TWINDROPGS: RETHINKING DROPOUT REGULARIZATION FOR SPARSE-VIEW GAUSSIAN SPLATTING WITH TWIN CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

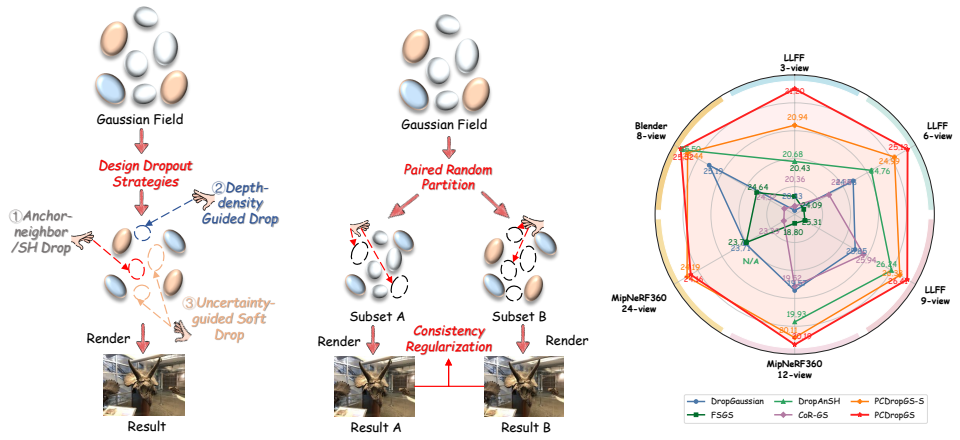
Sparse-view novel view synthesis is a fundamental yet challenging problem in 3D reconstruction, where limited observations often lead to severe overfitting and unreliable scene representations. Recent dropout-based 3D Gaussian Splatting methods alleviate this issue by randomly removing Gaussian primitives during training. However, these methods typically optimize only a single stochastic Gaussian subfield at each iteration, leaving the discrepancy across different dropout realizations unobserved and unconstrained. In this work, we rethink dropout regularization from a consistency-centric perspective and propose **TwinDropGS**, a **Twin Drop**out-induced Consistency Regularization framework for sparse-view Gaussian Splatting. To make dropout-induced stochastic discrepancy explicitly observable and regularizable, we first construct twin Gaussian subfields from a shared Gaussian field while ensuring complete joint Gaussian coverage. Building on it, we introduce an error-guided low-frequency consistency regularization that emphasizes regions with larger reconstruction errors and aligns their low-frequency luminance responses, avoiding excessive constraints on ambiguous high-frequency details. Additionally, a progressive consistency scheduling strategy gradually reduces subfield sharing and strengthens consistency regularization throughout training, enabling stable structure formation in early optimization and stronger stochastic regularization in later stages. Extensive experiments demonstrate that TwinDropGS consistently improves reconstruction quality and training stability over existing dropout-based 3DGS methods while retaining a simple and plug-and-play design. The source code will be released at `http://github.com`.

1 INTRODUCTION

Sparse-view novel view synthesis (NVS) aims to reconstruct 3D scenes from only a few input images and supports a wide range of applications in immersive media, robotics, embodied intelligence, and 3D content creation. Recently, 3D Gaussian Splatting (3DGS) has emerged as an efficient representation for high-quality novel view synthesis due to its fast optimization and real-time rendering. However, limited observations often cause the model to overfit the input views and generalize poorly to unseen viewpoints. Existing sparse-view methods alleviate this problem through geometry or depth-aware priors, frequency regularization, co-regularization, and self-ensembling Niemeyer et al. (2022); Yang et al. (2023); Wang et al. (2023); Li et al. (2024); Zhang et al. (2024); Zhao et al. (2025). Among them, dropout-based regularization has emerged as a particularly attractive direction because of its simplicity and effectiveness. DropGaussian Park et al. (2025) demonstrated that simply dropping Gaussian primitives at random during training can substantially alleviate sparse-view overfitting without additional priors, motivating a growing family of dropout-based methods Xu et al. (2025); Song et al. (2026); Li et al. (2026); Fang et al. (2026); Guo et al. (2025); Chen et al. (2025). Subsequent works further improve this paradigm by designing specialized strategies for selecting which Gaussians to drop and how to drop them.

However, a more fundamental issue in dropout-based Gaussian optimization remains insufficiently explored. The basic framework these methods follow is illustrated in Fig. 1(a). **First**, existing methods optimize only one randomly dropped Gaussian subfield, leaving the discrepancy across

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Existing methods (b) TwinDropGS (c) Radar chart

Figure 1: **Comparison of existing dropout-based methods and TwinDropGS.** (a) Existing dropout-based methods mainly improve sparse-view reconstruction by designing increasingly sophisticated dropout strategies within a single-branch framework. (b) In contrast, TwinDropGS constructs twin stochastic Gaussian subfields from a shared Gaussian field and explicitly regularizes their consistency for more stable and robust sparse-view reconstruction. (c) Comparisons with existing methods across different sparse-view settings.

different dropout realizations unobserved and unconstrained. This stochastic inconsistency is amplified by the gap between dropped subfields used during training and the complete Gaussian field used at inference. **Second**, uniform full-image consistency is undesirable under sparse-view supervision as high-frequency details are ambiguous and weakly constrained Yang et al. (2023); Nguyen et al. (2025), while reconstruction reliability is spatially non-uniform. Thus, consistency should determine both *what* to match and *where* to emphasize. **Third**, since early optimization favors sufficient sharing while later stages benefit from stronger regularization, consistency regularization should determine both *when* to regularize and *how strongly* to enforce it Park et al. (2025).

To address these issues, we rethink dropout-based sparse-view Gaussian Splatting from a consistency-centric perspective and propose **TwinDropGS**, a **Twin Drop**out-induced Consistency Regularization framework. As shown in Fig. 1(b), TwinDropGS constructs twin stochastic Gaussian subfields from a shared Gaussian field and explicitly regularizes their consistency for more stable and robust sparse-view reconstruction. **First**, TwinDropGS retains simple random dropout but reformulates the conventional single-branch optimization into a twin learning framework. Specifically, two stochastic Gaussian subfields are jointly constructed from the same Gaussian field and rendered under the same training view, making dropout-induced discrepancy explicitly observable and regularizable while ensuring complete joint Gaussian coverage. **Second**, we introduce an error-guided low-frequency consistency regularization, where low-frequency luminance responses determine *what* should remain consistent and a pixel-wise reconstruction error map determines *where* consistency should be emphasized, avoiding excessive constraints on ambiguous high-frequency details. **Third**, we introduce a progressive consistency scheduling strategy that gradually reduces subfield sharing and strengthens consistency regularization throughout training, transitioning from high sharing and weak consistency in early optimization to stronger stochastic diversity and consistency in later stages. The main contributions of our work are summarized as follows:

- We **rethink** dropout-based sparse-view Gaussian Splatting from a consistency-centric perspective and propose **TwinDropGS**. By constructing twin stochastic Gaussian subfields from a shared Gaussian field, TwinDropGS makes dropout-induced discrepancy explicitly observable and regularizable while ensuring complete joint Gaussian coverage.
- We design an **error-guided low-frequency consistency regularization**, where low-frequency responses determine *what* information should remain consistent and a pixel-wise reconstruction error map determines *where* consistency should be emphasized, avoiding excessive constraints on ambiguous high-frequency details.
- We introduce a **progressive consistency scheduling strategy** that gradually reduces subfield sharing and strengthens consistency regularization throughout training, enabling sta-

ble structure formation in early optimization and stronger stochastic regularization in later stages.

- Extensive experiments on multiple sparse-view benchmarks demonstrate that **TwinDropGS** consistently improves reconstruction quality and training stability over existing dropout-based 3DGS methods while retaining a simple and plug-and-play design, as illustrated in Fig. 1 (c).

2 Related Work

2.1 Novel View Synthesis with 3D Gaussian Splatting

Novel view synthesis (NVS) synthesizes photo-realistic images from unseen viewpoints given input observations. Neural Radiance Fields (NeRF) Mildenhall et al. (2020) represent scenes as continuous volumetric radiance fields, with subsequent works improving rendering quality, efficiency, and sparse-view generalization Barron et al. (2021); Yu et al. (2021); Wang et al. (2021); Chen et al. (2021); Guo et al. (2024); Xiao et al. (2026), but usually requiring expensive ray sampling and repeated network evaluation. 3D Gaussian Splatting (3DGS) Kerbl et al. (2023) instead represents scenes with explicit anisotropic Gaussian primitives and enables real-time rendering. Follow-up works improve anti-aliasing Yu et al. (2024), structured representations Lu et al. (2024), Gaussian propagation Cheng et al. (2024), rendering quality Jiang et al. (2024), and geometry-aware reconstruction Liu et al. (2024); Zhou et al. (2025). Despite these advances, most improvements are developed under sufficiently observed settings, while reconstruction from only a few input views remains substantially more challenging. However, sparse-view optimization remains under-constrained and prone to overfitting.

2.2 Sparse-view Novel View Synthesis

Sparse-view NVS is inherently ill-posed due to limited observations. Geometry-aware methods introduce depth, epipolar, or other geometric priors, including RegNeRF Niemeyer et al. (2022), SparseNeRF Wang et al. (2023), DNGaussian Li et al. (2024), DRGS Chung et al. (2024), SparseGS Xiong et al. (2023), NexusGS Zheng et al. (2025), and D^2GS Song et al. (2026). Regularization-based methods include frequency regularization Yang et al. (2023); Nguyen et al. (2025), co-regularization Zhang et al. (2024), and self-ensembling Zhao et al. (2025). Other methods improve Gaussian distribution or representation ability, such as FSGS Zhu et al. (2024), GaussianPro Cheng et al. (2024), ScaffoldGS Lu et al. (2024), MSA-Splatting Zhao et al. (2026), and generative sparse-view reconstruction Kong et al. (2025). These studies demonstrate that improving reconstruction reliability under limited observations generally requires stronger constraints on geometry, representation, or optimization. Many of these approaches rely on additional priors, auxiliary representations, or specialized optimization mechanisms.

2.3 Dropout-based Sparse-view Reconstruction

Dropout-based regularization offers a simple yet effective solution for sparse-view Gaussian Splatting. Motivated by its advantages, a number of methods have been proposed. DropoutGS Xu et al. (2025) introduces random Gaussian dropout, while DropGaussian Park et al. (2025) shows that randomly removing Gaussian primitives improves the visibility and gradients of the remaining ones. Subsequent methods design more specialized dropout strategies. D^2GS Song et al. (2026) introduces depth-and-density guided dropout, DropAnSH-GS Fang et al. (2026) combines anchor-neighbor and spherical-harmonic dropout, UGOD Guo et al. (2025) employs uncertainty-guided soft dropping, DOC-GS Li et al. (2026) explores reliability-oriented dropout and pruning, and Co-Adaptation Chen et al. (2025) analyzes Gaussian co-adaptation through random subset rendering. These advances demonstrate the effectiveness of stochastic Gaussian removal for mitigating overfitting under sparse-view supervision, but largely treat each dropped subfield as an independent training realization. However, these methods mainly focus on primitive-level dropout strategies, while stochastic inconsistency across different dropout realizations remains relatively underexplored. TwinDropGS instead revisits dropout-based sparse-view Gaussian Splatting from a consistency-centric perspective. It retains simple random dropout, constructs twin complementary Gaussian subfields, and explicitly regularizes their rendered responses under sparse-view supervision.

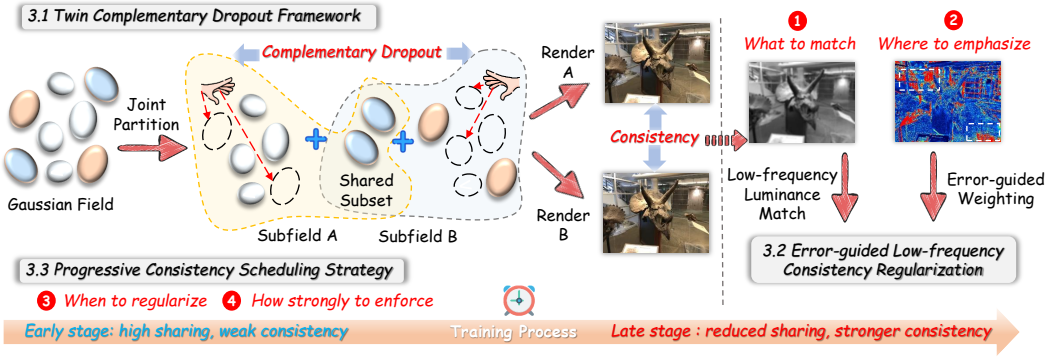


Figure 2: **Overview of TwinDropGS.** TwinDropGS first constructs twin complementary Gaussian subfields from a shared Gaussian field, making dropout-induced discrepancy explicitly observable while ensuring complete joint coverage. The error-guided low-frequency consistency regularization determines *what* to match and *where* to emphasize across the twin renderings. Progressive consistency scheduling further determines *when* to regularize and *how strongly* to enforce consistency, transitioning from high sharing and weak consistency in early training to reduced sharing and stronger consistency in later stages.

3 METHOD

TwinDropGS follows the standard SfM-initialized Gaussian Splatting pipeline Park et al. (2025); Song et al. (2026); Chen et al. (2025); Li et al. (2026) and revisits dropout-based sparse-view reconstruction from a **consistency-centric perspective**. As illustrated in Fig. 2, TwinDropGS first constructs twin stochastic Gaussian subfields to make dropout-induced discrepancy explicitly observable, then applies error-guided low-frequency consistency to determine *what* and *where* to regularize, and finally adopts progressive consistency scheduling to control *when* and *how strongly* the regularization is enforced. Together, these designs improve the stability of sparse-view Gaussian optimization under stochastic dropout.

3.1 TWIN COMPLEMENTARY DROPOUT FRAMEWORK

Existing dropout-based methods usually render only one randomly dropped Gaussian subset during each training iteration Park et al. (2025); Fang et al. (2026); Li et al. (2026). However, different dropout realizations from the same Gaussian field may produce inconsistent structural responses, while such discrepancy remains unobserved under single-branch optimization. To make this dropout-induced inconsistency explicitly observable, TwinDropGS constructs twin Gaussian subfields from the same Gaussian field under the same training view, providing the basis for subsequent consistency regularization.

A straightforward design is to independently sample two dropout masks. However, the two branches are not explicitly coordinated Yang et al. (2025), which may lead to redundant overlap or common omission of Gaussians and incomplete joint coverage. Consequently, we introduce a **complementary dropout** strategy. Specifically, at training iteration t with dropout ratio d_t, we randomly select $\mathrm{round}((1-2d_t)N_t)$ Gaussians from the current Gaussian field $\mathcal{G}_t = \{g_i\}_{i=1}^{N_t}$ as the shared subset $\mathcal{S}_t$, where N_t denotes the total number of Gaussians, and evenly partition the remaining Gaussians into two mutually exclusive subsets $\mathcal{U}_t^A$ and $\mathcal{U}_t^B$. The twin Gaussian subfields associated with branches A and B are constructed as:

$$\mathcal{G}_t^A = \mathcal{S}_t \cup \mathcal{U}_t^A, \qquad \mathcal{G}_t^B = \mathcal{S}_t \cup \mathcal{U}_t^B, \tag{1}$$

and they satisfy:

$$\mathcal{G}_t^A \cup \mathcal{G}_t^B = \mathcal{G}_t, \qquad \mathcal{G}_t^A \cap \mathcal{G}_t^B = \mathcal{S}_t, \tag{2}$$

where $\cup$ and $\cap$ denote set union and intersection, respectively. The two branches preserve stochastic diversity while jointly covering the complete Gaussian field. Given the same training view v_t, the corresponding rendered images are obtained based on the differentiable Gaussian rasterization $\mathcal{R}(\cdot)$:

$$I_t^A = \mathcal{R}(\mathcal{G}_t^A, v_t), \qquad I_t^B = \mathcal{R}(\mathcal{G}_t^B, v_t), \tag{3}$$

Both branches are supervised by the same ground-truth image I_t^{gt} using the standard 3DGS photometric loss $\mathcal{L}_{rgb}$:

$$\mathcal{L}_{rgb}(I, I^{gt}) = \mathcal{L}_{L_1}(I, I^{gt}) + \lambda_{d\text{-}ssim}\left(1 - \mathrm{SSIM}(I, I^{gt})\right), \tag{4}$$

where $\lambda_{d\text{-}ssim}$ balances L_1 and SSIM terms. The two-branch reconstruction loss$\mathcal{L}_{rec}^{twin}$ is written as:

$$\mathcal{L}_{rec}^{twin} = \frac{1}{2}\left[\mathcal{L}_{rgb}(I_t^A, I_t^{gt}) + \mathcal{L}_{rgb}(I_t^B, I_t^{gt})\right]. \tag{5}$$

This twin complementary formulation provides coordinated stochastic subfields with complete joint coverage, forming the basis for the subsequent consistency regularization.

3.2 Error-Guided Low-Frequency Consistency Regularization

Although the twin stochastic subfields make dropout-induced discrepancy observable, uniformly enforcing consistency over the entire rendered image may be suboptimal under sparse-view supervision. **Where to emphasize.** Reconstruction reliability is spatially non-uniform, and local image-space signals can effectively indicate unreliable regions that require stronger optimization Zhang et al. (2024); Ye et al. (2024). Therefore, consistency should place greater emphasis on regions with larger reconstruction errors rather than treating all pixels equally. **What to match.** Sparse-view supervision weakly constrains fine details, making high-frequency components more susceptible to overfitting and ambiguity Yang et al. (2023); Nguyen et al. (2025). In contrast, low-frequency components offer more reliable structural cues for sparse-view reconstruction Nguyen et al. (2025). Accordingly, we impose the consistency constraint on low-frequency responses instead of the entire rendered images. Specifically, we first estimate the reconstruction error of the twin branches using a pixel-wise error map $E_t(p)$:

$$E_t(p) = \frac{1}{2}\sum_{k\in\{A,B\}} \left\| I_t^k(p) - I_t^{gt}(p) \right\|_1, \tag{6}$$

where $k \in \{A, B\}$ indexes the twin branches, $I_t^k(p)$ denotes the rendered color of branch k at pixel p and iteration t, and $I_t^{gt}(p)$ denotes the corresponding ground-truth color. The resulting error map is then normalized to obtain a spatial importance weight:

$$W_t(p) = 1 + \beta\,\mathrm{Norm}(E_t(p)). \tag{7}$$

Here $W_t(p)$ denotes the consistency weight assigned to pixel p, $\mathrm{Norm}(\cdot)$ denotes the normalization operation applied to the error map, and β controls the strength of error-guided spatial weighting. Gradients are stopped through W_t , such that the error map only serves as a spatial importance estimator rather than an additional optimization target. We further constrain the low-frequency luminance responses of the twin branches, and define the error-guided low-frequency consistency loss $\mathcal{L}_{cons}$ as:

$$\mathcal{L}_{cons} = \frac{\sum_p W_t(p)\left|\Phi\big(Y(I_t^A)\big)(p) - \mathrm{sg}\left(\Phi\big(Y(I_t^B)\big)(p)\right)\right|}{\sum_p W_t(p)}, \tag{8}$$

where $Y(\cdot)$ denotes the luminance conversion operator, $\Phi(\cdot)$ denotes the low-pass filtering operator, $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation, and p indexes all image pixels. This design emphasizes regions with larger reconstruction errors while aligning only reliable low-frequency responses, improving structural consistency without over-constraining ambiguous high-frequency details.

3.3 Progressive Consistency Scheduling Strategy

While the previous section determines *what* to match and *where* to emphasize, consistency regularization should also adapt to the training stage. Inspired by progressive regularization Ni et al. (2024) and sparse-view overfitting observations Park et al. (2025); Song et al. (2026), we further consider *when* to strengthen regularization and *how strongly* to enforce it. **When to regularize.** Early training favors higher sharing for stable structure formation, whereas later stages benefit from stronger stochastic diversity to alleviate overfitting. **How strongly to enforce.** Consistency is kept weak in early training and progressively strengthened as the Gaussian representation stabilizes. Accordingly, we jointly adjust subfield sharing and consistency strength throughout training. Specifically, the dropout ratio is scheduled as:

$$d_t = d_{\max}\frac{t}{T}, \tag{9}$$

where d_t denotes the dropout ratio at iteration t, $d_{\max}$ denotes the maximum dropout ratio, and T denotes the total number of training iterations. As d_t increases, the shared Gaussian proportion decreases, making the twin subfields progressively more complementary. Meanwhile, the consistency strength is defined as:

$$\lambda_t = \lambda_{\max} \min\left(1, \frac{t}{T_{\text{warm}}}\right), \tag{10}$$

where λ_t denotes the consistency weight at iteration t, $\lambda_{\max}$ determines its maximum value, and T_{warm} denotes the warm-up period for consistency regularization. The overall training objective is defined as:

$$\mathcal{L}_t = \mathcal{L}_{rec}^{twin} + \lambda_t \mathcal{L}_{cons}. \tag{11}$$

Accordingly, TwinDropGS transitions from high sharing and weak consistency in early training to stronger stochastic diversity and consistency in later stages, balancing stable structure formation and effective regularization.

3.4 Discussion

TwinDropGS does not rely on external priors, auxiliary networks, or specialized Gaussian-specific dropout strategies. Instead, it revisits dropout-based sparse-view optimization from a consistency-centric perspective. The twin complementary dropout framework makes dropout-induced discrepancy explicitly observable while ensuring complete joint Gaussian coverage; the error-guided low-frequency consistency determines *what* to regularize and *where* to emphasize; and the progressive consistency scheduling further controls *when* and *how strongly* the regularization is enforced. Together, these designs improve the stability and robustness of sparse-view Gaussian optimization while preserving a simple and plug-and-play formulation.

4 Experiments

4.1 Implementation Details

Training Details. In this work, we propose TwinDropGS and its lightweight variant, namely TwinDropGS-S, for sparse-view Gaussian splatting. For TwinDropGS, we set $\lambda_{d\text{-}ssim} = 0.2$, $d_{\max} = 0.2$, $\lambda_{\max} = 0.05$, $T_{warm} = 7000$, and the error-guided weighting coefficient $\beta = 0.25$. The low-pass filtering operator is implemented using Gaussian blur with kernel size 11 and standard deviation $\sigma = 3.0$. Following previous works Park et al. (2025); Fang et al. (2026), the total number of iterations is 10000. TwinDropGS-S adopts a shortened training schedule to reduce the training cost introduced by twin branches. Its total number of iterations is 5000 and $T_{warm} = 4000$, while all other settings are unchanged. Our implementation is built upon DropGaussian Park et al. (2025), and all other Gaussian optimization settings follow this work unless otherwise specified. Following a previous sparse-view reconstruction method Fang et al. (2026), we report the average results over three independent runs. All experiments are conducted on a single NVIDIA A800 GPU.

Datasets and Metrics. Following previous sparse-view reconstruction methods Park et al. (2025); Fang et al. (2026); Wang et al. (2023), we evaluate our methods on LLFF Mildenhall et al. (2019), Mip-NeRF360 Barron et al. (2022), and Blender Mildenhall et al. (2020) datasets, with 3/6/9, 12/24, and 8 input views, respectively. Consistent with prior works Yang et al. (2023); Wang et al. (2023); Park et al. (2025), we employ the standard sparse-view data splits and downsampling settings, evaluate rendering quality using PSNR, SSIM, and LPIPS, and use training time as the efficiency metric.

State-of-the-art Methods. We compare our methods with some representative NeRF-based methods, including Mip-NeRF Barron et al. (2021), DietNeRF Jain et al. (2021), RegNeRF Niemeyer et al. (2022), FreeNeRF Yang et al. (2023), and SparseNeRF Wang et al. (2023), as well as some 3DGS-based methods, including vanilla 3DGS Kerbl et al. (2023), DNGaussian Li et al. (2024), FSGS Zhu et al. (2024), CoR-GS Zhang et al. (2024), and NexusGS Zheng et al. (2025). Since our methods are designed for dropout-based sparse-view reconstruction, we pay particular attention to comparisons with existing dropout-based methods, i.e., DropoutGS Xu et al. (2025), DropGaussian Park et al. (2025), and DropAnSH-GS Fang et al. (2026), to demonstrate the effectiveness.

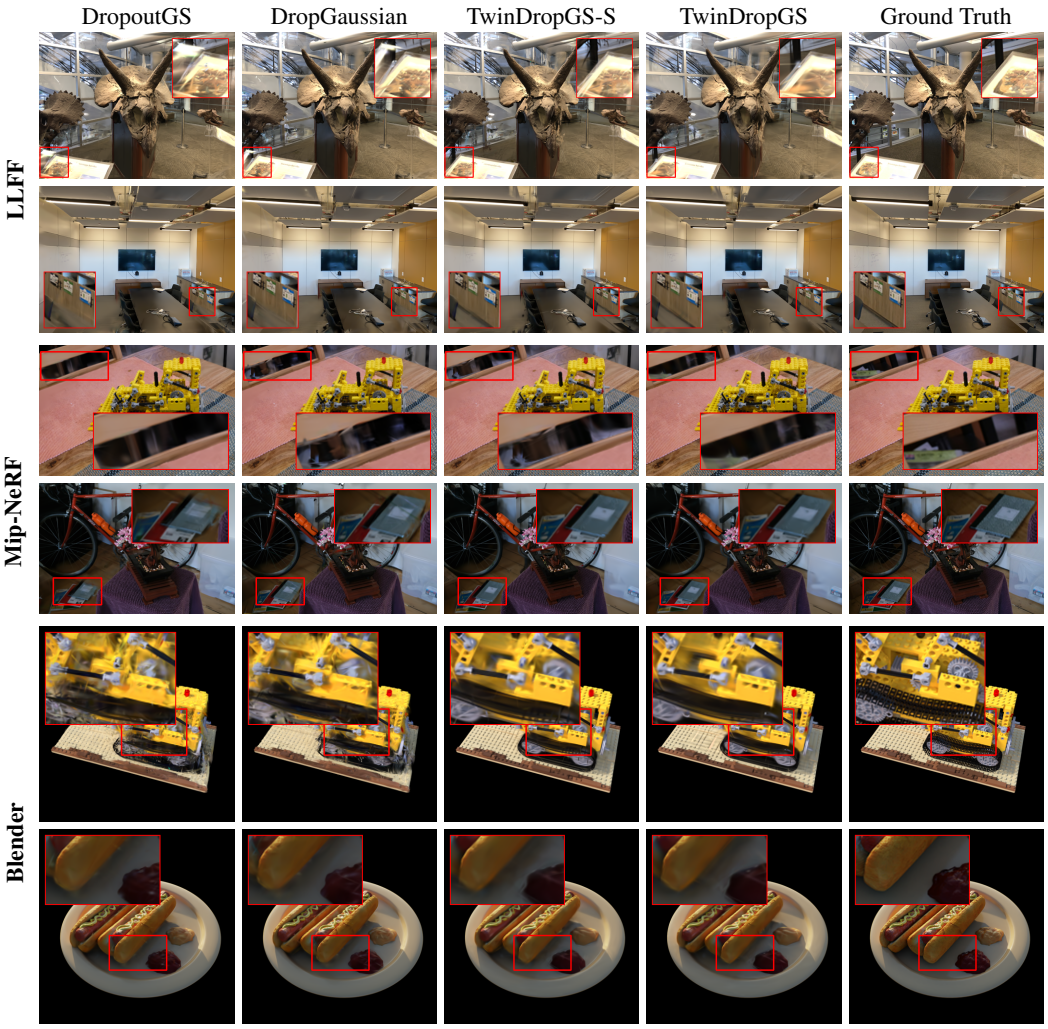


Figure 3: Qualitative comparison on the LLFF dataset under the 3-view setting, Mip-NeRF360 dataset under the 12-view setting, and Blender dataset under the 8-view setting.

4.2 Quantitative Comparison

We compare TwinDropGS and TwinDropGS-S with representative NeRF and 3DGS-based methods on LLFF Mildenhall et al. (2019), Mip-NeRF360 Barron et al. (2022), and Blender Mildenhall et al. (2020). As shown in Tab. 1 and Tab. 2, TwinDropGS consistently achieves strong performance across all sparse-view settings. In particular, on LLFF 3-view, TwinDropGS reaches 21.20 dB PSNR, outperforming 3DGS by 2.03 dB and DropAnSH-GS by 0.52 dB. Similar improvements are observed on LLFF 6 and 9-view, Mip-NeRF360, and Blender. TwinDropGS-S also remains competitive while reducing the additional training cost. These results validate the effectiveness of TwinDropGS, showing that consistency regularization across twin stochastic subfields together with progressive consistency scheduling improves reconstruction stability under sparse supervision.

4.3 Qualitative Comparison

As shown in Fig. 3, we compare TwinDropGS with representative dropout-based methods, including DropoutGS Xu et al. (2025), DropGaussian Park et al. (2025), and DropAnSH-GS Fang et al. (2026). Regions with noticeable differences are highlighted by red boxes. Existing methods may still produce blurred boundaries or local structural inconsistencies, while TwinDropGS explicitly regularizes twin stochastic subfields through error-guided low-frequency consistency. As a result,

Table 1: Quantitative comparison on the LLFF dataset under sparse-view settings. Colors indicate best, second best, and third best results.

	Methods	LLFF (3-view)			LLFF (6-view)			LLFF (9-view)		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-based	Mip-NeRF Barron et al. (2021)	16.11	0.401	0.460	22.91	0.756	0.213	24.88	0.826	0.170
	DietNeRF Jain et al. (2021)	14.94	0.370	0.496	21.75	0.717	0.248	24.28	0.801	0.183
	RegNeRF Niemeyer et al. (2022)	19.08	0.587	0.336	23.10	0.760	0.206	24.86	0.820	0.161
	FreeNeRF Yang et al. (2023)	19.63	0.612	0.308	23.73	0.779	0.195	25.13	0.827	0.160
	SparseNeRF Wang et al. (2023)	19.86	0.624	0.328	–	–	–	–	–	–
3DGS-based	3DGS Kerbl et al. (2023)	19.17	0.646	0.268	23.74	0.807	0.162	25.44	0.860	0.096
	DNGaussian Li et al. (2024)	19.12	0.591	0.294	22.18	0.755	0.198	23.17	0.788	0.180
	FSGS Zhu et al. (2024)	20.43	0.682	0.248	24.09	0.823	0.145	25.31	0.860	0.122
	CoR-GS Zhang et al. (2024)	20.36	0.710	0.202	24.34	0.831	0.122	25.94	0.872	0.088
	DropoutGS Xu et al. (2025)	19.39	0.632	0.279	24.02	0.816	0.144	25.13	0.869	0.099
	DropGaussian Park et al. (2025)	20.33	0.709	0.201	24.58	0.830	0.125	25.85	0.864	0.093
	DropAnSH-GS Fang et al. (2026)	20.68	0.724	0.194	24.76	0.837	0.116	26.24	0.875	0.088
	TwinDropGS-S (Ours)	20.94	0.747	0.188	24.99	0.841	0.115	26.33	0.877	0.090
	TwinDropGS (Ours)	21.20	0.758	0.180	25.12	0.848	0.114	26.41	0.882	0.085

Table 2: Quantitative comparison on the Mip-NeRF360 and Blender datasets under sparse-view settings. Colors indicate best, second best, and third best results.

Methods	Mip-NeRF360 (12-view)			Mip-NeRF360 (24-view)			Blender (8-view)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS Kerbl et al. (2023)	18.52	0.523	0.415	22.80	0.708	0.276	21.56	0.847	0.130
FSGS Zhu et al. (2024)	18.80	0.531	0.418	23.70	0.745	0.230	24.64	0.895	0.095
CoR-GS Zhang et al. (2024)	19.52	0.558	0.418	23.39	0.727	0.271	24.32	0.892	0.084
DropGaussian Park et al. (2025)	19.57	0.562	0.360	23.71	0.752	0.227	25.19	0.890	0.090
NexusGS Zheng et al. (2025)	–	–	–	23.86	0.753	0.206	24.37	0.893	0.087
DropAnSH-GS Fang et al. (2026)	19.93	0.576	0.362	–	–	–	25.50	0.891	0.088
TwinDropGS-S (Ours)	20.11	0.584	0.363	24.19	0.760	0.213	25.44	0.893	0.091
TwinDropGS (Ours)	20.19	0.592	0.360	24.26	0.767	0.211	25.52	0.896	0.086

TwinDropGS and TwinDropGS-S recover sharper boundaries and more coherent local structures, yielding better visual quality.

4.4 Compatibility

To further verify the generality of TwinDropGS, we integrate it into FSGS Zhu et al. (2024), CoR-GS Zhang et al. (2024), and DropGaussian Park et al. (2025) while keeping their original settings unchanged. For non-dropout baselines, a lightweight Gaussian dropout is added to construct twin subfields. `Ours` uses the full TwinDropGS schedule, while `Ours-S` adopts the shortened schedule for comparable training cost against baselines. As shown in Tab. 3, TwinDropGS consistently improves all three baselines, while `Ours-S` retains clear gains with little additional training cost. These results demonstrate that TwinDropGS can serve as a simple plug-and-play regularizer for different 3DGS-based sparse-view frameworks.

4.5 Ablation Studies

We conduct ablation studies on LLFF under the 3-view setting using DropGaussian Park et al. (2025) as the baseline. All variants follow the same training protocol and evaluation setting to isolate the contribution of each proposed component.

Effect of the Proposed Components. As shown in Tab. 4, introducing the twin framework (Two-B.) improves PSNR from 20.34 to 20.79 dB. Adding error-guided low-frequency consistency (EG-LF Cons.) further increases it to 21.04 dB, while progressive scheduling (Prog. S.) yields the best overall performance of 21.20 dB PSNR, 0.758 SSIM, and 0.180 LPIPS. These results validate the effectiveness of all three components. The progressive improvements across successive variants

Table 3: Compatibility of our method with existing representative sparse-view reconstruction methods on the LLFF dataset under the 3-view setting.

Method	PSNR↑	SSIM↑	LPIPS↓	Avg. Time↓
FSGS	20.43	0.682	0.248	276 s
FSGS + *Ours*	**20.77**	**0.739**	**0.217**	422 s
FSGS + *Ours-S*	20.69	0.728	0.223	**262 s**
CoR-GS	20.36	0.710	0.202	**373 s**
CoR-GS + *Ours*	**20.82**	**0.751**	**0.198**	532 s
CoR-GS + *Ours-S*	20.71	0.744	0.200	378 s
DropGaussian	20.33	0.709	0.201	**74 s**
DropGaussian + *Ours*	**21.21**	**0.757**	**0.180**	151 s
DropGaussian + *Ours-S*	20.99	0.744	0.192	**74 s**

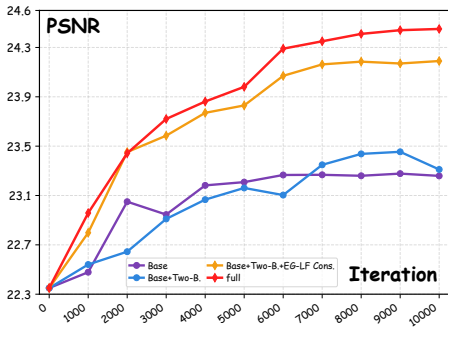


Figure 4: Training PSNR curves of different ablation variants on the challenging LLFF 3-view *fortress* scene.

Table 4: Ablation study of the main components on the LLFF dataset under the 3-view setting.

Two-B.	EG-LF Cons.	Prog. S.	PSNR↑	SSIM↑	LPIPS↓	Avg. Time↓
×	×	×	20.34	0.711	0.201	**74 s**
✓	×	×	20.79	0.732	0.200	128 s
✓	✓	×	21.04	0.744	0.186	152 s
✓	✓	✓	**21.20**	**0.758**	**0.180**	151 s

Table 5: Effect of numbers of branches on the LLFF dataset under the 3-view setting.

Model	Branch	PSNR↑	SSIM↑	LPIPS↓	Avg. Time↓
TwinDropGS-S	1	20.26	0.687	0.218	**35 s**
	2	20.99	0.744	0.192	74 s
	3	**21.00**	**0.748**	**0.191**	113 s
TwinDropGS	1	20.36	0.711	0.202	**74 s**
	2	21.21	0.757	0.180	151 s
	3	**21.29**	**0.758**	0.181	230 s

also indicate that the proposed components provide complementary benefits rather than relying on a single dominant factor. Figure 4 further compares the training PSNR curves of different ablation variants on the challenging LLFF 3-view *fortress* scene. The twin framework improves the optimization trajectory over the single-branch baseline, but the performance gain does not solely come from introducing an additional branch. When error-guided low-frequency consistency is imposed across the twin branches, the reconstruction quality is further improved, indicating that explicitly regularizing their stochastic discrepancy is essential for fully exploiting the twin formulation. With progressive scheduling, the full model maintains continued performance gains and achieves the highest and most stable PSNR in the later training stage, demonstrating that consistency regularization together with stage-adaptive scheduling effectively improves both reconstruction quality and optimization stability. These observations are consistent with the quantitative results in Tab. 4.

Effect of the Number of Branches. We further study the trade-off between reconstruction quality and training cost with different numbers of branches. Besides the single-branch baseline and our two-branch framework, we additionally implement a three-branch extension using the same training protocol for comparison. Tab. 5 compares different numbers of stochastic branches. Moving from one to two branches brings substantial gains for both TwinDropGS and TwinDropGS-S, whereas a third branch provides only marginal improvement with considerably higher training cost. This indicates that two complementary branches already provide sufficient stochastic diversity for effective consistency regularization. We therefore adopt two branches as the default setting.

5 Conclusion

In this paper, we rethink dropout regularization from a consistency-centric perspective and propose **TwinDropGS**, a **Twin Drop**out-induced Consistency Regularization framework for sparse-view 3D Gaussian Splatting. By explicitly regularizing the stochastic discrepancy induced by single-branch dropout, TwinDropGS improves sparse-view optimization through twin stochastic subfields, error-guided low-frequency consistency, and progressive consistency scheduling. Extensive experiments demonstrate that our method achieves consistent gains in reconstruction quality and training stability across multiple benchmarks while retaining a simple and plug-and-play design. These results highlight the importance of consistency regularization for dropout-based sparse-view Gaussian Splatting.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024.

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

Kangjie Chen, Yingji Zhong, Zhihao Li, Jiaqi Lin, Youyu Chen, Minghan Qin, and Haoqian Wang. Quantifying and alleviating co-adaptation in sparse-view 3D gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 38, pp. 128537–128566, 2025.

Tianyu Chen, Wei Xiang, Kang Han, Yu Lu, Di Wu, Gaowen Liu, and Ramana Rao Kompella. GIF-Splat: Generative prior-guided iterative feed-forward 3D gaussian splatting from sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26031–26040, 2026.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386, 2024.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. GaussianPro: 3D gaussian splatting with progressive propagation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 8123–8140, 2024.

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

Shuangkang Fang, I-Chao Shen, Xuanyang Zhang, Zesheng Wang, Yufeng Wang, Wenrui Ding, Gang Yu, and Takeo Igarashi. Dropping anchor and spherical harmonics for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33312–33322, 2026.

Shuai Guo, Qiuwen Wang, Yijie Gao, Rong Xie, and Li Song. Depth-guided robust and fast point cloud fusion NeRF for sparse input views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 1976–1984, 2024.

Zhihao Guo, Peng Wang, Zidong Chen, Xiangyu Kong, Yan Lyu, Guanyu Gao, and Liangxiu Han. UGOD: Uncertainty-guided differentiable opacity and soft dropout for enhanced sparse-view 3DGS. arXiv:2508.04968, 2025.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5885–5894, 2021.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5322–5332, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

Hanyang Kong, Xingyi Yang, and Xinchao Wang. Generative sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26745–26755, 2025.

Hantang Li, Qiang Zhu, Xiandong Meng, Debin Zhao, and Xiaopeng Fan. DOC-GS: Dual-domain observation and calibration for reliable sparse-view gaussian splatting. arXiv:2604.06739, 2026.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. DNGaussian: Optimizing sparse-view 3D gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20775–20785, 2024.

Zhaoliang Liu, Jinhe Su, Guorong Cai, Yidong Chen, Binghui Zeng, and Zongyue Wang. GeoRGS: Geometric regularization for real-time novel view synthesis from sparse inputs. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(12):13113–13126, 2024.

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

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 38(4):29:1–29:14, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421, 2020.

Hung Nguyen, Runfa Li, An Le, and Truong Q. Nguyen. DWTGS: Rethinking frequency regularization for sparse-view 3D gaussian splatting. In *IEEE International Conference on Visual Communications and Image Processing*, pp. 1–5, 2025.

Shiwen Ni, Min Yang, Ruifeng Xu, Chengming Li, and Xiping Hu. Layer-wise regularized dropout for neural language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pp. 10208–10218, 2024.

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

Hyunwoo Park, Gun Ryu, and Wonjun Kim. DropGaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21600–21609, 2025.

Meixi Song, Xin Lin, Dizhe Zhang, Haodong Li, Xiangtai Li, Bo Du, and Lu Qi. D^2gs: Depth-and-density guided gaussian splatting for stable and accurate sparse-view reconstruction. In *International Conference on Learning Representations*, 2026.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. SparseNeRF: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9065–9076, 2023.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2021.

Yuru Xiao, Xianming Liu, Deming Zhai, Kui Jiang, Junjun Jiang, and Xiangyang Ji. SGCNeRF: Few-shot neural rendering via sparse geometric consistency guidance. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(3):3862–3873, 2026.

Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. SparseGS: Real-time 360° sparse view synthesis using gaussian splatting. arXiv:2312.00206, 2023.

Yexing Xu, Longguang Wang, Minglin Chen, Sheng Ao, Li Li, and Yulan Guo. DropoutGS: Dropping out gaussians for better sparse-view rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 701–710, 2025.

Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8254–8263, 2023.

Lihe Yang, Zhen Zhao, and Hengshuang Zhao. UniMatch V2: Pushing the limit of semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 3031–3048, 2025.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. AbsGS: Recovering fine details in 3D gaussian splatting. In *Proceedings of the ACM International Conference on Multimedia*, pp. 1053–1061, 2024.

Youngho Yoon and Kuk-Jin Yoon. ReSplat: Degradation-agnostic feed-forward gaussian splatting via self-guided residual diffusion. In *International Conference on Learning Representations*, 2026.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456, 2024.

Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. CoR-GS: Sparse-view 3D gaussian splatting via co-regularization. In *European Conference on Computer Vision*, pp. 335–352, 2024.

Chen Zhao, Xuan Wang, Tong Zhang, Saqib Javed, and Mathieu Salzmann. Self-ensembling gaussian splatting for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4940–4950, 2025.

Yaoyong Zhao, Guoqing Chen, Boying Wu, Qiyu Jin, and Tieyong Zeng. MSA-Splatting: Multi-scale adaptive gaussian splatting for high-fidelity view synthesis. *IEEE Transactions on Multimedia*, 28:5357–5367, 2026.

Yulong Zheng, Zicheng Jiang, Shengfeng He, Yandu Sun, Junyu Dong, Huaidong Zhang, and Yong Du. NexusGS: Sparse view synthesis with epipolar depth priors in 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26800–26809, 2025.

Ting Zhou, Siyuan Chen, Siyao Wan, Hanyun Lv, Zheng Luo, and Jianhui Wu. GEDR: Gaussian-enhanced detail reconstruction for real-time high-fidelity 3D scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(12):12265–12277, 2025.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pp. 145–163, 2024.

APPENDIX

A RELATED WORK ON FEED-FORWARD SPARSE-VIEW RECONSTRUCTION

Feed-forward sparse-view reconstruction has recently emerged as an efficient alternative to per-scene optimization. Instead of optimizing a scene-specific representation from scratch, these methods learn cross-scene priors from large-scale multi-view data and directly infer 3D representations from a small number of input images. Representative methods such as pixelSplat Charatan et al. (2024) predict pixel-aligned Gaussian primitives from image features, while MVSplat Chen et al. (2024) incorporates multi-view stereo cues to construct geometry-aware Gaussian representations. Subsequent works further improve cross-view feature aggregation, geometric reasoning, and Gaussian prediction, including GIFSplat Chen et al. (2026) and ReSplat Yoon & Yoon (2026). These feed-forward approaches substantially reduce scene-specific optimization cost and enable fast reconstruction and rendering for previously unseen scenes. Despite these advantages, feed-forward methods mainly target cross-scene generalization and efficient inference, and their performance largely depends on the learned priors and the distribution of the training data. In contrast, TwinDropGS follows the per-scene optimization paradigm and focuses on improving the robustness of sparse-view Gaussian optimization without relying on a pre-trained feed-forward reconstruction network. Specifically, TwinDropGS explicitly models and regularizes the stochastic inconsistency induced by Gaussian dropout through twin subfields, error-guided low-frequency consistency, and progressive consistency scheduling. Therefore, feed-forward reconstruction and TwinDropGS address complementary settings: the former emphasizes direct generalization and fast inference across scenes, whereas the latter improves the reliability of scene-specific optimization under extremely sparse observations.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 MORE QUALITATIVE COMPARISON

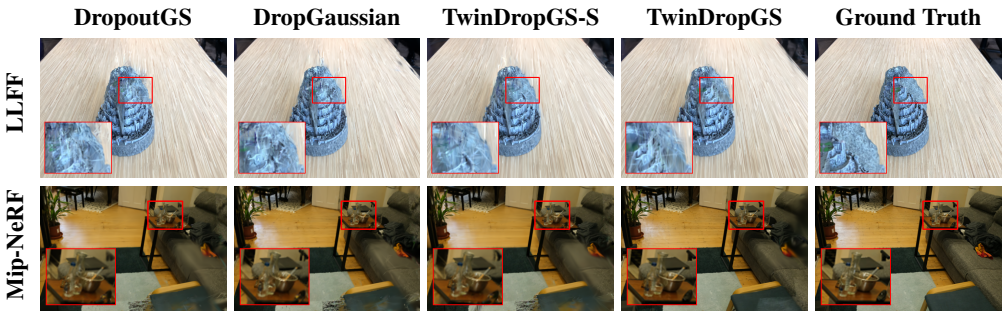


Figure 5: Qualitative comparison on the LLFF *fortress* scene under the 3-view setting and the Mip-NeRF360 *room* scene under the 12-view setting.

We further provide additional qualitative results on LLFF 3-view and Mip-NeRF360 12-view settings in Figure 5. Compared with DropoutGS and DropGaussian, TwinDropGS-S and TwinDropGS recover more complete structures and cleaner details with fewer artifacts. These results further demonstrate the effectiveness and robustness of the proposed consistency regularization across different sparse-view scenes.

B.2 STABILITY ANALYSIS

To evaluate robustness to training randomness against other dropout-based methods, we compare DropoutGS Xu et al. (2025), DropGaussian Park et al. (2025), and TwinDropGS under multiple random seeds on LLFF, Mip-NeRF360, and Blender datasets. Following prior multi-seed evaluation

Table 6: Stability analysis on the LLFF dataset under the 3-view setting. We report mean ± standard deviation over multiple random seeds.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓
Fern	DropoutGS Xu et al. (2025)	21.55±0.186	0.647±0.0088	0.285±0.0097
	DropGaussian Park et al. (2025)	22.42±0.254	0.724±0.0094	0.202±0.0081
	TwinDropGS (Ours)	**23.38±0.123**	**0.787±0.0055**	**0.182±0.0042**
Flower	DropoutGS Xu et al. (2025)	19.20±0.273	0.571±0.0106	0.308±0.0103
	DropGaussian Park et al. (2025)	20.02±0.399	0.654±0.0112	0.221±0.0098
	TwinDropGS (Ours)	**20.77±0.089**	**0.694±0.0062**	**0.200±0.0062**
Fortress	DropoutGS Xu et al. (2025)	22.50±0.427	0.684±0.0126	0.261±0.0090
	DropGaussian Park et al. (2025)	23.38±0.152	0.756±0.0091	0.182±0.0081
	TwinDropGS (Ours)	**24.33±0.156**	**0.814±0.0054**	**0.168±0.0028**
Horns	DropoutGS Xu et al. (2025)	17.97±0.326	0.599±0.0105	0.334±0.0103
	DropGaussian Park et al. (2025)	18.92±0.266	0.670±0.0098	0.249±0.0102
	TwinDropGS (Ours)	**19.86±0.102**	**0.723±0.0077**	**0.216±0.0052**
Leaves	DropoutGS Xu et al. (2025)	16.30±0.242	0.581±0.0091	0.290±0.0097
	DropGaussian Park et al. (2025)	17.18±0.266	0.660±0.0094	0.209±0.0094
	TwinDropGS (Ours)	**17.95±0.168**	**0.718±0.0064**	**0.190±0.0049**
Orchids	DropoutGS Xu et al. (2025)	14.90±0.395	0.481±0.0141	0.316±0.0115
	DropGaussian Park et al. (2025)	15.52±0.286	0.558±0.0140	0.232±0.0103
	TwinDropGS (Ours)	**16.54±0.137**	**0.599±0.0067**	**0.206±0.0058**
Room	DropoutGS Xu et al. (2025)	21.13±0.262	0.777±0.0146	0.234±0.0094
	DropGaussian Park et al. (2025)	22.17±0.325	0.840±0.0110	0.169±0.0091
	TwinDropGS (Ours)	**23.26±0.094**	**0.875±0.0047**	**0.138±0.0041**
Trex	DropoutGS Xu et al. (2025)	21.33±0.254	0.679±0.0120	0.213±0.0082
	DropGaussian Park et al. (2025)	**23.43±0.161**	0.812±0.0105	0.156±0.0083
	TwinDropGS (Ours)	23.37±0.145	**0.852±0.0056**	**0.138±0.0041**
Average	DropoutGS Xu et al. (2025)	19.36±0.1965	0.626±0.0029	0.279±0.0023
	DropGaussian Park et al. (2025)	20.38±0.1724	0.710±0.0024	0.200±0.0022
	TwinDropGS (Ours)	**21.20±0.0050**	**0.758±0.0013**	**0.180±0.0011**

protocols Song et al. (2026); Chen et al. (2025), all methods are trained with identical settings except for the random seed, and we report mean ± standard deviation.

As shown in Tab. 6, Tab. 7, Tab. 8, and Fig. 6, TwinDropGS consistently achieves higher average reconstruction quality with lower variance. On LLFF, it improves the average PSNR from 20.38 ± 0.1724 to 21.20 ± 0.0050 compared with DropGaussian, while also improving SSIM and LPIPS. Similar trends are observed on Mip-NeRF360 (20.19 ± 0.071 vs. 19.38 ± 0.114 PSNR) and Blender (25.52 ± 0.025 vs. 25.18 ± 0.116), demonstrating that TwinDropGS is less sensitive to dropout-induced randomness across different sparse-view settings.

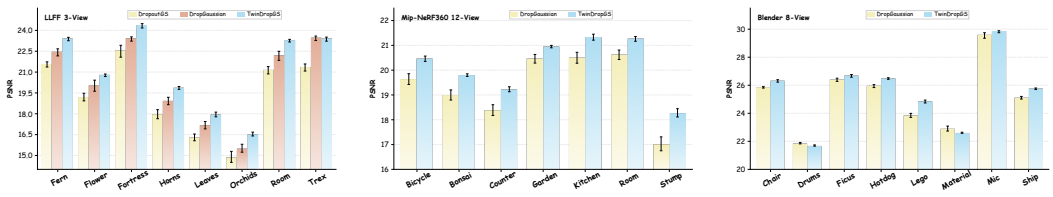


Figure 6: Per-scene PSNR fluctuation on three datasets under different sparse-view settings. From left to right: LLFF 3-view, Mip-NeRF360 12-view, and Blender 8-view.

Table 7: Stability analysis on the Mip-NeRF360 dataset under the 12-view setting. We report mean ± standard deviation over multiple random seeds.

Scene	Method	PSNR↑
Bicycle	DropGaussian Park et al. (2025)	19.64±0.217
	TwinDropGS (Ours)	**20.46±0.108**
Bonsai	DropGaussian Park et al. (2025)	19.00±0.204
	TwinDropGS (Ours)	**19.80±0.052**
Counter	DropGaussian Park et al. (2025)	18.39±0.217
	TwinDropGS (Ours)	**19.23±0.100**
Garden	DropGaussian Park et al. (2025)	20.46±0.174
	TwinDropGS (Ours)	**20.95±0.049**
Kitchen	DropGaussian Park et al. (2025)	20.50±0.220
	TwinDropGS (Ours)	**21.33±0.120**
Room	DropGaussian Park et al. (2025)	20.62±0.191
	TwinDropGS (Ours)	**21.26±0.095**
Stump	DropGaussian Park et al. (2025)	17.03±0.280
	TwinDropGS (Ours)	**18.28±0.167**
Average	DropGaussian Park et al. (2025)	19.38±0.114
	TwinDropGS (Ours)	**20.19±0.071**

Table 8: Stability analysis on the Blender dataset under the 8-view setting. We report mean ± standard deviation over multiple random seeds.

Scene	Method	PSNR↑
Chair	DropGaussian Park et al. (2025)	25.85±0.051
	TwinDropGS (Ours)	**26.31±0.087**
Drums	**DropGaussian Park et al. (2025)**	**21.87±0.050**
	TwinDropGS (Ours)	21.70±0.043
Ficus	DropGaussian Park et al. (2025)	26.39±0.112
	TwinDropGS (Ours)	**26.67±0.095**
Hotdog	DropGaussian Park et al. (2025)	25.94±0.105
	TwinDropGS (Ours)	**26.48±0.057**
Lego	DropGaussian Park et al. (2025)	23.85±0.143
	TwinDropGS (Ours)	**24.84±0.104**
Material	**DropGaussian Park et al. (2025)**	**22.91±0.177**
	TwinDropGS (Ours)	22.61±0.038
Mic	DropGaussian Park et al. (2025)	29.55±0.196
	TwinDropGS (Ours)	**29.82±0.072**
Ship	DropGaussian Park et al. (2025)	25.11±0.094
	TwinDropGS (Ours)	**25.75±0.055**
Average	DropGaussian Park et al. (2025)	25.18±0.116
	TwinDropGS (Ours)	**25.52±0.025**

C DISCUSSIONS

Although TwinDropGS achieves consistent gains with a simple and plug-and-play design, the twin branches introduce additional computational overhead. Future work will explore more efficient consistency regularization and extend this perspective to feed-forward and dynamic Gaussian reconstruction. We hope this consistency-centric view can inspire further research on explicitly coordinating and regularizing stochastic representations in Gaussian Splatting.

AI USE STATEMENT

We used large language models only for language polishing and improving the clarity of presentation. All research ideas, methodological designs, experiments, analyses, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of paper.