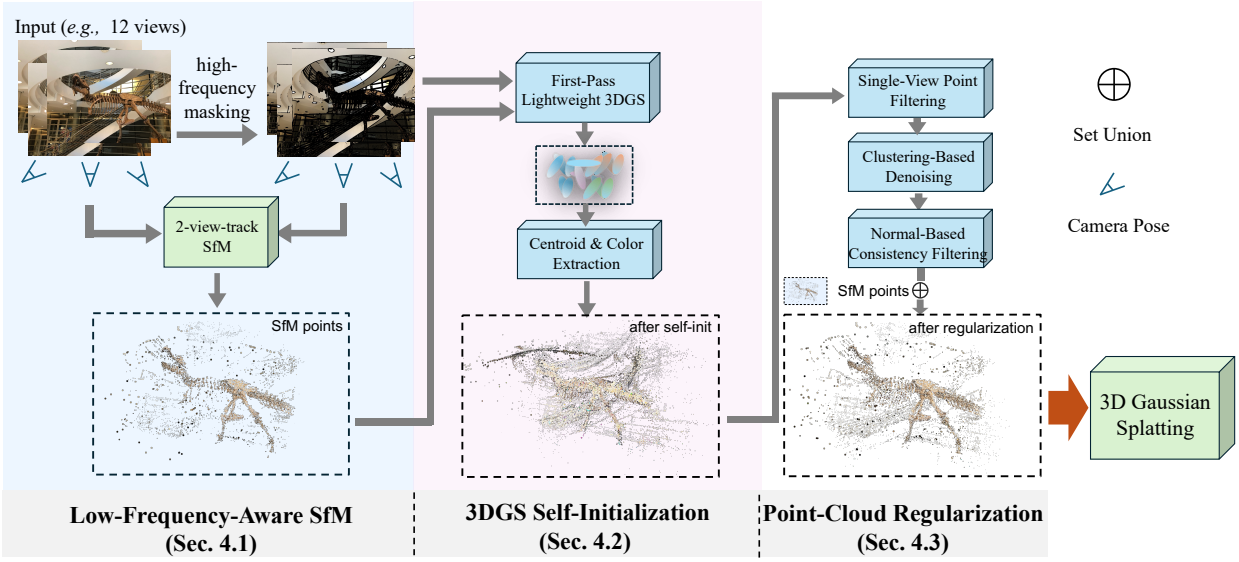


Fig. 3: Our initialization pipeline for sparse-view 3DGS. Given sparse multi-view images with camera poses, we first mask high-frequency regions and perform SfM to obtain a raw point cloud, which better captures smooth areas. To compensate for regions lacking distinctive features, we train a lightweight first-pass 3DGS and incorporate its primitive centroids into the initialization. Finally, we apply single-view filtering, clustering-based denoising, and normal-based consistency filtering to rectify the point cloud and suppress noise.

triangulation and bundle adjustment, yielding a conservative but reliable reconstruction. Candidate points are obtained via two-view triangulation T:

$$X_m \;=\; T\big(\tilde{x}_{i,m},\, \tilde{x}_{j,m},\, P_i,\, P_j\big), \quad i \neq j, \tag{1}$$

Formally,

$$\{\hat{P}_i\},\, \{\hat{X}_m\} = \underset{\{P_i\},\, \{X_m\}}{\arg\min} \sum_{m:\, |\mathcal{V}_m| \geq 3} \sum_{i \in \mathcal{V}_m} \rho\Big(\big\|\pi(K_i, P_i, X_m) - \tilde{x}_{i,m}\big\|_2^2\Big) \tag{2}$$

where $\mathcal{V}_m = \{\, i : \text{point } m \text{ observed in image } i \,\}$, $\tilde{x}_{i,m} \in \mathbb{R}^2$ is the observed pixel, $\pi(\cdot)$ denotes the standard perspective projection with intrinsics K_i and pose (R_i, t_i), and $\rho(\cdot)$ is a robust loss.

3D Gaussian Splatting (3DGS) [10]. 3DGS represents a scene with anisotropic 3D Gaussian primitives $\mathcal{G} = \{g_n\}_{n=1}^{M}$, each $g_n = (\mu_n, \Sigma_n, \alpha_n, \mathbf{c}_n)$ (mean, covariance, opacity, color). In standard practice, primitives are initialized from the SfM point cloud $\{X_m\}$, with both their positions and colors initialized from the reconstructed points, and then optimized end-to-end with fixed camera poses. During training, the primitive set is adaptively densified by splitting or cloning Gaussians to grow $\mathcal{G}$ in regions indicated by image evidence. Rendering produces predictions $\hat{I}_i$ via differentiable alpha compositing of depth-sorted 2D projections of the Gaussians, and the parameters are learned via a compact photometric objective with a structural-similarity term and regularization:

$$\min_{\mathcal{G}} \sum_{i=1}^{N} \Big[(1-\lambda) \sum_{p \in \Omega_i} \big\|\hat{I}_i(p) - I_i(p)\big\|_1 + \lambda\, \text{D-SSIM}\big(\hat{I}_i, I_i\big)\Big] + \beta\, \mathcal{R}_{\text{reg}}(\mathcal{G}), \tag{3}$$

where Ω_i is the pixel domain, $\lambda \in [0,1]$ balances the terms, I_i and $\hat{I}_i$ are the i-th ground-truth and rendered images, $\text{D-SSIM}(\hat{I}_i, I_i) = 1 - \text{SSIM}(\hat{I}_i, I_i)$, and $\mathcal{R}_{\text{reg}}$ collects mild priors on scale/opacity.

Initialization Strength. Ideally, a strong initialization should approximate a point cloud in which the point set fully covers all object visible surfaces. Such an ideal seed could, in principle, be obtained by running SfM on a sufficiently large set of views of the scene. Since unlimited viewpoints are not available, we approximate initialization strength per scene by varying the number of input views for SfM.

Formally, let $\mathcal{V}_{\text{all}}$ denote all available views of a scene, and let the SfM reconstruction from $\mathcal{V}_{\text{all}}$ serve as a best-available proxy for an ideal initialization, producing a point cloud $\mathcal{X}_{\text{all}}$. For a view budget n, we uniformly sample a subset $\mathcal{V}_n \subset \mathcal{V}_{\text{all}}$ with $|\mathcal{V}_n| = n$ in camera pose space, run SfM on $\mathcal{V}_n$, and obtain a seed point cloud $\mathcal{X}_n$. Larger n therefore corresponds to higher initialization strength in expectation. On the Mip-NeRF360 [43] dataset, we instantiate three levels of initialization strength: `Init_v6`, `Init_v12`, and `Init_vall`, where the subscript denotes the number of input views used by SfM (with `Init_vall` using all available views).

B. Empirical Analysis and Findings

To investigate the causes of sparse-view overfitting, we study two common strategies: (i) improving the SfM initialization cloud and (ii) adding training-time regularization of Gaussian primitives. These strategies are treated as two factors in a controlled ablation, enabling us to isolate their individual and combined effects. Specifically, for training-time constraints, we adopt representative off-the-shelf methods to reflect current practice: FSGS [11], Cor-GS [16], and DropGaussian [38]. For initialization, rather than comparing heterogeneous approaches, we treat our designed *Initialization Strength* as a controlled variable to isolate its effect. Experiments are conducted on the Mip-NeRF360 dataset using the standard 12-view sparse-view setting adopted in prior sparse-view 3DGS evaluations [11], [38]: 12 input views are uniformly selected for training, and average PSNR is reported across all scenes.

The performance curves are shown in Figure 2. Across all three initialization levels, different regularization methods show some effectiveness: compared with vanilla 3DGS, they mitigate overfitting by delaying the performance peak and slightly improving reconstruction quality, albeit at the cost of increased computation. Strikingly, the strength of initialization proves decisive for the final outcome. The curves stratify into distinct bands according to initialization level, while regularization mainly refines the solution within, or slightly above, the performance band induced by the initial geometry. Thus, our analysis does not imply that regularization is unnecessary; rather, it shows that a stronger initialization provides a better operating point for subsequent regularization. This initialization-dominated phenomenon motivates us to design a more effective initialization strategy for sparse-view 3DGS.

IV. METHOD

In this section, we aim to develop a rich and reliable initialization pipeline tailored for 3D Gaussian Splatting (3DGS) under sparse-view settings. Building upon vanilla 3DGS, our objective is to compensate for undercovered regions by fully leveraging the visual information available in the input images. The overall framework is illustrated in Figure 3. It comprises three successive components: 1) Low-Frequency-Aware SfM (Sec. IV-A), which augments vanilla SfM with high-frequency masking to improve initialization in smooth regions; 2) 3DGS Self-Initialization (Sec. IV-B), which exploits the pixel-level photometric supervision of 3DGS to supplement points in regions with few distinctive features; and 3) Point-Cloud Regularization (Sec. IV-C), which applies clustering, filtering, and geometry-inspired constraints to refine the point cloud and suppress noise.

A. Low-Frequency-Aware Structure-from-Motion (SfM)

Under sparse-view capture, limited overlap makes tracks with $|\mathcal{V}_m| \geq 3$ scarce in vanilla SfM, leading to under-covered and uneven point distributions. To better exploit limited views, EAP-GS [12] retains two-view tracks in addition to the standard three-view requirement with fixed camera poses $\hat{P}_i$.

$$\{\hat{X}'_m\} = \underset{\{X'_m\}}{\operatorname{argmin}} \sum_{m:\,|\mathcal{V}_m|\geq 2} \sum_{i\in\mathcal{V}_m} \rho\left(\left\|\pi(K_i, \hat{P}_i, X'_m) - \tilde{x}_{i,m}\right\|_2^2\right). \tag{4}$$

where camera poses $\hat{P}_i$ can be either provided or estimated from SfM. Unlike EAP-GS, which identifies low-density regions and performs SfM twice merely to enlarge the point set, we adopt a more effective strategy. High-frequency regions naturally produce more track matches in SfM due to their richer image features, leading to an undesired concentration of points. To mitigate this, we pre-mask these regions and feed two image sets into SfM simultaneously, thereby generating a more balanced point cloud.

Formally, given N scene views $\{I_i\}_{i=1}^N$ with camera poses $\{P_i\}_{i=1}^N$, we derive low-frequency images $\{I_{mask}\}$ by dividing each grayscale image into non-overlapping 32×32 blocks and masking blocks above the $\tau_f = 0.90$ quantile of Fourier energy outside the centered 10×10 region. We then build an augmented view set $\mathcal{I}^{\text{aug}} = \{I_i, I_{mask}\}_{i=1}^N$, extract features on all $2N$ views, and obtain the initial 3D point set $\mathcal{P}_0$ by running SfM once with Equation 4.

B. 3D Gaussian Splatting Self-Initialization

While SfM reconstructs 3D points by detecting and matching keypoints across views, its reliance on local features causes failures in weakly textured or repetitive regions. To overcome this bottleneck, we introduce a novel 3DGS self-initialization method, which elevates pixel-level photometric supervision into 3D space and constructs complementary point clouds by repurposing primitive centers as new points.

Light-weight 3DGS. We train a light-weight first-pass 3DGS $\mathcal{G}^{(0)}$ on low-resolution images seeded by $\mathcal{P}_0$, aiming to convert dense photometric cues into additional 3D points rather than high-fidelity rendering. Let $\mathcal{G}^{(0)} = \{(\mu_n, \Sigma_n, \alpha_n, \mathbf{f}_n)\}_{n=1}^M$ denote the Gaussian set, where μ_n is the center, Σ_n is the anisotropic covariance parameterized by scale and rotation, α_n is opacity, and $\mathbf{f}_n$ denotes the SH color coefficients. The Gaussian centers and DC colors are initialized from $\mathcal{P}_0$. For each sampled training view, the rendered image is supervised by the following masked photometric objective:

$$\begin{aligned} \hat{I}_i &= \mathcal{R}(\mathcal{G}^{(0)}, P_i), \\ \mathcal{L}_{\text{self}} &= (1-\lambda)\left\|M_i \odot \hat{I}_i - M_i \odot I_i^{\downarrow}\right\|_1 \\ &\quad + \lambda\left(1 - \text{SSIM}\left(\hat{I}_i, I_i^{\downarrow}\right)\right). \end{aligned} \tag{5}$$

where $\mathcal{R}(\cdot)$ denotes differentiable Gaussian rasterization, M_i is the high-frequency suppression mask used in the light-weight pass, and λ balances the photometric and structural terms. Details of the blockwise high-frequency suppression mask M_i are provided in the supplementary material. After this pass, we form a colored point set by taking, for each surviving primitive g_n, the primitive center and its DC color, *i.e.*, $\mathcal{P}_1 = \{(X_n, C_n)\}_{n=1}^M$ with $X_n = \mu_n$ and $C_n = \text{SH2RGB}(\mathbf{f}_{n,0})$.

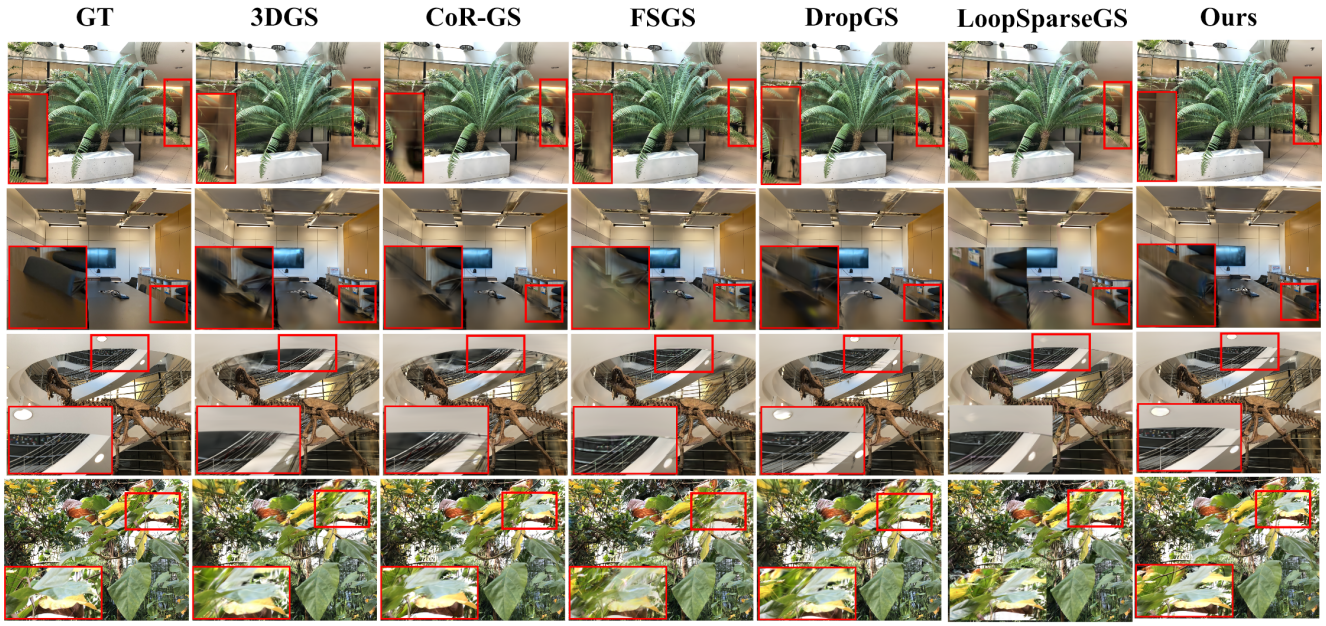


Fig. 4: Qualitative comparisons on the LLFF dataset.

C. Point Cloud Regularization

Before feeding the merged points into the final 3DGS optimization, we regularize the initial colored point cloud $\mathcal{P}_{\text{init}} = \{(X_k, C_k)\}$ obtained by combining SfM points and lightweight 3DGS points, $\mathcal{P}_{\text{init}} = \mathcal{P}_0 \cup \mathcal{P}_1$, without learned weighting. While this union improves coverage, it also aggregates errors from both sources: (i) 3DGS-generated points with only single-view supervision, which lack geometric consistency. (ii) noisy/duplicated points introduced by 3DGS split/clone densification; and (iii) outliers from unstable two-view tracks; To obtain a reliable and uniformly distributed point set, we introduce three complementary procedures: *single-view point filtering*, *clustering-based denoising*, and *normal-based consistency filtering*.

In all experiments, we apply the procedures in the listed order, with each stage taking the output of the preceding stage. We present the pseudocode in the Appendix.

Single-view Point Filtering. During 3DGS self-initialization, points visible in only one training view suffer from inherent depth ambiguity and are more likely to become floating points. We therefore use a visibility-count filter that retains all multi-view-supported points and uniformly preserves a small subset of single-view points.

Starting from the colored point cloud $\mathcal{P}_{\text{init}} = \{(X_k, C_k)\}$, we define the visible-view set of a point as

$$\mathcal{V}(X) = \{i \mid z_i(X) > \tau_d,\ \pi_i(X) \in \Omega_i\}, \tag{6}$$

where $\pi_i(\cdot)$ projects a 3D point into view i, $z_i(X)$ is its camera-space depth, Ω_i is the image domain, and $\tau_d = 0.01$ is the depth threshold. With single-view keep ratio $\rho = 0.2$, the filtered cloud is

$$\begin{aligned} \mathcal{P}_{\text{sv}} &= \{(X, C) \in \mathcal{P}_{\text{init}} \mid |\mathcal{V}(X)| = 1\}, \\ \mathcal{P}_{\text{mv}} &= \{(X, C) \in \mathcal{P}_{\text{init}} \mid |\mathcal{V}(X)| \geq 2\}, \\ \mathcal{P}_{\text{filter}} &= \mathcal{P}_{\text{mv}} \cup \text{Uniform}_{\rho}(\mathcal{P}_{\text{sv}}). \end{aligned} \tag{7}$$

Here Uniform_{ρ} samples $\max(1, \lfloor \rho |\mathcal{P}_{\text{sv}}| \rfloor)$ points without replacement when $\mathcal{P}_{\text{sv}}$ is non-empty, and returns an empty set otherwise.

Clustering-Based Point-Set Denoising. To reduce the noisy points introduced from unstable 2-view SfM tracks and duplicated points from densify process in 3DGS self-initialization, we propose a denoising technique based on the clustering algorithm, which discards 70% of the points. To formalize:

Given the single-view filtered cloud $\mathcal{P}_{\text{filter}} = \{X_k, C_k\}$,

we standardize the point coordinates and apply K-means with $K_c = \min(1000, \lfloor |\mathcal{P}_{\text{filter}}|/10 \rfloor)$ clusters. For each cluster c with centroid μ_c and size $|\mathcal{Q}_c|$, small clusters are fully retained, while larger clusters keep a 30% subset using centroid-biased sampling. The cluster-filtered cloud is represented by the retained cluster subsets:

$$\mathcal{P}_{\text{clu}} = \bigcup_{c=1}^{K_c} \mathcal{S}_c, \quad \mathcal{S}_c \subseteq \mathcal{Q}_c. \tag{8}$$

Here $\mathcal{S}_c$ retains all points if $|\mathcal{Q}_c| \leq 10$; otherwise, it keeps 30% of the points, with the nearest half selected by distance to μ_c and the remainder sampled from the cluster.

Normal-based Consistency Filtering. Finally, we remove geometrically inconsistent points by enforcing local normal agreement in 3D space. Starting from the cluster-filtered cloud $\mathcal{P}_{\text{clu}} = \{(X_k, C_k)\}$, we estimate a normal n_k for each point via PCA on its 10 nearest neighbors in Euclidean space. Let $\mathcal{N}_k$ denote the neighbor set and assume all normals are unit-length. We compute the mean cosine similarity

$$\bar{c}_k = \frac{1}{|\mathcal{N}_k|} \sum_{j \in \mathcal{N}_k} n_k^{\top} n_j, \tag{9}$$

and retain a point if $\bar{c}_k \geq 0.2$. This rejects isolated outliers and unstable estimates whose normals disagree with the local

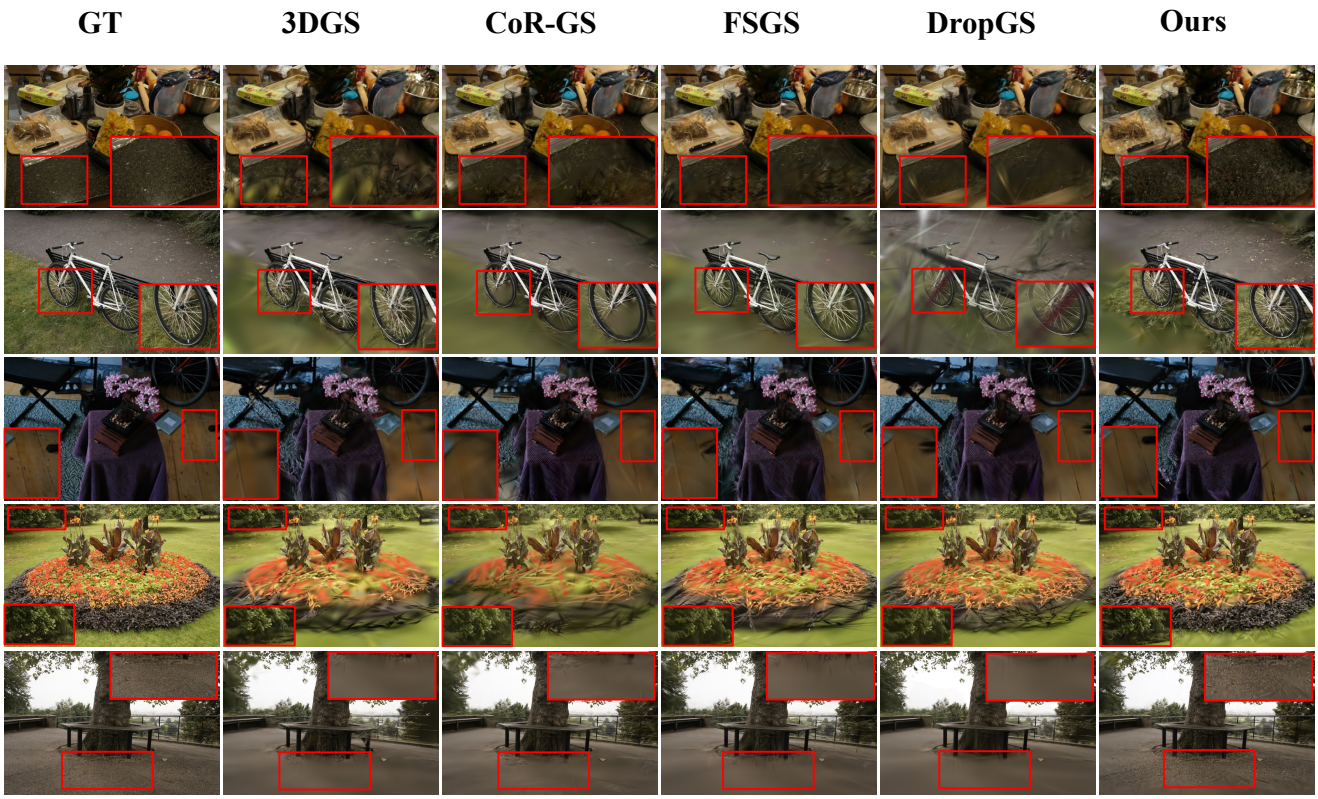


Fig. 5: Qualitative comparisons on the Mip-NeRF360 dataset.

surface. The final set is

$$\mathcal{P}_{\text{final}} = \{(X_k, C_k) \in \mathcal{P}_{\text{clu}} \mid \bar{c}_k \geq 0.2\}, \tag{10}$$

which yields a clean, geometrically consistent point cloud.

V. Experiments

A. Experimental Settings

Evaluation Datasets. We evaluate our method on two established and representative benchmarks designed for novel view synthesis under sparse inputs: Mip-NeRF360 [43] and LLFF [44]. Mip-NeRF360 represents complex, unbounded 360° scenes with challenging background details and lighting, serving as a test for significant geometric extrapolation. Conversely, LLFF provides forward-facing scenes with restricted baseline movements, representing a scenario of extreme sparsity and limited angular diversity.

Following the conventional sparse-view 3DGS setup [11], we thoroughly test our generalization ability across two distinct sparsity levels. We employ 12 uniformly distributed input views for Mip-NeRF360 to assess reconstruction capability in unbounded settings, and only 3 highly limited views for LLFF to evaluate robustness in extremely constrained forward-facing environments. Camera poses are provided by the datasets. For consistent evaluation, every eighth image in the full sequence is reserved as a test view. Furthermore, input images are uniformly downsampled by a factor of $4\times$ on both datasets. This downsampling significantly reduces GPU memory and training cost, which is essential for ensuring a fair and efficient comparison across various methods without critically compromising input resolution.

Metrics. For quantitative comparison, we report three widely-used metrics: peak signal-to-noise ratio (PSNR) for measuring reconstruction fidelity, structural similarity index (SSIM) [45] for perceived structural integrity, and learned perceptual image patch similarity (LPIPS) [46] for assessing perceptual quality.

Baselines. To demonstrate the effectiveness of our initialization strategy, we compare against several state-of-the-art 3DGS-based methods, including FSGS [11], CoR-GS [16], EAP-GS [12], LoopSparseGS [36], DDGS [47], and DropGaussian [38].

Implementation Details. Following the official training setups of each baseline, we train FSGS, CoR-GS, EAP-GS, DDGS, and DropGaussian for 10k, 10k, 5k, 10k, and 10k iterations, respectively, on the Mip-NeRF360 dataset, and for 10k, 30k, 5k, 10k, and 10k iterations on the LLFF dataset. For vanilla 3DGS and our method, we adopt 5k iterations on both benchmarks. COLMAP [42] is configured with the same parameters as FSGS [11] to initialize all baselines, except EAP-GS, which uses its official initializer, and LoopSparseGS, which uses its official loop-based pipeline. All experiments are conducted on the same hardware with a single NVIDIA RTX 4090 GPU. For the 3DGS self-initialization stage, the first-pass light-weight 3DGS also uses $4\times$ downsampled images, *i.e.*, each view is loaded at resolution $(W_i/4, H_i/4)$. We optimize this pass for 5k iterations with $\lambda = 0.2$ in Eq. 5. The active SH

TABLE I: Comparison on Mip-NeRF360 and LLFF datasets. Runtime is decomposed into initialization, training, and total time. Init., Train., and Total denote initialization time, training time, and total runtime, respectively. We color each cell as **best**, **second best**, and **third best**. †: We reimplemented vanilla 3DGS with Multi-view Stereo (MVS, a native dense reconstruction stage in COLMAP following SfM) initialization, which is used in our reproduced runs of FSGS, CoR-GS, DDGS, and DropGS. It brings sizeable performance gains but increases the initialization cost. EAP-GS uses its official initializer. LoopSparseGS is reproduced with its official pipeline; initialization and training times are not separable. A dash indicates that the corresponding setting is not provided in the original paper.

Methods	Mip-NeRF360 Dataset							LLFF Dataset						
	PSNR↑	SSIM↑	LPIPS↓	Init.↓	Train.↓	Total↓	Iter	PSNR↑	SSIM↑	LPIPS↓	Init.↓	Train.↓	Total↓	Iter
3DGS†	19.24	0.5743	0.3660	5m04s	3m52s	8m56s	5k	18.95	0.6464	0.1862	1m32s	2m28s	4m00s	5k
FSGS	19.25	0.5719	0.4072	5m04s	7m38s	12m42s	10k	19.88	0.6120	0.3400	1m32s	18m03s	19m35s	10k
CoR-GS	19.52	0.5580	0.4180	5m04s	34m07s	39m11s	30k	19.45	0.6520	0.2664	1m32s	18m13s	19m45s	10k
EAP-GS	19.21	0.5721	0.3072	2m15s	4m27s	6m42s	5k	18.84	0.6358	0.1768	2m23s	3m15s	5m38s	5k
LoopSparseGS	–	–	–	–	–	–	–	20.07	0.6606	0.3072	Not separable		29m07s	–
DDGS	19.53	0.5800	0.4058	5m04s	4m27s	9m31s	10k	19.67	0.6690	0.2681	1m32s	2m49s	4m21s	10k
DropGS	19.74	0.5770	0.3640	5m04s	6m21s	11m25s	10k	19.54	0.6549	0.1856	1m32s	5m39s	7m11s	10k
Ours	19.77	0.5892	0.3374	6m06s	4m42s	10m48s	5k	19.60	0.6681	0.1852	4m16s	1m30s	5m46s	5k
+DropGS	20.07	0.5992	0.3276	6m06s	4m32s	10m38s	10k	19.91	0.6835	0.1659	4m16s	4m17s	8m33s	10k

TABLE II: Gaussian counts after initialization and training.

Methods	Mip-NeRF360		LLFF	
	Init.GS	Final.GS	Init.GS	Final.GS
3DGS†	395k	419k	15k	42k
FSGS	395k	510k	15k	138k
CoR-GS	395k	523k	15k	94k
EAP-GS	21k	60k	7k	36k
DDGS	395k	509k	14k	96k
DropGS	395k	501k	15k	95k
Ours	29k	73k	12k	40k
+DropGS	29k	179k	12k	71k

degree starts from 0 and is increased every 1000 iterations up to degree 3; standard 3DGS densification/pruning is enabled after 500 iterations and applied every 100 iterations. The Gaussian point cloud used in Sec. IV-B is extracted from the saved primitives at iteration 5000. Additional implementation details of our method are provided in the supplementary material.

B. Performance Evaluation

Quantitative Results. Table I reports the sparse-view results. On the highly challenging Mip-NeRF360 (12 views) dataset, our proposed initialization pipeline surpasses all existing regularization-centric baselines, including DropGS and FSGS. Our 5k-iteration model achieves the second-best PSNR (19.77), outperforming the baseline 3DGS† by a significant margin (0.53 dB). Combined with DropGS, our initialization achieves the best PSNR and SSIM with competitive LPIPS on Mip-NeRF360, and leads most metrics on LLFF. LoopSparseGS, which also improves Gaussian initialization, achieves a higher LLFF PSNR but requires a substantially longer total runtime (29m07s). The DDGS entries in Table I are our reproduced results.

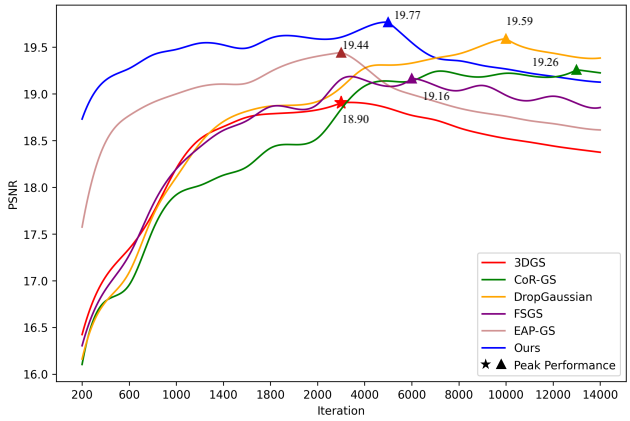


Fig. 6: Performance curves between ours and baselines on the Mip-NeRF360 dataset.

Table II reports the Gaussian counts after initialization and training. Ours uses substantially fewer Gaussians than MVS-initialized baselines while achieving better quality, indicating that the improvement comes from initialization quality rather than simply increasing primitive count.

Efficiency. We analyze the performance curves on the Mip-NeRF360 dataset in Figure 6. While competing approaches typically require 10k iterations or more, our approach reaches its final result at 5k iterations. This accelerated convergence provides evidence that our three-stage pipeline furnishes the 3DGS optimizer with a better initial state. Specifically, the geometric prior reduces the severity of the initial densification and pruning cycles in 3DGS optimization.

Rendering Results. Figures 4 and 5 separately compare the methods on LLFF and Mip-NeRF360. On LLFF, Ours preserves sharper foliage, railing, and furniture boundaries than the baselines, including LoopSparseGS; on Mip-NeRF360, it

TABLE III: Ablation study on the Mip-NeRF360 dataset. Each module is incrementally added to vanilla 3DGS.

Configuration	PSNR↑	SSIM↑	LPIPS↓
vanilla 3DGS	18.52	0.5230	0.4150
+ Low-Frequency-Aware SfM (Sec. IV-A)	$19.25^{\uparrow 0.73}$	$0.5758^{\uparrow 0.053}$	$0.3575^{\downarrow 0.0575}$
+ 3DGS Self-Initialization (Sec. IV-B)	$19.42^{\uparrow 0.17}$	$0.5924^{\uparrow 0.017}$	$0.3246^{\downarrow 0.0329}$
+ Single-View Point Filtering (Sec. IV-C)	$19.49^{\uparrow 0.07}$	$0.5872^{\downarrow 0.005}$	$0.3363^{\uparrow 0.011}$
+ Clustering-Based Denoising (Sec. IV-C)	$19.61^{\uparrow 0.12}$	$0.5910^{\downarrow 0.004}$	$0.3356^{\downarrow 0.001}$
+ Normal-Based Consistency Filtering (Sec. IV-C)	$19.77^{\uparrow 0.16}$	$0.5892^{\downarrow 0.002}$	$0.3374^{\uparrow 0.002}$

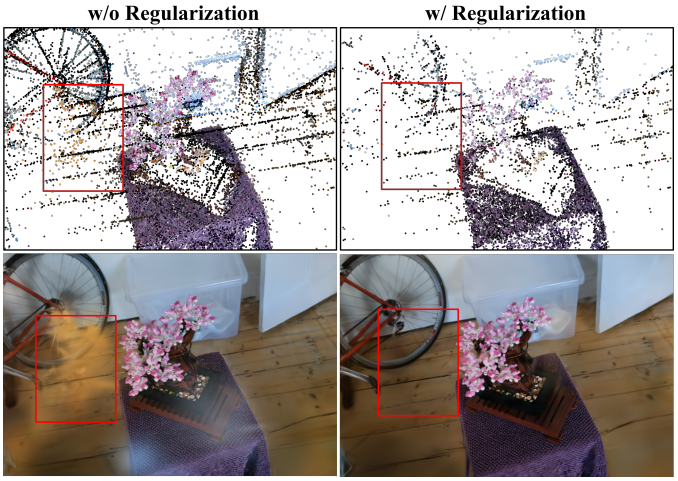


Fig. 7: Qualitative visualization of our Point Cloud Regularization pipeline. The initial point cloud (Left) derived from the merged SfM and Self-Initialization points is often noisy (highlighted in red boxes). Our regularization steps—including single-view visibility filtering, cluster-based denoising, and normal consistency checks—result in a significantly denser and cleaner point cloud (Right), providing a robust geometric prior for subsequent 3DGS optimization.

reduces blur and floating artifacts in the highlighted regions. LoopSparseGS is shown only on LLFF because no matching 12-view Mip-NeRF360 setting is available. Baselines relying heavily on regularization often suffer from two primary types of artifacts in novel views: floating artifacts, typically observed in empty spaces or near depth discontinuities, resulting from unstable high-density Gaussian generation; and visible blurring or smearing on complex textures and object edges, indicating that Gaussians have become overly stretched and elongated during optimization to cover geometric uncertainty inherited from a sparse initial state. Our multi-stage initialization lifts photometric information into additional seed points, including in low-texture areas. Consequently, our method accurately reconstructs more balanced and correct textures in challenging low-frequency regions (e.g., the large sky areas), achieves robust recovery in less-featured areas (e.g., smooth walls), and produces noticeably sharper object boundaries. More detailed visualization can be found in the Appendix.

Initialization and Optimization. Figure 8 illustrates how initialization quality affects the subsequent 3DGS optimization. Vanilla SfM produces sparse and fragmented points, especially near scene boundaries and low-texture regions. With few reliable spatial anchors, early densification tends to follow view-specific photometric residuals and may create elongated or weakly supported Gaussians. Our low-frequency-aware SfM and self-initialization provide denser and more consistent seed points, so the optimizer starts from a more plausible geometric support. The comparison between Ours and +DropGS in Table I further indicates that training-time regularization is complementary: it can refine a strong initialization, but has limited ability to recover missing scene support from a weak one.

Time Cost Breakdown. Table I reports runtime as initialization time, training time, and total time. Our full initialization pipeline, encompassing Low-Frequency-Aware SfM, 3DGS Self-Initialization, and Point Cloud Regularization, incurs a modest, one-time setup overhead (e.g., approximately 1m02s more than 3DGS† on the Mip-NeRF360 dataset) compared to relying solely on simple SfM+MVS. Given the halved training iterations required to achieve state-of-the-art performance (5k iterations vs. 10k or more for DropGS/FSGS), the amortized total runtime is comparable to, or often faster than, most competing regularization methods.

View-Count Robustness. Table IV reports a controlled Mip-NeRF360 view-count analysis from 3 to 12 input views, where our initialization improves vanilla 3DGS across all sparsity levels. In the 3- and 6-view settings, vanilla 3DGS is valid on only 4/7 and 5/7 scenes because insufficient matches prevent COLMAP/SfM from triangulating a usable seed cloud; our pipeline augments the same SfM seeds with low-frequency-aware reconstruction and self-initialized points, yielding 7/7 complete reconstructions at each view count.

Object-Centric Reconstruction. Beyond scene-level reconstruction, Table V evaluates object-centric reconstruction on DTU against representative DTU sparse-view NVS baselines [10], [16], [48], [49]. The results show strong object-centric reconstruction quality: our method remains competitive at 3 views and achieves the best PSNR in the 6/9-view settings, while keeping SSIM and LPIPS competitive.

C. Ablation Study

We evaluate the contribution of each pipeline component, sensitivity to key hyperparameters, and the effect of point-cloud regularization. Results are reported in Tables III and VII,

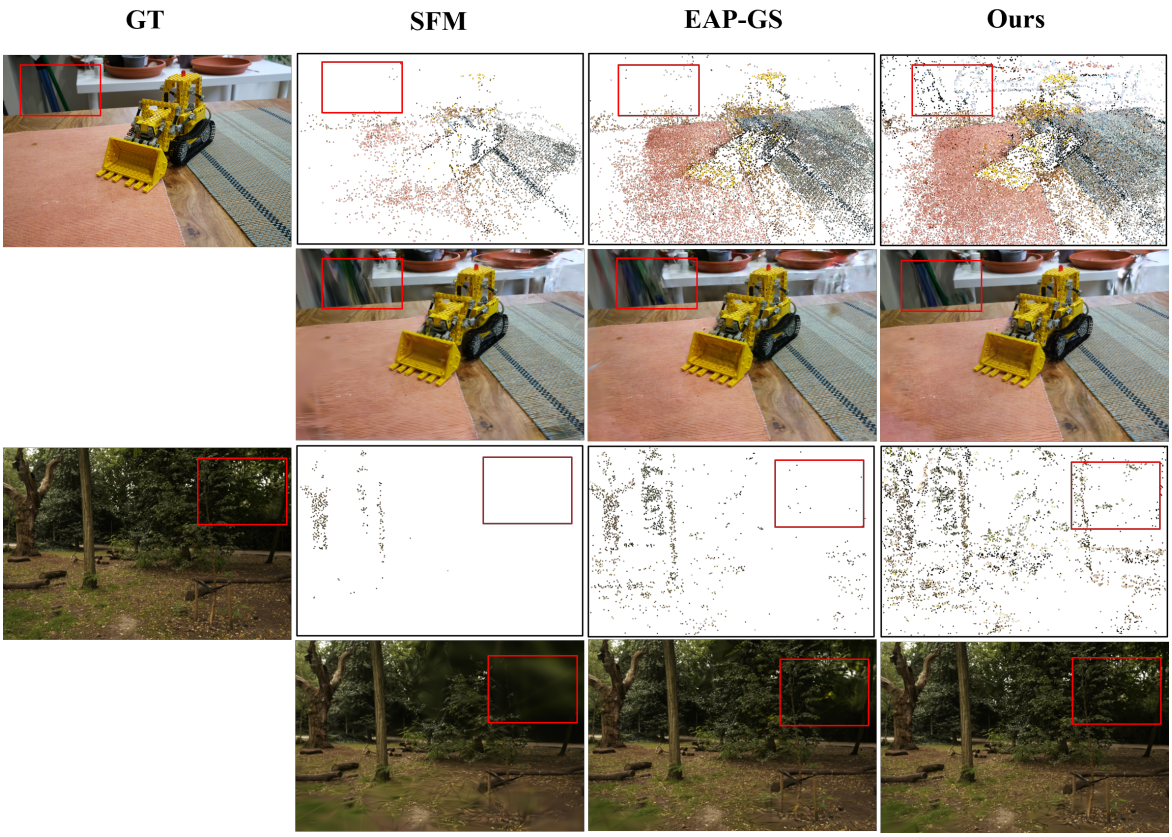


Fig. 8: Comparison of initialization quality and its impact on reconstruction performance. Our method provides more accurate initialization points, especially around scene edges, leading to improved final renderings.

TABLE IV: Mip-NeRF360 view-count analysis. We compare vanilla 3DGS and our initialization under 3/6/9/12 input views.

Views	Method	PSNR↑	SSIM↑	LPIPS↓
3	3DGS	12.77	0.3193	0.7241
	Ours	**14.23**	**0.3685**	**0.5579**
6	3DGS	14.78	0.4169	0.5969
	Ours	**16.55**	**0.4770**	**0.4328**
9	3DGS	17.10	0.4559	0.5513
	Ours	**18.25**	**0.5546**	**0.3546**
12	3DGS	19.38	0.5751	0.3894
	Ours	**19.76**	**0.5891**	**0.3373**

with clustering and alternative-prior ablations in Tables VIII and IX and qualitative results in Figure 7.

Component Analysis. Specifically, the proposed Low-Frequency-Aware SfM (Sec. IV-A) yields the single largest and most fundamental improvement, demonstrating a substantial 0.73 dB increase in PSNR over the baseline. Following this, the 3DGS Self-Initialization (Sec. IV-B) provides the next significant performance boost (0.17 dB). Furthermore, our final Point Cloud Regularization techniques (Sec. IV-C) provide an incremental but essential final refinement. By suppressing noise, filtering outliers, and removing geometric redundancy, these steps produce the final initialization used for 3DGS optimization.

Effect of Training Budget. Table VI compares longer vanilla 3DGS training with our self-initialization under comparable iteration budgets. Increasing vanilla 3DGS from 5k to 15k iterations changes PSNR only marginally (19.38 to 19.34–19.40), whereas the 5k+5k self-initialization setting reaches 19.77 PSNR with substantially better LPIPS. This ablation indicates that the gain comes mainly from the improved geometric starting point rather than simply increasing the optimization length.

Hyperparameter. Table VII evaluates the frequency-mask threshold and cluster-retention ratio. The best tested setting uses $\tau_f = 0.90$ and retains 30% of each large cluster. Retaining 20% removes more noise but can discard useful surface support, whereas 40% preserves more geometry but leaves more duplicated or floating points. These thresholds remain empirical rather than theoretically optimal; density-adaptive filtering is left for future work.

Clustering Alternatives. We also compare the default K-means clustering with density-based alternatives to assess the effect of the clustering choice. Table VIII shows that K-means gives the best PSNR, SSIM, and LPIPS under the same downstream pipeline. DBSCAN and HDBSCAN produce fewer retained points, but their density assumptions are less stable for sparse-view point clouds whose sampling density varies strongly across scene depth and texture. This supports our use of K-means as a simple and stable default, while still leaving

TABLE V: DTU object-centric benchmark. We compare with representative DTU sparse-view NVS baselines under 3/6/9 views and report PSNR/SSIM/LPIPS.

Method	3 views			6 views			9 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RegNeRF [48]	18.89	0.75	0.19	22.20	0.84	0.12	24.93	0.88	0.09
FreeNeRF [49]	19.92	0.79	0.18	23.25	0.84	0.14	25.38	0.89	0.10
3DGS [10]	17.65	0.82	0.15	24.00	0.91	0.08	26.85	0.94	**0.05**
CoR-GS [16]	19.21	**0.85**	**0.12**	24.51	**0.92**	**0.07**	27.18	**0.95**	**0.05**
Ours	**19.58**	0.80	0.18	**27.54**	0.91	0.10	**30.03**	**0.95**	0.06

TABLE VI: Self-initialization versus longer vanilla training on Mip-NeRF360. T1 and T2 denote the first-pass self-initialization and final 3DGS training iterations, respectively.

Variant	Iterations	PSNR↑	SSIM↑	LPIPS↓	Time(s)↓
Vanilla 3DGS	5k	19.38	0.5751	0.3894	159
Vanilla 3DGS	10k	19.40	0.5752	0.3891	286
Vanilla 3DGS	15k	19.34	0.5737	0.3882	405
Self-init	3k+5k	19.31	0.5779	0.3836	300
Self-init	5k+5k	**19.77**	**0.5892**	**0.3374**	282
Self-init	7k+5k	19.70	0.5869	0.3716	515

TABLE VII: Ablation Study of Initialization Parameters on Mip-NeRF360 (PSNR↑, SSIM↑, LPIPS↓). Results for the Low-Frequency-Aware SfM Threshold and the Point-Cloud Regularization Ratio are presented.

Freq. Threshold				**Cluster Ratio**			
Thr.	PSNR	SSIM	LPIPS	Ratio	PSNR	SSIM	LPIPS
0.85	19.46	0.5782	0.3547	0.2	19.51	0.5825	0.3514
0.90	**19.77**	**0.5892**	**0.3374**	0.3	**19.77**	**0.5892**	**0.3374**
0.95	19.61	0.5837	0.3460	0.4	19.36	0.5808	0.3497

TABLE VIII: Clustering alternatives for point-cloud denoising on Mip-NeRF360.

Method	PSNR↑	SSIM↑	LPIPS↓	Init.GS	Final.GS
K-means	**19.76**	**0.5891**	**0.3373**	33k	77k
DBSCAN	19.46	0.5784	0.3849	25k	66k
HDBSCAN	19.49	0.5775	0.3857	24k	65k

TABLE IX: Ablations for monocular depth regularization and learning-based initialization on Mip-NeRF360.

Setting	Variant	PSNR↑	SSIM↑	LPIPS↓
Depth prior	Ours	**19.76**	**0.5891**	**0.3373**
Depth prior	+Depth Anything V2	19.14	0.5803	0.3768
Learned init.	VGGT 5k	18.47	0.5040	0.4855
Learned init.	VGGT 10k	18.46	0.5032	0.4872
Learned init.	VGGT+self-init	18.86	0.5275	0.4431
Learned init.	Ours	**19.76**	**0.5891**	**0.3373**

adaptive density-aware clustering as future work.

Alternative Priors. Table IX examines two alternative priors for the initialization stage.

Monocular Depth Prior. Adding monocular relative-depth supervision from Depth Anything V2 decreases performance in our sparse unbounded setting, likely because affine-invariant monocular depth is unreliable in distant background and sky regions and its gradients disturb photometric geometry recovery. This result does not imply that depth priors are generally ineffective; it only shows that direct relative-depth regularization is not a robust substitute for our generate-then-filter strategy here.

Learning-Based Initialization Prior. We evaluate three learning-prior settings in Table IX: VGGT-only initializes 3DGS with VGGT points and trains the final 3DGS model; VGGT+self-init uses VGGT points as the seed for our lightweight self-initialization before final 3DGS training; Ours uses our low-frequency-aware SfM seed followed by the same self-initialization and final training. Although VGGT can provide coarse geometry without classical feature matching, its predictions may not generalize uniformly to the sparse-view, unbounded real scenes considered here, and small geometric inaccuracies can be amplified during subsequent 3DGS optimization. VGGT+self-init improves over VGGT-only, but remains below Ours.

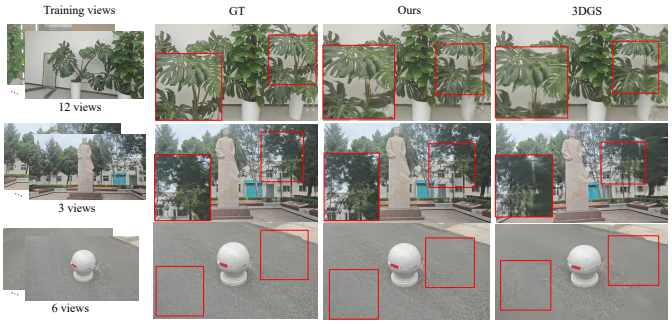


Fig. 9: Qualitative comparison on three self-captured real-world scenes. The models use 12, 3, and 6 training views, respectively, and GT denotes a held-out test view.

TABLE X: Resolution sensitivity on Mip-NeRF360 and LLFF. r is the image downsampling factor; $r = 1$ denotes original resolution.

Dataset	r	PSNR↑		SSIM↑		LPIPS↓	
		3DGS†	Ours	3DGS†	Ours	3DGS†	Ours
Mip-NeRF360	4	19.240	**19.770**	0.5743	**0.5892**	0.3660	**0.3374**
	2	19.024	**19.329**	0.6105	**0.6216**	**0.4301**	0.4548
	1	19.110	**19.140**	0.6795	**0.6844**	0.4768	**0.4348**
LLFF	4	18.950	**19.600**	0.6464	**0.6681**	0.1862	**0.1852**
	2	19.007	**19.114**	0.6632	**0.6657**	0.3019	**0.2836**
	1	18.845	**19.031**	0.6664	**0.6669**	0.4121	**0.4064**

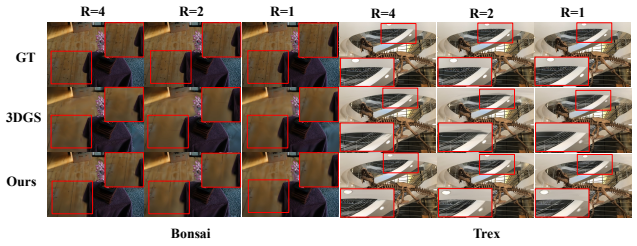


Fig. 10: Qualitative comparisons at different image resolutions on the Bonsai (Mip-NeRF360) and Trex (LLFF) scenes.

Effect of Point Cloud Regularization. Figure 7 compares the merged point cloud before and after regularization. The unregularized cloud contains noisy outliers and spurious geometry in sparse-feature regions (left). Single-view filtering, cluster-based denoising, and normal-consistency filtering remove these artifacts and produce a cleaner final cloud (right) for subsequent 3DGS optimization.

High-Resolution Results. Additionally, we conduct higher-resolution experiments against 3DGS†. Table X shows that Ours improves PSNR/SSIM in all six settings and LPIPS in five of six. While the default $4\times$ setting follows common practice, it may underrepresent fine-scale artifacts; Table X shows that our gains persist at $2\times$ and original resolution, although broader high-resolution evaluation remains future work. Figure 10 provides the corresponding visual comparison: across all three resolutions, Ours better preserves the highlighted floor textures and railing boundaries than 3DGS†.

Real-World Results. Figure 9 shows that Ours better preserves foliage, background structure, and road texture, with less blur and distortion than 3DGS.

VI. Conclusion

We have presented an analysis demonstrating that the critical bottleneck in sparse-view 3D Gaussian Splatting (3DGS) lies in the initial geometric prior: the quality, density, and geometric consistency of the seed point cloud largely dictate the final reconstruction fidelity. This initialization-dominated observation motivates our three-stage geometric initialization pipeline, which combines Low-Frequency-Aware SfM, 3DGS Self-Initialization, and Point-Cloud Regularization to obtain a cleaner and denser starting representation. Quantitative and qualitative experiments on Mip-NeRF360 and LLFF, together with ablations and generalization studies, confirm that stronger initialization provides a better foundation for high-quality novel view synthesis and remains complementary to training-time regularization.

Limitations and Future Work. Our method focuses on constructing a stronger geometric prior before 3DGS optimization, and it still assumes reasonably accurate camera poses and a usable sparse SfM seed. Extremely textureless, reflective, dynamic, or poorly calibrated captures may require stronger pose estimation or additional scene-level priors beyond the current pipeline. A promising direction is to exploit priors from large-scale pretrained vision and geometry models to infer more complete and uncertainty-aware initialization points, especially in regions where sparse-view SfM provides little support. Future work will study how to integrate such learned priors with multi-view consistency and density-aware filtering, so that the initialized point cloud becomes more adaptive while remaining geometrically reliable.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62173045), the Beijing Natural Science Foundation under Grant F2024203115, and the China Postdoctoral Science Foundation under Grant Number 2024M750255.

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] E. R. Chan, K. Nagano, M. A. Chan, A. W. Bergman, J. J. Park, A. Levy, M. Aittala, S. De Mello, T. Karras, and G. Wetzstein, "Generative novel view synthesis with 3d-aware diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4217–4229.

[3] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 9298–9309.

[4] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as new era: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[5] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, "State of the art of virtual reality technology," in *2016 IEEE aerospace conference*. IEEE, 2016, pp. 1–19.

[6] B. Zhou, S. Zheng, H. Tu, R. Shao, B. Liu, S. Zhang, L. Nie, and Y. Liu, "Gps-gaussian+: Generalizable pixel-wise 3d gaussian splatting for real-time human-scene rendering from sparse views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[7] P. Cao, F. Zhou, Q. Song, and L. Yang, "Controllable generation with text-to-image diffusion models: A survey," *arXiv preprint arXiv:2403.04279*, 2024.

[8] J. Wang, J.-C. Xie, X. Li, F. Xu, C.-M. Pun, and H. Gao, "Gaussianhead: High-fidelity head avatars with learnable gaussian derivation," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[9] 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. 12 882–12 891.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[11] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," in *European conference on computer vision*. Springer, 2024, pp. 145–163.

[12] D. Dai and Y. Xing, "Eap-gs: Efficient augmentation of pointcloud for 3d gaussian splatting in few-shot scene reconstruction," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 498–16 507.

[13] Y. Wang, J. Wang, R. Gao, Y. Qu, W. Duan, S. Yang, and Y. Qi, "Look at the sky: Sky-aware efficient 3d gaussian splatting in the wild," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[14] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[15] Y. Jang and E. Pérez-Pellitero, "Comapgs: Covisibility map-based gaussian splatting for sparse novel view synthesis," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 779–26 788.

[16] 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," in *European Conference on Computer Vision*. Springer, 2024, pp. 335–352.

[17] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[18] O. Wegen, W. Scheibel, M. Trapp, R. Richter, and J. Dollner, "A survey on non-photorealistic rendering approaches for point cloud visualization," *IEEE transactions on visualization and computer graphics*, pp. 1–20, 2024.

[19] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.

[20] X. Lin, S. Luo, X. Shan, X. Zhou, C. Ren, L. Qi, M.-H. Yang, and N. Vasconcelos, "Hqgs: High-quality novel view synthesis with gaussian splatting in degraded scenes," in *The Thirteenth International Conference on Learning Representations*.

[21] J. Zhang, F. Zhan, M. Xu, S. Lu, and E. Xing, "Fregs: 3d gaussian splatting with progressive frequency regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 424–21 433.

[22] D. Chen, H. Li, W. Ye, Y. Wang, W. Xie, S. Zhai, N. Wang, H. Liu, H. Bao, and G. Zhang, "Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction." *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 9, pp. 6100–6111, 2025.

[23] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.

[24] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–18.

[25] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, "Splatter image: Ultra-fast single-view 3d reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 10 208–10 217.

[26] V. Voleti, C.-H. Yao, M. Boss, A. Letts, D. Pankratz, D. Tochilkin, C. Laforte, R. Rombach, and V. Jampani, "Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion," in *European Conference on Computer Vision*. Springer, 2024, pp. 439–457.

[27] X. Shi, L. Wang, X. Liu, J. Wu, and Z. Shao, "Scene-aware foveated neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[28] J. Chung, J. Oh, and K. M. Lee, "Depth-regularized optimization for 3d gaussian splatting in few-shot images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 811–820.

[29] Y. Zheng, Z. Jiang, S. He, Y. Sun, J. Dong, H. Zhang, and Y. Du, "Nexusgs: Sparse view synthesis with epipolar depth priors in 3d gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 800–26 809.

[30] N. C. Mithun, T. Pham, Q. Wang, B. Southall, K. Minhas, B. Matei, S. Mandt, S. Samarasekera, and R. Kumar, "Diffusion-guided gaussian splatting for large-scale unconstrained 3d reconstruction and novel view synthesis," *arXiv preprint arXiv:2504.01960*, 2025.

[31] Y. Wan, M. Shao, Y. Cheng, and W. Zuo, "S2gaussian: Sparse-view super-resolution 3d gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 711–721.

[32] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, "Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting," *arXiv preprint arXiv:2312.00206*, 2023.

[33] Y. Chen, C. Zheng, H. Xu, B. Zhuang, A. Vedaldi, T.-J. Cham, and J. Cai, "Mvsplat360: Feed-forward 360 scene synthesis from sparse views," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 064–107 086, 2024.

[34] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai, "Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images," in *European Conference on Computer Vision*. Springer, 2024, pp. 370–386.

[35] T. Liu, G. Wang, S. Hu, L. Shen, X. Ye, Y. Zang, Z. Cao, W. Li, and Z. Liu, "Mvsgaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo," in *European Conference on Computer Vision*. Springer, 2024, pp. 37–53.

[36] Z. Bao, G. Liao, K. Zhou, K. Liu, Q. Li, and G. Qiu, "Loopsparsegs: Loop based sparse-view friendly gaussian splatting," *IEEE Transactions on Image Processing*, 2025.

[37] Y. Xu, L. Wang, M. Chen, S. Ao, L. Li, and Y. Guo, "Dropoutgs: Dropping out gaussians for better sparse-view rendering," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 701–710.

[38] H. Park, G. Ryu, and W. Kim, "Dropgaussian: Structural regularization for sparse-view gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 600–21 609.

[39] A. Paliwal, W. Ye, J. Xiong, D. Kotovenko, R. Ranjan, V. Chandra, and N. K. Kalantari, "Coherentgs: Sparse novel view synthesis with coherent 3d gaussians," in *European Conference on Computer Vision*. Springer, 2024, pp. 19–37.

[40] 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. 20 775–20 785.

[41] X. Lei, X. Wang, L. Liu, H. Li, and H. Zhang, "Leop-gs: Learned optimizer with dynamic gradient update for sparse-view 3dgs," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[42] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[43] 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.

[44] 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.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[46] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[47] Q. Song *et al.*, "D2gs: Depth-and-density guided gaussian splatting for stable and accurate sparse-view reconstruction," *arXiv preprint arXiv:2510.08566*, 2025, accepted to ICLR 2026.

[48] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. 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 (CVPR)*, 2022, pp. 5480–5490.

[49] 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 (CVPR)*, 2023, pp. 8254–8263.



Feng Zhou received the B.S. degree in Internet of Things Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2022. He is currently pursuing the Ph.D. degree in Control Science and Engineering with the School of Intelligent Engineering and Automation, BUPT. His research interests include computer vision, 3D scene reconstruction and understanding, and controllable visual generation.



Wenkai Guo received the B.S. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently pursuing the M.S. degree in Artificial Intelligence at BUPT. His research interests include 3D reconstruction, computer vision, and deep learning.



Pu Cao received the B.S. degree from University of Science and Technology Beijing (USTB), Beijing, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include multimodal understanding and generation, multimodal large language models, and diffusion models.



Zhicheng Zhang received the B.S. degree in Information Engineering, the M.S. degree in Pattern Recognition, and the Ph.D. degree in Control Theory and Engineering from Jilin University, Changchun, China, in 2005, 2007, and 2011, respectively. He is currently an Associate Professor with the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include embodied intelligence, machine learning, and swarm intelligence.



Jianqin Yin received the Ph.D. degree from Shandong University, Jinan, China, in 2013. She is currently a Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include service robots, pattern recognition, machine learning, and image processing.