

# AugGS: Self-augmented Gaussians with Structural Masks for Sparse-view 3D Reconstruction

Bi'an Du, Lingbei Meng, Wei Hu\*

Wangxuan Institute of Computer Technology, Peking University, Beijing, China  
{pkudba, 2100017808}@stu.pku.edu.cn, forhuwei@pku.edu.cn

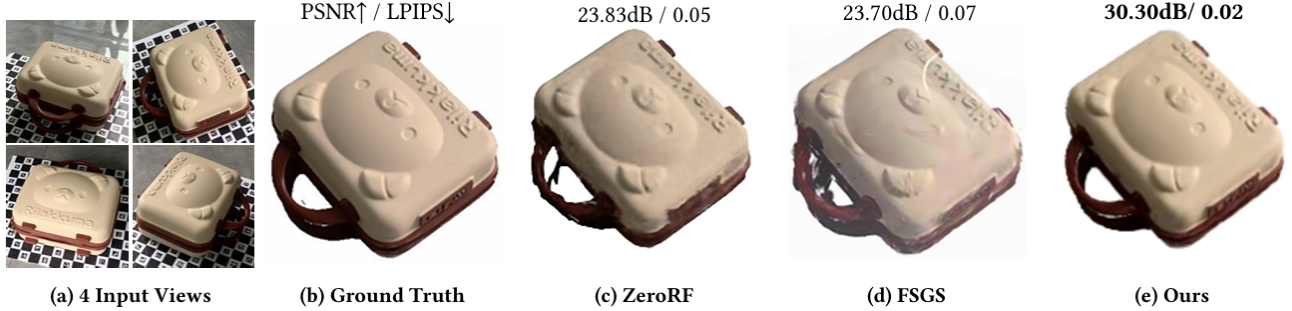


Fig. 1: Our method enables high-quality 3D reconstruction of sparse-view scenes with self-augmented Gaussian splatting, surpassing the current SOTA methods in both qualitative and quantitative aspects for sparse view 3D reconstructions.

**Abstract**—Sparse-view 3D reconstruction is a major challenge in computer vision, aiming to create complete three-dimensional models from limited viewing angles. Key obstacles include: 1) a small number of input images with inconsistent information; 2) dependence on input image quality; and 3) large model parameter sizes. To tackle these issues, we propose a self-augmented two-stage Gaussian splatting framework enhanced with structural masks for sparse-view 3D reconstruction. Initially, our method generates a basic 3D Gaussian representation from sparse inputs and renders multi-view images. We then fine-tune a pre-trained 2D diffusion model to enhance these images, using them as augmented data to further optimize the 3D Gaussians. Additionally, a structural masking strategy during training enhances the model’s robustness to sparse inputs and noise. Experiments on benchmarks like MipNeRF360, OmniObject3D, and OpenIllumination demonstrate that our approach achieves state-of-the-art performance in perceptual quality and multi-view consistency with sparse inputs.

**Index Terms**—Gaussian Splatting, 3D Reconstruction, Sparse-view Reconstruction

## I. INTRODUCTION

Reconstructing and rendering three-dimensional objects from two-dimensional images is a fundamental problem in the realm of computer vision. This intricate process involves translating sparse visual data into detailed and comprehensive 3D models, enabling applications such as augmented reality, robotics and 3D game/movie asset creation. Despite the potential, this task requires not only understanding complex geometries, but also typically capturing dozens of multi-view

images, which is cumbersome and costly. Hence, it is desirable to efficiently reconstruct high-quality 3D objects from highly sparse captured images.

Previous methods primarily aim to reduce dependence on dense image captures by using auxiliary consistency loss to prevent degenerate reconstructions [1]–[5]. However, significant challenges remain with extremely sparse views, such as having only four images covering 360°. The limited input severely restricts detailed representation, often resulting in low-fidelity reconstructions and poor surface textures due to insufficient data consistency. Additionally, existing models heavily rely on the quality and number of input images, making them vulnerable to variations in object complexity and environmental noise, which highlights a robustness gap. Furthermore, the large parameter sizes of these models increase training burdens and reduce inference efficiency, thus limiting their practicality and scalability.

To address these challenges, we propose a novel two-stage Gaussian splatting method enhanced with perceptual data augmentation and structural masks for consistent 3D geometry and appearance reconstruction from sparse views. In the first stage, a Gaussian model generates a basic 3D representation from limited inputs, serving as the initialization for the subsequent refinement. However, sparse views may lead to incomplete or blurry details in unobserved areas. To overcome this, we fine-tune a pre-trained image-conditioned ControlNet [6] to enhance multi-view images. By leveraging the cross-pixel consistency of a pre-trained 2D diffusion model

\* Corresponding author.

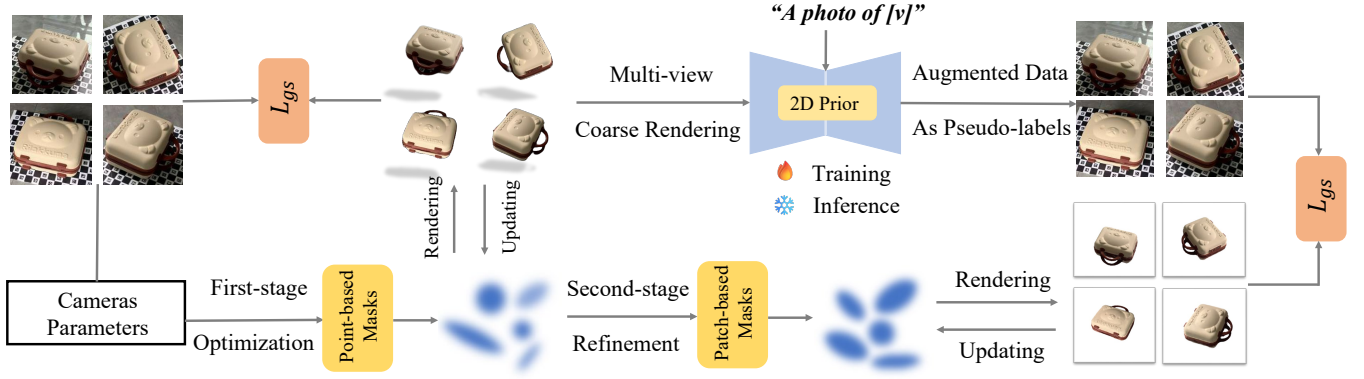


Fig. 2: The overall architecture of our self-augmented Gaussian splatting method. We first create a coarse 3D Gaussian model from sparse-view images, generating a coarse point cloud and renderings from novel views. Multi-view renders and the 2D prior enhance perceptual quality, with structural masks integrated into the two-stage Gaussian process.

during denoising and detail restoration, we produce more reliable augmented data for the subsequent 3D Gaussian training, minimizing discrepancies between views. Additionally, we introduce structure-aware mask—point-based for the coarse Gaussian model and patch-based for the fine model to further improve robustness against sparse inputs and noise. This integrated approach streamlines the reconstruction process and enhances the quality and reliability of neural rendering in real-world scenarios.

Building on previous works [3]–[5], [7], we trained and evaluated our method on the MipNeRF360 [8], OmniObject3D [9], and OpenIllumination [10] datasets. Our approach achieves state-of-the-art performance, outperforming the leading competitor [7] by 5–30% in LPIPS, 8–25% in PSNR, and 5–10% in SSIM across various view counts and datasets. Additionally, our solution without 2D priors not only surpasses the top competitor in performance but also enhances the training efficiency by 20× and inference efficiency by 10%.

Our contributions are summarized as follows:

- We introduce a two-stage Gaussian splatting method that generates pseudo-labels and utilizes 2D priors for perceptual data augmentation, enabling high-quality reconstruction of undersampled regions from sparse views.
- To address occlusions and unobserved areas, we implement point-based masks and patch-based masks for the two-stage Gaussians, respectively, which improve the robustness of the model against sparse input and noise.
- Experiments on benchmark datasets show that our method outperforms existing state-of-the-art methods in both qualitative and quantitative metrics. Additionally, we significantly improve training and inference efficiency without compromising the reconstruction quality.

## II. RELATED WORKS

### A. Differentiable Point-based Rendering

Recent advances in differentiable point-based rendering have revolutionized scene reconstruction. SynSin [11] enhanced point clouds with additional features and used CNNs

to generate images, addressing issues like incompleteness and noise. Point-NeRF [12] improved NeRF by rendering discrete neural points, reducing the computational cost of volumetric methods, though still requiring seconds for rendering. Point-Radiance [13] introduced spherical harmonics and a novel splatting strategy for real-time rendering, offering better color interpolation and higher visual quality, especially in complex lighting.

Building on these advances, 3D Gaussian Splatting (3DGS) [14] uses 3D Gaussian models for high-quality, rapid scene reconstructions. However, point-based methods still require dense point configurations, making sparse-view 360° reconstructions challenging. Our approach integrates object structure priors with point-based rendering, providing a robust solution for sparse-view reconstruction and setting a new benchmark for efficiency and quality in complex scenarios.

### B. Neural Rendering for Sparse View Reconstruction

NeRF’s original approach struggles with undersampled data in sparse-view reconstruction. Methods using SfM-derived visibility or depth [15] show promise but require closely aligned views and costly depth maps. Alternative approaches [4] use monocular depth estimation but often produce coarse results. Vision-language models [16] ensure semantic consistency but lack fine details. Methods like [3] capture overall appearance but miss details.

The GaussianObject model [7] introduced structure-prior Gaussian initialization, enabling reconstruction from sparse inputs by reducing required views from over twenty (e.g., FSGS [5]) to four. However, it and similar models struggle with sparse 360° views, often relying on SfM points, which limit performance with scarce data. Our approach enhances GaussianObject by integrating a robust framework for sparse setups, reducing dependence on precise point clouds and improving neural rendering’s real-world applicability.

### III. METHOD

This section explains our self-augmented Gaussian splatting method, beginning with an overview of key ideas. We then outline the two-stage training process, perceptual data augmentation using 2D prior knowledge, and the integration of structural masks in the Gaussian process, followed by the training objective.

#### A. Overview

We start with sparse reference images  $X^{\text{ref}} = \{x_i\}_{i=1}^N$  covering a  $360^\circ$  range, along with camera parameters  $\Pi^{\text{ref}} = \{\pi_i\}_{i=1}^N$  and masks  $M^{\text{ref}} = \{m_i\}_{i=1}^N$ . The goal is a two-stage 3D Gaussian model  $\mathcal{G}$  for photo-realistic rendering from any viewpoint. In the first stage, the model generates coarse point clouds and renderings, optimized by perceptual data augmentation as pseudo-labels. Point-based masks and patch-based masks are used for the two-stage Gaussians, respectively. The framework is shown in Fig. 2.

#### B. Two-stage Training Gaussians

Sparse views, especially with only four images, offer minimal 3D information. In such cases, essential SFM points for initializing 3DGS are often lost, leading to missing geometry and reduced accuracy and efficiency in 3D Gaussian densification.

To address geometric defects and density gaps in sparse views, the Gaussian point distribution is refined stage by stage. Using perceptual-level data augmentation, we guide point distribution in sparse regions, improving geometric consistency and density. In the first stage, we train a set of coarse Gaussian points, focusing on overall geometry and providing a spatial framework for maintaining structural coherence in unobserved regions. The coarse point cloud  $\mathcal{P}_c = \{p_i : \mu_i, q_i, s_i, \sigma_i, sh_i\}_{i=1}^P$  consists of Gaussian points with parameters: center  $\mu$ , rotation quaternion  $q$ , scaling vector  $s$ , opacity  $\sigma$ , and SH coefficient  $sh$ .

In the second stage, Gaussian attributes are refined with perceptual data augmentation, improving geometric consistency and filling sparse regions. We convert spherical harmonics to color, keeping only spatial coordinates  $\mu_i$  and color  $c_i$ , while excluding other attributes. This focus enhances geometry without adding complexity, simplifying model precision.

#### C. Perceptual View data Augmentation

3D Gaussians from sparse views often show blurriness or incoherence due to incomplete perspective information. However, a pre-trained 2D prior can restore missing details, creating more realistic images. We propose using multi-view coarse renders and the 2D prior to enhance perceptual views.

Following [7], we add 3D noise  $\epsilon_s$  to Gaussian attributes, as the leave-one-out approach is slow with little improvement. Degraded images are rendered from noisy Gaussians, and LoRA weights fine-tune a pre-trained Control-Net [6] using these images, based on Stable Diffusion v1.5 [17].

We render images  $X^c = \{x_j\}_{j=1}^{N'}$  from the first-stage Gaussians in  $N'$  novel views. These coarse images are input

into the fine-tuned 2D prior model to generate restored images  $\tilde{X}^c$ , and we augment the reference images to  $X^{\text{aug}} = X^{\text{ref}} \cup \tilde{X}^c$  with corresponding camera parameters  $\Pi^{\text{aug}} = \Pi^{\text{ref}} \cup \Pi^c$ . A second-stage training on the Gaussians yields photo-realistic renderings from any viewpoint,  $x = \mathcal{G}_f(\pi|\{x_i, \pi_i\}_{i=1}^{N+N'})$ . Our approach, relying solely on the provided dataset without external visual data, ensures high accuracy and consistency in the 3D model.

The fine-tuned 2D prior model optimizes multi-view images from the second-stage Gaussians, improving perceptual quality. Even without post-processing, our method outperforms [7], accelerating training and inference. With post-processing, it matches their speeds while further enhancing reconstruction quality (Tab. I).

The optimization of the first-stage Gaussians  $\mathcal{G}_c$  and the second-stage Gaussians  $\mathcal{G}_f$  incorporates color and monocular depth losses. The color loss combines L1 and D-SSIM losses:

$$\mathcal{L}_1 = \|x - x^{\text{ref}}\|_1, \mathcal{L}_{\text{D-SSIM}} = 1 - \text{SSIM}(x, x^{\text{ref}}), \quad (1)$$

where  $x$  is the rendering and  $x^{\text{ref}}$  is the corresponding reference image. A shift and scale invariant depth loss is utilized to guide geometry:

$$\mathcal{L}_d = \|D^* - D_{\text{pred}}^*\|_1, \quad (2)$$

where  $D^*$  and  $D_{\text{pred}}^*$  are per-frame rendered depths and monocularly estimated depths [18] respectively. The depth value are computed following a normalization strategy [19]:

$$D^* = \frac{D - \text{median}(D)}{\frac{1}{M} \sum_{i=1}^M |D - \text{median}(D)|}, \quad (3)$$

where  $M$  denotes the number of valid pixels. The overall loss combines these components as

$$\mathcal{L}_{\text{gs}} = (1 - \lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{D-SSIM}} + \lambda_d\mathcal{L}_d, \quad (4)$$

where  $\lambda_{\text{SSIM}}$  and  $\lambda_d$  control the magnitude of each term.

#### D. Integration of Structure-aware Masks in Coarse-to-Fine Gaussian Process

1) *Point-based Masks For the First Stage:* We apply point-based random masks to 3D Gaussians to enhance the model's robustness to sparse inputs and noise. Specifically, we set a mask ratio  $r_c$  and an update interval  $t_c$ . Every  $t_c$  iterations, we randomly select  $r_c * P$  3D points to discard, preventing them from participating in the densification process.

In the first-stage Gaussian model, point-based masks improve structural accuracy by targeting Gaussian centers from sparse data. These masks are dynamically generated by analyzing rendering errors against real images, identifying regions with significant divergence. They direct focus to areas with higher errors, guiding adjustments to Gaussian parameters and conserving resources on well-represented regions, setting the stage for detailed refinement.

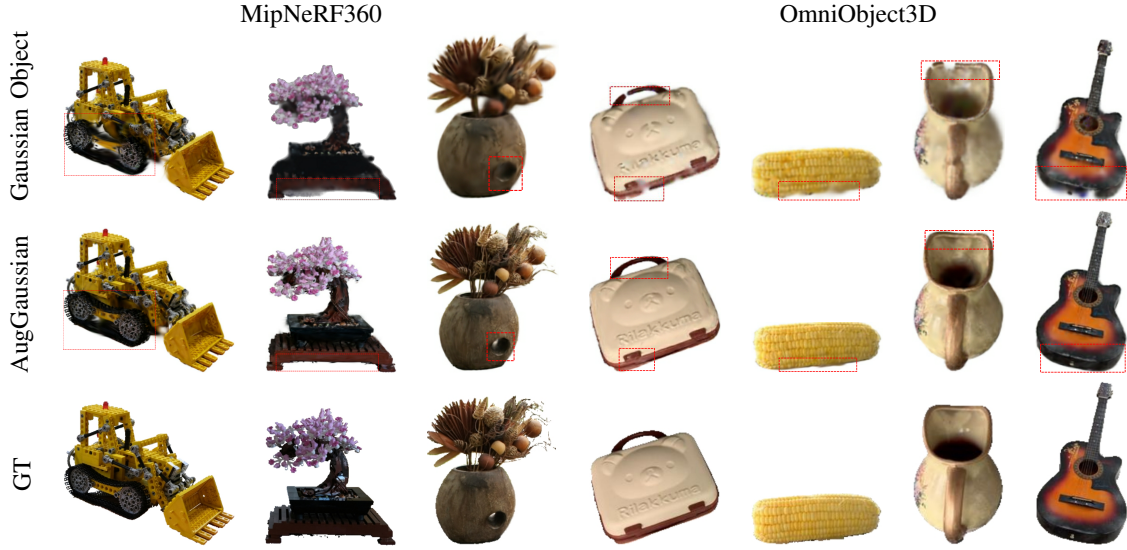


Fig. 3: Qualitative examples on the MipNeRF360 and OmniObject3D dataset with 4 input views.

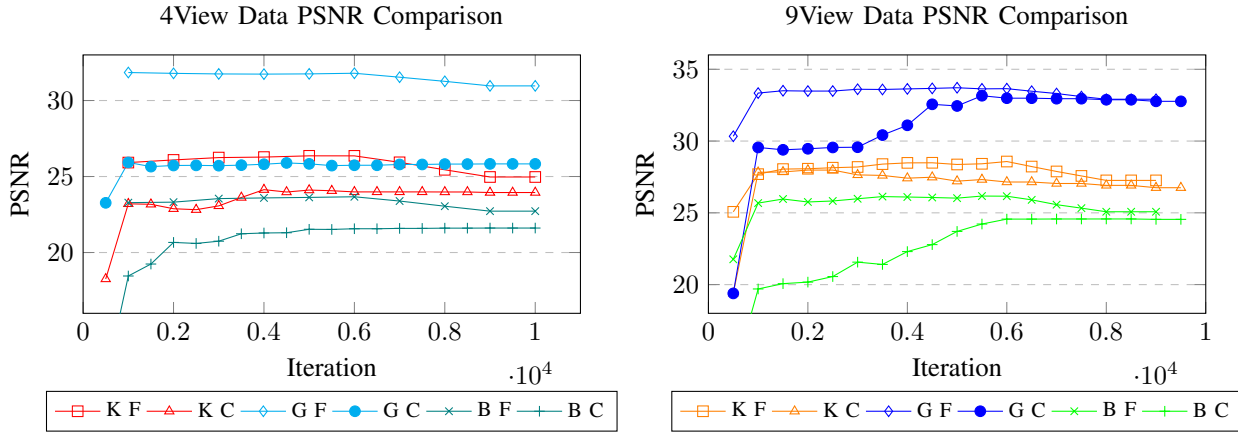


Fig. 4: Comparative analysis of PSNR metrics for 4View and 9View configurations across different objects and Gaussian iteration processes. Note: 'K', 'G', and 'B' represent objects Kitchen, Garden, and Bonsai, respectively. 'C' refers to the Coarse Gaussian iteration process, while 'F' denotes the Fine Gaussian iteration process.

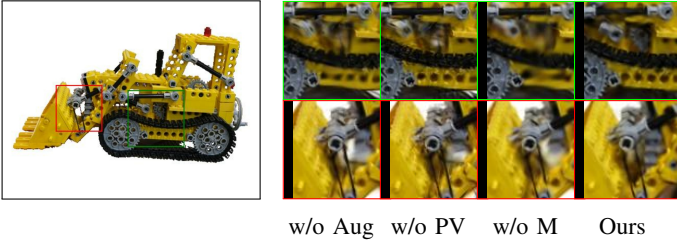


Fig. 5: Ablation study on different augmentation strategies. "Aug" denotes for augmentation, "PV" denotes for perceptual view augmentation and "M" for mask augmentation.

2) *Patch-based Masks For the Second Stage*: Sparse-view images often lack information from certain object parts, like a side of the kitchen. In the SFM point cloud, this appears as

missing or sparse points, which is challenging to fully address during densification.

To enhance the second-stage 3D Gaussian model's robustness to missing data and noise, we use patch-based random masks. We extract  $C$  patch centers via farthest point sampling (FPS) and generate  $C$  patches using the kNN algorithm. With an iteration gap  $t_f$  and mask ratio  $r_f$ , every  $t_f$  iterations,  $r_f * C$  patches are randomly selected and excluded from the densification process. These masks refine second-stage Gaussian parameters, enhancing resolution in key areas. This targeted approach improves perceptual quality, using iterative feedback for high-fidelity reconstruction while preserving authenticity without external data.



TABLE I: Comparisons on MipNeRF360 and OmniObject3D datasets with varying input views. LPIPS\* = LPIPS  $\times 10^2$  throughout this paper. PP represents the post-processing step using the fine-tuned 2D prior model to optimize the final output images.

	Method	4-view			6-view			9-view		
		LPIPS* $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS* $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS* $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
MipNeRF360	DVGO [20]	24.43	14.39	0.7912	26.67	14.30	0.7676	25.66	14.74	0.7842
	3DGS [14]	10.80	20.31	0.8991	8.38	22.12	0.9134	6.42	24.29	0.9331
	DietNeRF [1]	11.17	18.90	0.8971	6.96	22.03	0.9286	5.85	23.55	0.9424
	RegNeRF [2]	20.44	13.59	0.8476	20.72	13.41	0.8418	19.70	13.68	0.8517
	FreeNeRF [21]	16.83	13.71	0.8534	6.84	22.26	0.9332	5.51	27.66	0.9485
	SparseNeRF [4]	17.76	12.83	0.8454	19.74	13.42	0.8316	21.56	14.36	0.8235
	ZeroRF [3]	19.88	14.17	0.8188	8.31	24.14	0.9211	5.34	27.78	0.9460
	FSGS [5]	9.51	21.07	0.9097	7.69	22.68	0.9264	6.06	25.31	0.9397
	GaussianObject [7]	4.98	24.81	0.9350	3.63	27.00	0.9512	2.75	28.62	0.9638
	Ours (w/o PP)	4.05	27.85	0.9515	3.27	29.12	0.9621	2.67	30.25	0.9701
OmniObject3D	Ours (full setting)	<b>3.85</b>	<b>28.46</b>	<b>0.9587</b>	<b>3.14</b>	<b>30.02</b>	<b>0.9685</b>	<b>2.60</b>	<b>30.96</b>	<b>0.9732</b>
	DVGO [20]	14.48	17.14	0.8952	12.89	18.32	0.9142	11.49	19.26	0.9302
	3DGS [14]	8.60	17.29	0.9299	7.74	18.29	0.9378	6.50	20.26	0.9483
	DietNeRF [1]	11.64	18.56	0.9205	10.39	19.07	0.9267	10.32	19.26	0.9258
	RegNeRF [2]	16.75	15.20	0.9091	14.38	15.80	0.9207	10.17	17.93	0.9420
	FreeNeRF [21]	8.28	17.78	0.9402	7.32	19.02	0.9464	7.25	20.35	0.9467
	SparseNeRF [4]	17.47	15.22	0.8921	21.71	15.86	0.8935	23.76	17.16	0.8947
	ZeroRF [3]	4.44	27.78	0.9615	3.11	31.94	0.9731	3.10	32.93	0.9747
	FSGS [5]	6.25	24.71	0.9545	6.05	26.36	0.9582	4.17	29.16	0.9695
	GaussianObject [7]	2.07	30.89	0.9756	1.55	33.31	0.9821	1.20	35.49	0.9870
	Ours (w/o PP)	1.87	32.59	0.9796	1.54	33.75	0.9844	1.18	35.51	0.9876
	Ours (full setting)	<b>1.75</b>	<b>34.02</b>	<b>0.9834</b>	<b>1.48</b>	<b>34.19</b>	<b>0.9912</b>	<b>1.12</b>	<b>35.86</b>	<b>0.9931</b>

TABLE II: Quantitative comparisons on the OpenIllumination dataset. Methods with  $\dagger$  mean the metrics are from the Gaussian Object paper.

Method	4-view			6-view		
	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
DVGO $\dagger$	11.84	21.15	0.8973	8.83	23.79	0.9209
3DGS $\dagger$	30.08	11.50	0.8454	29.65	11.98	0.8277
DietNeRF $\dagger$	10.66	23.09	0.9361	9.51	24.20	0.9401
RegNeRF $\dagger$	47.31	11.61	0.6940	30.28	14.08	0.8586
FreeNeRF $\dagger$	35.81	12.21	0.7969	35.15	11.47	0.8128
SparseNeRF $\dagger$	22.28	13.60	0.8808	26.30	12.80	0.8403
ZeroRF $\dagger$	9.74	24.54	0.9308	7.96	26.51	0.9415
G.O. $\dagger$	6.71	24.64	0.9354	5.44	26.54	0.9443
Ours (w/o PP)	5.86	24.66	0.9372	5.08	26.55	0.9453
Ours (full setting)	<b>5.75</b>	<b>25.23</b>	<b>0.9458</b>	<b>4.79</b>	<b>27.00</b>	<b>0.9582</b>

TABLE III: Ablation study of augmentation strategies conducted on the 'kitchen' dataset with 4 input views.

Setting	LPIPS* $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
w/o augmentation	8.17	22.82	0.9141
w/o perceptual view augmentation	6.76	23.98	0.9228
w/o mask augmentation	6.56	24.11	0.9229
Ours (full setting)	<b>5.14</b>	<b>26.87</b>	<b>0.9379</b>

## IV. EXPERIMENT

### A. Implementation Detail

Our framework extends 3D Gaussian Splatting (3DGS) [14] with a two-stage approach for sparse-view 3D reconstruction. The first stage involves multiple training iterations, utilizing ZoeDepth [18] for accurate depth estimation. A point-based mask strategy selectively refines critical geometric areas to enhance fidelity.

In the second stage, a fine Gaussian model is trained with a patch-based mask strategy to refine surface details. Using FPS

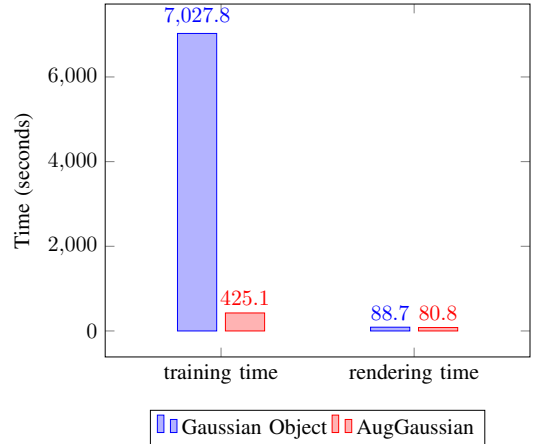


Fig. 6: Efficiency comparison under same settings with 4 input views on dataset MipNeRF360.

and KNN for point cloud segmentation, it enhances texture and detail. The process takes under 10 minutes on a Tesla T4 GPU with 4, 6, and 9 779 $\times$ 520 images, demonstrating its efficiency for high-fidelity 3D reconstruction from sparse data.

### B. Dataset

We evaluate our method on three datasets for sparse-view 360 $^\circ$  object reconstruction: Mip-NeRF360 [8], OmniObject3D [9], and OpenIllumination [10]. Mip-NeRF360 extends mip-NeRF [22] to handle unbounded scenes using non-linear parameterization and online distillation. OmniObject3D includes over 6,000 real-scanned 3D objects across 190 categories, enhancing 3D representation learning. OpenIllumination, cap-

tured with LightStage, is utilized in its sparse form as introduced in ZeroRF [3].

### C. Evaluation

We compare our AugGS model to state-of-the-art 3D reconstruction methods, including GaussianObject [7], 3DGS [14], DVGO [20], RegNeRF [2], DietNeRF [1], SparseNeRF [4], ZeroRF [3], and FSGS [5]. GaussianObject uses visual hull techniques and a pretrained repair model, while FSGS incorporates SfM-point initialization with additional points for sparse 360° views. Baseline and few-view models employ various regularization strategies. All models are evaluated using their publicly available implementations to ensure consistent comparisons (see Tables I and II). Our AugGS outperforms GaussianObject on key metrics across three datasets without extra optimizations, demonstrating our lightweight model’s efficiency and superior PSNR for image quality.

Our structural training approach addresses sparse data challenges and scales with more data. On diverse datasets, our model surpasses state-of-the-art benchmarks in PSNR and SSIM. Compared to GaussianObject [7], we improved LPIPS by 5–30%, PSNR by 8–25%, and SSIM by 5–10%. Qualitative results show smoother, detailed reconstructions of complex structures like gears and flower pots, demonstrating our model’s superior handling of intricate details.

### D. Ablation Studies

In our ablation study, we evaluate four augmentation setups: no augmentation, disabling perceptual view augmentation, disabling mask augmentation, and combining all. As shown in Tab. III and Fig. 5, each augmentation improves performance, with the combined approach achieving the highest fidelity. Without all augmentations, the model fails to accurately reconstruct obscured structures like the quarter-exposed gear and vehicle front. Using all strategies enable effective detail capture, highlighting the essential role of augmentation in enhancing structural integrity and surface detail for high-quality 3D reconstructions.

We have also examined how iteration numbers affect model performance on the MipNeRF360 and Omni3D datasets. The coarse Gaussian phase uses groups every 500 iterations, while the fine phase uses every 1000. Optimal coarse iterations range from 1000 to 5000, and fine Gaussian peak around 6000 iterations. More complex structures require additional fine iterations, indicating that structural complexity necessitates tailored iteration counts. Optimal coarse iterations do not guarantee better fine refinement, highlighting the need for a customized approach based on object complexity.

Additionally, we evaluate the efficiency of our AugGS model (Fig. 6). Compared to the state-of-the-art GaussianObject [7], AugGS reduces training time to 6.3% and overall runtime to 7.1% by utilizing a simple framework without complex feature extraction or pre-trained models. On the 4-view kitchen dataset, AugGS maintains similar rendering times while requiring less memory and GPU resources. This greater efficiency accelerates the research and development

cycle and enables deployment on edge devices, making 3D reconstruction technology more accessible and sustainable for real-world applications.

## V. CONCLUSION

This paper presents a sparse-view 3D reconstruction method using self-augmented Gaussian splatting method and structure-aware masks to generate high-fidelity models from limited 2D inputs. Our method improves geometric precision, outperforming state-of-the-art methods with higher efficiency and lower cost. Future work will optimize modeling, enhance mask generation, and extend to dynamic scenes, setting a new benchmark for sparse-view 3D reconstruction.

## REFERENCES

- [1] Ajay Jain, Matthew Tancik, and Pieter Abbeel, “Putting nerf on a diet: Semantically consistent few-shot view synthesis,” 2021.
- [2] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan, “Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [3] Ruoxi Shi, Xinyue Wei, Cheng Wang, and Hao Su, “Zerorf: Fast sparse view 360deg reconstruction with zero pretraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21114–21124.
- [4] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu, “Sparsenerf: Distilling depth ranking for few-shot novel view synthesis,” *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [5] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang, “Fsgs: Real-time few-shot view synthesis using gaussian splatting,” 2023.
- [6] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [7] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian, “Gaussianobject: Just taking four images to get a high-quality 3d object with gaussian splatting,” 2024.
- [8] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469, 2021.
- [9] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu, “Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation,” 2023.
- [10] Isabella Liu, Linghao Chen, Ziyang Fu, Liwen Wu, Haian Jin, Zhong Li, Chin Ming Ryan Wong, Yi Xu, Ravi Ramamoorthi, Zexiang Xu, et al., “Openillumination: A multi-illumination dataset for inverse rendering evaluation on real objects,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [11] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson, “Synsin: End-to-end view synthesis from a single image,” 2020.
- [12] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann, “Point-nerf: Point-based neural radiance fields,” 2023.
- [13] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide, “Differentiable point-based radiance fields for efficient view synthesis,” 2023.
- [14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, “3d gaussian splatting for real-time radiance field rendering,” 2023.
- [15] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan, “Depth-supervised nerf: Fewer views and faster training for free,” 2022.

- [16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, “Learning transferable visual models from natural language supervision,” in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, PMLR.
- [17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- [18] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller, “Zoedepth: Zero-shot transfer by combining relative and metric depth,” 2023.
- [19] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” 2020.
- [20] Cheng Sun, Min Sun, and Hwann-Tzong Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction,” 2022.
- [21] Jiawei Yang, Marco Pavone, and Yue Wang, “Freenerf: Improving few-shot neural rendering with free frequency regularization,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 8254–8263.
- [22] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan, “Mip-nerf: A multi-scale representation for anti-aliasing neural radiance fields,” *CoRR*, vol. abs/2103.13415, 2021.