PromptGS: Visual Prompting for Tiny Object Reconstruction in 3DGS Optimization

1st Xun Wang
Beihang University
Beijing, China
xun@buaa.edu.cn

2nd Xutao Xue
Beihang University
Beijing, China
xuexutao@buaa.edu.cn

3rd Siyuan Li
Hangzhou International Innovation Institute,
Beihang University, China
lsy316@buaa.edu.cn

4th Shayer Shabab Utsho
Beihang University
Beijing, China
s.ssutsho.ch@gmail.com

5th Kun Li
Tianjin University
Tianjin, China
lik@tju.edu.cn

6th Mengqi Ji✉
Beihang University
China
jimengqi@buaa.edu.cn

***Abstract*—Reconstructing tiny floating objects in large-scale 3D scene remains a fundamental challenge for 3D Gaussian Splatting (3DGS). These objects often receive insufficient point density and gradient supervision during training due to limited visibility and low image-space saliency, making them difficult to recover even after prolonged optimization. We present PromptGS, a visual prompting framework that incorporates lightweight human input to guide the 3DGS optimization process. PromptGS fuses projected 2D error maps with user-specified spatial prompts to form a 3D attention field, which acts as an optimization prior to guide Gaussian densification, adaptive resampling, and multiview selection. This mechanism directs training efforts toward regions with high semantic relevance but low point density, improving reconstruction in areas that are frequently overlooked. Furthermore, we design a Gaussian scoring function that ranks candidates based on their improvement potential, ensuring efficient resource allocation. Moreover, PromptGS achieves multiview consistent rendering of small objects, indicating that their geometry and appearance are faithfully reconstructed in 3D space rather than approximated through view-dependent texture projection. Experiments on public benchmarks and challenging synthetic scenes demonstrate that PromptGS consistently outperforms existing methods in both visual fidelity and efficiency.**

***Index Terms*—visual prompt 3D reconstruction, tiny object reconstruction, multiview consistency, Gaussian Splatting**

I. Introduction

In large-scale 3D scenes, accurately reconstructing small-scale and sparsely distributed objects remains a critical yet under-addressed challenge. Objects such as drones or debris often occupy only a few pixels in multiview images and be observed from limited views. Despite their small size, these objects may carry essential semantic or functional information. In safety-critical applications such as autonomous driving, robotics, or disaster response, failing to reconstruct such tiny objects can lead to incorrect scene understanding, missed detections, or unsafe behavior. Therefore, improving the reconstruction quality of small and low-saliency structures is vital for the reliability and robustness of 3D perception systems.

Corresponding author: jimengqi@buaa.edu.cn

However, current neural rendering pipelines, including recent advances such as 3D Gaussian Splatting (3DGS) [1], are not well-equipped to handle this challenge. 3DGS and its variants [2]–[4] generally rely on traditional structure-from-motion (SfM) and multiview stereo (MVS) algorithms, such as COLMAP [5], to initialize sparse point clouds. Yet, these classical methods often fail to reconstruct small-scale objects due to their low pixel coverage and lack of visual saliency, resulting in missing geometry from the very beginning of the pipeline. Although some methods introduce densification heuristics or multiview-aware training, they lack a principled mechanism to identify and prioritize semantically important but structurally underrepresented regions. Furthermore, existing pipelines lack mechanisms to incorporate human-defined spatial priors, which could help focus training on semantically important regions.

To address this gap, we propose **PromptGS**, a visual prompting framework that introduces lightweight human guidance into the 3DGS optimization loop. PromptGS allows users to mark spatial regions of interest, which are fused with projected 2D error maps to construct a unified 3D attention field. This attention field serves as an optimization prior, dynamically guiding Gaussian densification, adaptive resampling, and multiview selection toward regions that require focused reconstruction.

In addition, we introduce a Gaussian scoring function that evaluates each point's potential contribution to reconstruction quality, enabling efficient prioritization of training resources. PromptGS also supports prompt-guided resampling in sparse regions, increasing point cloud density where it is most needed. These mechanisms jointly enable the system to reconstruct small-scale objects with higher fidelity and consistency across views.

We validate PromptGS on public benchmarks and challenging synthetic scenes containing small, difficult-to-capture objects. Experimental results show that our method outperforms existing 3DGS variants in both visual fidelity and reconstruction completeness. Notably, PromptGS achieves multiview

Fig. 1. Overview of our method. The core idea of our method revolves around interactive human guidance, using a "circling and highlighting" method to focus on regions that require densification. In this region, we implement focus train by matching the most similar input views and prioritize the densification of Gaussians in the region according to their performance improvement scores. Resampling is applied to sparse small floating objects to enhance the accuracy of reconstruction.

consistent modeling of small objects, faithfully capturing their geometry and appearance in 3D space, rather than relying on view-dependent texture approximation.

This work contributes a general and interpretable human-guided optimization framework for 3D scene reconstruction, enabling 3DGS to recover semantically important tiny structures that are often overlooked by existing methods.

II. Literature Review

A. Traditional 3D Reconstruction Methods

Traditional 3D reconstruction methods, such as multiview stereo (MVS) and structure-from-motion (SFM) algorithms, are widely used in 3D reconstruction. MVS reconstructs 3D scenes by matching and estimating depth from multiple-view images. SFM recovers camera motion trajectories and scene structure from multiple images to achieve 3D reconstruction. These methods, which rely on meshes and point clouds [6]–[9], or implicit representations [10] often face challenges in accuracy and efficiency when dealing with complex scenes. Some studies [11] [12], introduce a deep learning algorithm to improve the accuracy and robustness of 3D model reconstruction through the end-to-end learning framework and the volume view selection method.

B. The Emergence of Neural Radiance Fields

Neural Radiance Fields (NeRF) [13] represent a significant advancement in neural scene representation, enabling high-fidelity novel view synthesis from multiview images. While NeRF achieves high-quality rendering, its reliance on dense volumetric sampling and per-ray MLP inference leads to high computational cost, limiting its applicability in real-time scenarios. To address this, researchers have explored various optimization methods to improve its computational efficiency and practicality. Examples of such algorithms include Mip-NeRF [14], InstantNGP [15], Mip-nerf [16], and Plenoxels [17], all of which aim to enhance NeRF's computational efficiency.

C. 3D Gaussian Splatting

Gaussian Splatting [1] is a novel 3D reconstruction method that achieves real-time rendering by representing scenes with spatially continuous Gaussians, enabling fast and photorealistic novel view synthesis. However, existing 3DGS methods still face key challenges when handling complex scenes, including over-reconstruction, redundant Gaussian function generation, and resource inefficiency. To address these issues, by introducing a new codirectional view space position gradient as a densification criterion, AbsGS [2] effectively identifies and splits large Gaussian distributions in over-reconstructed regions, thereby recovering fine details. ResGS [18] proposes a residual segmentation densification method that progressively refines the details by adaptively adding smaller Gaussians to complement geometric details. Scaffold-GS [19] reduces redundant Gaussian functions by using a neural Gaussian anchor distribution and improves scene coverage through anchor growth and pruning strategies. MVGS [4] introduces a multiview training strategy that optimizes multiview attributes jointly, effectively overcoming the issue of overfitting to single views and significantly enhancing 3DGS reconstruction quality.

Although these methods improve certain aspects of 3DGS, none of them addresses the challenge of identifying and optimizing small-scale objects that are missing from initialization or receive weak supervision during training. In contrast, our

method introduces an externally guided optimization mechanism, allowing users to inject spatial priors that guide densification and view sampling toward critical regions, thereby enabling targeted reconstruction of small and underrepresented objects.

III. Method

Our goal is to improve the reconstruction of small-scale, sparsely distributed objects in large-scale 3D scenes using 3D Gaussian Splatting (3DGS). To this end, we propose PromptGS, a framework that introduces lightweight human guidance into the 3DGS optimization process through spatial prompts. PromptGS consists of three key components: (1) *Error-Aware View Selection* localizes high-loss regions via projected error maps and refines them through similarity-based multiview selection; (2) *Prompt-Driven Gaussian Prioritization* ranks Gaussians by their reconstruction contribution and prioritizes densification in attention-weighted critical areas; and (3) *Attention-Guided Resampling* augments point density in low-coverage zones to recover tiny object structure.

A. Preliminaries: 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents 3D scenes as a set of spatially continuous Gaussians, enabling efficient and photorealistic rendering in real time. Unlike NeRF-based methods that rely on volumetric sampling and MLP inference, 3DGS directly optimizes Gaussian parameters, significantly reducing redundancy and enabling real-time rendering. Each Gaussian is parameterized by a center $\mu \in \mathbb{R}^3$, a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ encoding shape and orientation, and an opacity scalar $o \in \mathbb{R}$. The theoretical contribution of a single 3D Gaussian to a point x in space is formulated as follows:

$$G(x) = oe^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \quad \Sigma = RSS^TR^T \tag{1}$$

where R denotes the rotation matrix and S is the scaling matrix. The anisotropic covariance of each Gaussian enables it to adaptively model both sharp edges and smooth regions, improving geometric fidelity. View-dependent appearance is modeled using third-order Spherical Harmonics, which provide a compact and differentiable representation of reflectance and lighting effects. For a particular viewpoint, the visible set of 3D Gaussians is rendered in a tile-based, differentiable rasterizer to obtain a 2D image by α-blending their projections. During training, the parameters of all Gaussians, including position, orientation, scale, opacity, and SH coefficients, are jointly optimized by minimizing a composite loss that combines pixel-wise L1 error and SSIM between rendered and ground truth images.

B. Error-Aware View Selection

The standard single-view random sampling strategy in 3DGS often fails to capture high-error regions, especially when these areas are occluded or only visible in a few views. As a result, such regions may be optimized only sporadically, leading to inefficient and non-continuous convergence. Consequently, error-prone regions in complex scenes may remain underoptimized even after prolonged training, resulting in slow densification and inefficient allocation of computational resources. To identify high-error regions, we compute the 2D per-pixel loss $\mathcal{L}^{2D}(x, y)$ between rendered and ground truth images, and back-project it into 3D space using known camera intrinsics and poses. The accumulated 3D loss $\mathcal{L}^{3D}(X)$ highlights regions with persistent reconstruction errors. For each such region, we retrieve K camera views with the highest image space similarity to the projected region, enabling targeted multiview optimization.

The practical aspects of the representation are highly dependent on the initial distribution of point clouds [20]. To address this challenge more effectively, we employ a progressive training strategy. Specifically, we start by downsampling the input images and conducting training on low-resolution data, which helps generate a reasonably distributed initial point cloud. We then gradually increase the resolution of the input images during subsequent training stages, continuously refining the model to produce more detailed and accurate point cloud reconstructions.

Given the per-pixel loss maps and calibrated camera poses, we back-project high-loss pixels into 3D space to form a volumetric error field. Regions with high accumulated errors are selected as reconstruction-critical targets for further optimization. By retrieving camera views with high similarity to the identified high-error regions, we enable targeted multiview optimization that concentrates resources on the most error-prone areas, leading to improved reconstruction accuracy. To balance memory usage and training diversity, we alternate between random multiview sampling and similarity-based view selection. When GPU memory is constrained, we dynamically adjust the number of active views by sampling subsets, ensuring full camera coverage over time without exceeding memory limits. This strategy enables targeted multiview training, directing computational resources to high-loss areas, and improving the overall accuracy of the model. Cameras with higher similarity are prioritized for computation. By visualizing 3D loss distributions, the system allows users to specify spatial prompts in regions with persistent errors, enhancing optimization in areas where automatic methods under-perform.

C. Prompt-Driven Gaussian Prioritization

The original 3DGS pipeline often fails to reconstruct small-scale, sparsely distributed objects due to weak optimization signals and insufficient Gaussian density in these regions. Throughout the training process, the algorithm has self-diagnostic capabilities, which automatically detect and mark high-loss regions and key points that need optimization. To address this, we introduce lightweight human guidance in the form of spatial masks, allowing users to specify regions of interest that are underrepresented in the training signal. Meanwhile, a new score function is introduced to prioritize densifying Gaussians that yield higher performance improvement, avoiding wasted time on unnecessary densification and excessive resource consumption.

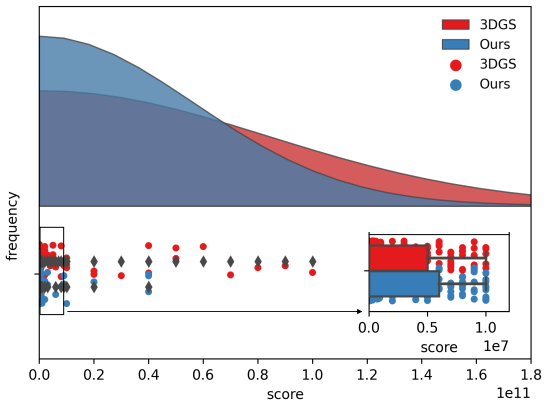

Fig. 2. The distribution of Gaussian scores. Compared with 3DGS, after prioritizing the densification of higher score Gaussians, the number of Gaussians with abnormally high scores decreases, the overall quantity increases, and the average score in the lower range shows a reasonable improvement.

Certain regions receive insufficient supervision during training, leading to poor convergence. Human-provided spatial priors help to identify and refine these regions more effectively. Users can interactively select high-error regions based on visualized 3D loss maps and provide spatial masks to guide focused optimization in these areas. For marked regions, we use images with high viewpoint relevance for focused training.

We define S_g as the Gaussian score for each Gaussian g. N is the total number of views, ∇g represents the Gaussian positional gradient, and d_g^i is the depth attribute in the i-th view of Gaussian g. The opacity, scale, and radius of Gaussian g are represented by o_g, s_g, r_g , respectively. To evaluate the Gaussian component, we introduce a scoring function F that combines these terms.

The combined loss P^i for the i-th view is a combination of the L_1 loss L_1^i, the loss of the Structural Similarity Index (D-SSIM) L_{D-SSIM}^i, and the edge loss L_{edge}^i. The Gaussian score S_g is then given by the equation:

$$S_g = \sum_{i=1}^{N} P^i \cdot F\left(\nabla_g + d_g^i + o_g + s_g + r_g\right) \tag{2}$$

where P^i is the combined loss for the i-th view, given by:

$$P^i = \lambda_1 L_1^i + \lambda_2 L_{D-SSIM}^i + \lambda_3 L_{edge}^i \tag{3}$$

In addition to the L1 and SSIM loss used in the original 3DGS, we incorporate depth and edge-aware losses to better capture geometric and structural details in sparse regions. The rationale for excluding semantic loss from the computation is to prevent introducing extra attributes to Gaussian primitives, which might decelerate the algorithm. Instead, depth and edge loss play analogous roles. As shown in Fig. 2, the data distribution first follows a long-tail distribution. Compared to 3DGS, after prioritizing the densification of higher-scoring Gaussians, the number of Gaussians increases, and the average value of the low-score part increases, while the number of Gaussians with abnormally high scores gradually decreases. This suggests that the scoring function effectively prioritizes Gaussians in underrepresented regions, leading to more balanced point distributions and improved reconstruction quality.

Fig. 3. Render results with mask (right) and without mask (left) are presented. With human-interaction method, the reconstruction shows fewer holes.

By calculating and ranking the scores of Gaussians within the region, we ensure that the highest-scoring parts, which contribute the most to overall model performance, are prioritized for densification. By scoring Gaussians with a combination of photometric and geometric losses, the algorithm prioritizes meaningful regions and suppresses unnecessary densification, thereby reducing redundant computation in over-saturated areas. By focusing on key regions and not computing loss for low-relevance views, we can significantly reduce resource wastage. According to the prompts of high-loss areas, human experts can mark and provide regions of interest and camera indices, guiding the algorithm densification. As shown in Figure 3, the input views on the right side indicate that due to perspective occlusion and the small proportion of the marked region in the scene, the original 3DGS rendering results are filled with voids. The addition of human-guided prompts leads to more complete surface coverage and finer geometric details in previously underrepresented regions.

D. Attention-Guided Resampling

To enhance the reconstruction of small-scale, sparsely distributed objects, we perform Gaussian resampling in user-specified regions of interest, increasing point density where the original representation is insufficient. Since reconstruction quality is closely correlated with point density [20], we perform targeted resampling to increase Gaussian density in underrepresented regions, thereby improving coverage of low-saliency objects. By concentrating training on resampled regions and their relevant camera views, the method improves both reconstruction quality and computational efficiency, avoiding unnecessary updates in low-impact areas.

IV. Experiments

A. Setup

Datasets. Following 3DGS, we select scenes with highly diverse capture styles, ranging from enclosed indoor environments to expansive outdoor settings without clear boundaries. Specifically, we use all 9 unbounded indoor and outdoor scenes presented in Mip-NeRF360, two scenes from and two scenes

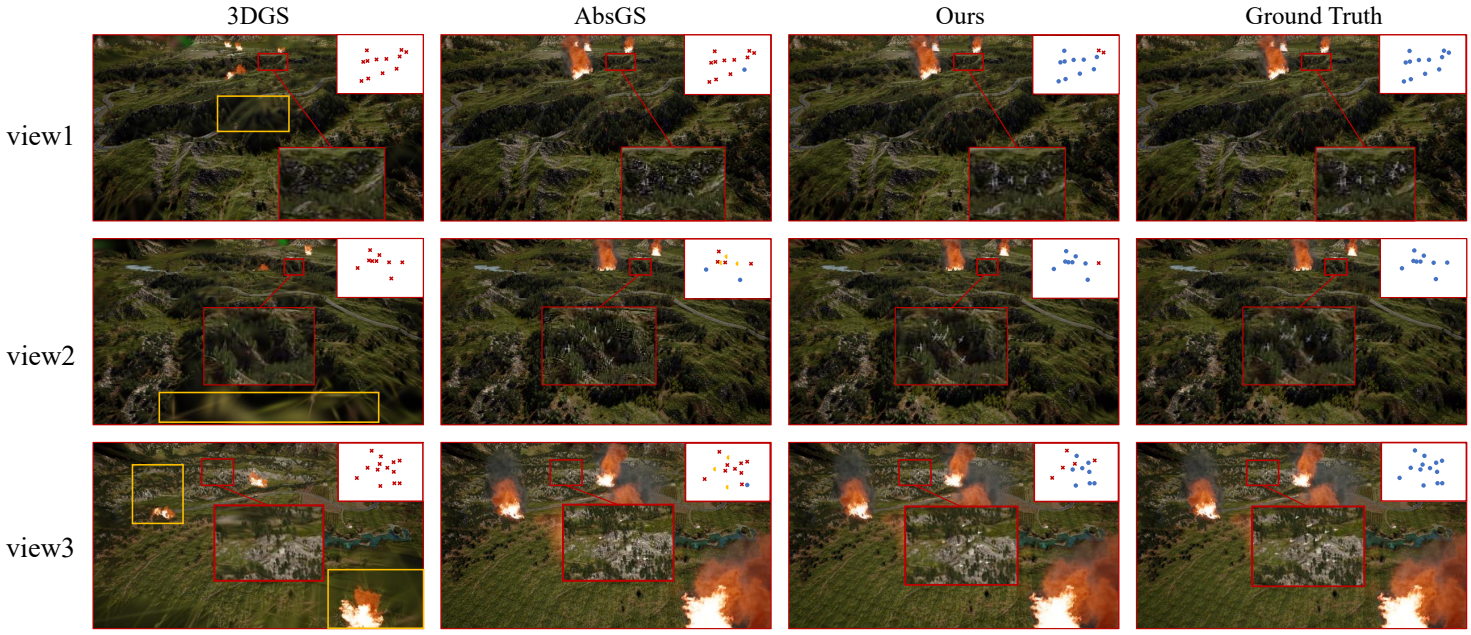

Fig. 4. Qualitative Comparisons of Different Methods on Scenes from Mountain Datasets. Our method clearly reconstructs the majority of the drones and provides better details on the slopes. In contrast, AbGS only reconstructs a few drones and incorrectly places many of the white drone patches on the slopes. Compared to 3DGS, our method not only avoids large areas of blurring but also reconstructs the flames more accurately.

provided by Tanks and temples dataset and Deep blending dataset, totaling 13 distinct environments. To ensure a fair comparison, all numerical data for these methods presented in the tables are directly sourced from the original publication. To handle complex scenes and tiny moving objects, we used simulation software to generate mountainous terrain and add flames and tiny floating drones. From the camera's perspective, the drones occupy no more than 10 pixels in 1080p images, thereby ensuring that they appear as tiny and hard-to-detect moving objects.

Implementation Details. Our experiments are conducted on a single NVIDIA 4090 GPU with 24GB of memory. Following common practice, we stop the Gaussian densification after 15,000 iterations and stop training at 30,000 iterations. Each downsampling factor (8×, 4×, 2×) is used for 7000 training epochs respectively during the progressive training strategy with low-resolution images. The multiview training strategy combines random multiview, related multiview, and random singleview in a ratio of 2:2:6. To avoid out-of-memory issues, we maximize the distance between the random multiview and specified related multiview, using single views to separate them, resulting in a final ratio distribution of 2:3:2:3. For the selection of masks in Table II, we used a centered mask to delineate the regions where x, y, and z are 1, and a mask of size 1*1*1 to delineate the regions. All Gaussians within the mask are densified, the default selection being the center. If the center is within the object and not on the surface, a small adjustment is made. We did not choose a random mask because the corresponding camera indices found would be fewer, so the center was selected. The size of the specified mask for delineation is the same as above, with the main region selected on the surface.

B. Performance Evaluation

In standard datasets shown in Table I, our method achieves high PSNR scores for reconstruction. Although a centered mask generally performs well, a user-defined mask often yields better results across most metrics. In the simulated mountain dataset, the advantages of our method are particularly evident. As shown in Table II, adding masks on the hillside or multiple masks significantly improves the results. The visual comparison shows that applying user-defined masks leads to improved object visibility and sharper boundaries in the rendered results. This comparison isolates the effect of spatial prompting by evaluating reconstructions with and without user-defined masks. For the tiny floating drones in the air, we have a general idea of their location, but not their exact positions. We therefore delineate the approximate area where the drones are located. The displayed images show that our algorithm can effectively reconstruct the drones, which account for only approximately 1/200 of the input image pixels. This result verifies that the algorithm ensures the successful reconstruction of such tiny objects.

TABLE I

QUANTITATIVE EVALUATION OF OUR METHOD COMPARED TO PREVIOUS WORK. EVALUATED OUR METHOD ON THREE DATASETS AND COMPARED IT WITH PREVIOUS WORK.RESULTS FROM OTHER ALGORITHMS ARE DIRECTLY ADOPTED FROM THEIR ORIGINAL PAPERS, AND OUR RESULTS ARE FROM OUR OWN EXPERIMENTS. THE BEST RESULTS ARE HIGHLIGHTED IN RED, THE SECOND-BEST IN YELLOW, AND THE THIRD-BEST IN LIGHT BLUE. OURS* REFERS TO SELECTING ONLY ONE FOCUS TRAIN REGION, WHICH IS POSITIONED AT THE CENTER OF THE COORDINATE SYSTEM. OURS REFERS TO SELECTING A FOCUS TRAIN REGION TOO, BUT THE REGION IS CHOSEN THROUGH HUMAN INTERACTION.

Dataset	Mip-NeRF360			Tanks&Temples			Deep Blending		
Method | Metric	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$
Plenoxels	0.626	23.08	0.463	0.719	21.08	0.379	0.795	23.06	0.510
INGP-Base	0.671	25.30	0.371	0.723	21.72	0.330	0.797	23.62	0.423
INGP-Big	0.699	25.59	0.331	0.745	21.92	0.305	0.817	24.96	0.390
3DGS	0.815	27.21	0.214	0.841	23.14	0.183	0.903	29.41	0.243
AbsGS	0.820	27.49	0.191	0.853	23.73	0.162	0.902	29.67	0.236
taming 3dgs	0.851	24.04	0.170	0.822	27.79	0.205	0.907	30.14	0.235
Ours*	0.818	27.95	0.201	0.870	25.02	0.145	0.904	29.98	0.235
Ours	0.825	28.12	0.186	0.891	26.12	0.144	0.908	30.21	0.233

TABLE II

QUANTITATIVE EVALUATION OF OUR METHOD COMPARED TO ABSGS AND 3DGS ON THE MOUNTAIN DATASET.

	PSNR↑	SSIM↑	LPIPS↓
3DGS	21.595	0.563	0.463
AbsGS	27.599	0.808	0.242
Ours(1 mask)	30.165	0.815	0.271
Ours(2 masks)	**31.018**	**0.819**	**0.235**

V. CONCLUSION

3DGS faces significant challenges in handling tiny floating objects in complex large-scale scenes, leading to blurred detailed features due to insufficient point cloud density in critical regions. To address these issues, we propose a training strategy that first identifies high-error regions via 2D-to-3D loss projection and then refines them using targeted multiview optimization. We also incorporate user-provided spatial prompts to focus optimization on areas of importance to semantics but underrepresented. The sparse regions of tiny objects undergo adaptive resampling to enhance local detail. Our method achieves higher accuracy and completeness in reconstructing small-scale objects, as demonstrated on both real-world and synthetic large-scale datasets.

REFERENCES

[1] 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.

[2] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou, "Absgs: Recovering fine details in 3d gaussian splatting," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 1053–1061.

[3] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, "Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 326–342.

[4] X. Du, Y. Wang, and X. Yu, "Mvgs: Multi-view-regulated gaussian splatting for novel view synthesis," *arXiv preprint arXiv:2410.02103*, 2024.

[5] 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.

[6] M. Botsch, A. Hornung, M. Zwicker, and L. Kobbelt, "High-quality surface splatting on today's gpus," in *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.* IEEE, 2005, pp. 17–141.

[7] C. Lassner and M. Zollhofer, "Pulsar: Efficient sphere-based neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1440–1449.

[8] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions On Graphics (TOG)*, vol. 38, no. 6, pp. 1–14, 2019.

[9] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler, "Extracting triangular 3d models, materials, and lighting from images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8280–8290.

[10] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, "Points2surf learning implicit surfaces from point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 108–124.

[11] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2307–2315.

[12] M. Ji, J. Zhang, Q. Dai, and L. Fang, "Surfacenet+: An end-to-end 3d neural network for very sparse multi-view stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4078–4093, 2020.

[13] 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.

[14] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.

[15] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[16] 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.

[17] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.

[18] Y. Lyu, K. Cheng, X. Kang, and X. Chen, "Resgs: Residual densification of 3d gaussian for efficient detail recovery," *arXiv preprint arXiv:2412.07494*, 2024.

[19] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.

[20] S. S. Mallick, R. Goel, B. Kerbl, M. Steinberger, F. V. Carrasco, and F. De La Torre, "Taming 3dgs: High-quality radiance fields with limited resources," in *SIGGRAPH Asia 2024 Conference Papers*, 2024, pp. 1–11.