2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Semantic-Guided Progressive Object Removal with Gaussian Splatting

Xianliang Huang[1], Chen Xiao[3], Yuanxiang Ni[2], Guanming Liu[3],
Mingkai Liu [1], Dikai Fan [1], Xiao Liu[1], Hao Zhang[2,†]

***Abstract*— Removing unwanted objects from reconstructed 3D scenes is an important task in computer vision, supporting applications in AR/VR, robotics, and digital content creation. Existing methods typically complete the entire masked region in a single step and without effectively utilizing semantic information from other views, leading to difficulties in handling complex geometric details and textures. In this work, we propose a novel framework that integrates Semantic-guided Block Matching (SBM) and Region-Wise Progressive Refinement (RPR) for high-quality 3D object removal. First, we leverage DINOv2 to encode semantic guidance from multi-view observations, and the best match tokens are decoded to complete missing regions in the target view while maintaining cross-view consistency. Second, we introduce a RPR strategy that segments the target mask into multiple subregions and selectively refines those with poor visual quality. Our method is built upon Gaussian Splatting, ensuring high-fidelity scene reconstruction with efficient computation. Experimental results demonstrate that our approach outperforms existing Gaussian-based methods in terms of perceptual quality and coherence in 3D object removal.**

I. INTRODUCTION

Three-dimensional scene reconstruction and manipulation have been significantly advanced by Neural Radiance Fields [1] and 3D Gaussian Splatting [2] (3DGS), which enable photorealistic and efficient rendering for a wide range of applications, including virtual and augmented reality [3], robotics [4], [5], and autonomous driving [6]. A fundamental yet challenging task in this domain is 3D object removal, which involves eliminating unwanted objects from scenes and realistically completing the resulting holes. This task becomes particularly difficult when removing large objects in unbounded 360° environments, where it is necessary to leverage multi-view observations, hallucinate previously unseen content, and maintain both visual consistency and geometric plausibility across all views. Among recent advances, 3DGS has emerged as a powerful solution for real-time novel view synthesis and editable 3D scene reconstruction. Consequently, accurate and consistent object removal within Gaussian-based representations is becoming increasingly crucial for interactive editing and downstream scene understanding tasks.

Despite recent advancements [7], [8], [9], existing 3D object removal methods still face challenges when dealing with complex occlusions and fine-grained geometry. A key limitation lies in their insufficient exploitation of semantic information across multiple views. For instance, methods like SPIn-NeRF [7] perform inpainting primarily from 2D inputs while largely neglecting cross-view semantic consistency. As a result, they often produce inconsistent reconstructions, lacking geometric coherence in object regions. Other approaches [10], [11], [12] leverage generative priors through Score Distillation Sampling (SDS) [13] to optimize 3D representations. However, these approaches frequently yield visually inaccurate or overly smooth reconstructions, as they lack explicit geometric guidance and struggle to preserve high-frequency details. Furthermore, these techniques adopt a one-shot completion strategy for the entire masked region, which restricts their ability to iteratively refine suboptimal regions and correct localized artifacts.

To overcome the above limitations, we propose a novel framework that incorporates Semantic-guided Block Matching (SBM) across different views, enabling accurate recovery of missing structures and textures by aligning semantically relevant content. In addition, we introduce a High-frequency feature extraction module that provides auxiliary supervision to guide the generative process toward sharper inpainting results. Specifically, the High-frequency prior is injected into a pre-trained diffusion model, allowing it to synthesize plausible content conditioned on both global semantics and fine-grained visual details.

To further enhance the visual fidelity and consistency of the removal regions, we present a RPR strategy by segmenting the removal area into several blocks and selectively refining the regions with low perceptual fidelity. Specifically, the RPR strategy is driven by frequency-aware UNet, which focuses computational resources on challenging regions while avoiding redundant updates to already satisfactory areas. This target-level refinement significantly boosts the realism and coherence in the reconstructed 3D scene, making our approach more robust and perceptually superior to existing one-shot pipelines.

Overall, our pipeline enables precise and efficient 3D object removal by addressing the core problems of multi-view generative inpainting. We integrate SBM with the RPR strategy, allowing for better structural alignment and localized detail enhancement. Extensive experiments on various datasets, which include both forward-facing and unbounded scenes, demonstrate that our framework outperforms the existing baselines in terms of visual fidelity and geometric consistency. The key contributions of our framework are summarized as follows: (1) We propose a novel 3D object removal framework that combines 3D Gaussian Splatting with Score Distillation Sampling to produce initial inpainting results. A High-frequency feature extractor is adopted to

[1]PICO, ByteDance Inc., [2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, [3] Fudan University
[†]For correspondence and questions: h.zhang10@siat.ac.cn

guide the pre-trained diffusion model in repainting sharper regions. (2) A semantic-guided block matching is introduced to enhance missing structures and textures by explicitly aligning semantically relevant content from other perspectives. (3) We propose a RPR strategy that divides the inpainting region into multiple subregions and selectively refines low-quality subregions, improving local fidelity and overall coherence of the removal region. (4) Extensive experiments are conducted to validate the effectiveness of our method, demonstrating superior performance over existing NeRF-based and Gaussian-based inpainting baselines in visual quality and geometric accuracy.

II. Related Work

A. Diffusion Model for Conditional Generation

Diffusion models have emerged as a powerful framework for conditional content generation, with applications spanning image restoration [14], editing [15], and 3D scene understanding [16], [17]. Early models such as DDPM [16] and Stable Diffusion [18] demonstrated strong performance in high-fidelity image generation, while latent diffusion greatly improved efficiency and enabled conditioning via text or spatial priors [19]. To enable fine-grained control over the generation process, a wide variety of conditioning techniques have been proposed. Prompt-based editing [20], CLIP-guided modulation [21], and attention modification have been used to direct generation semantically. Spatial control approaches such as SpaText [22] introduce mask-based conditioning for localized edits, while personalization methods like DreamBooth [23] and Textual Inversion [24] finetune diffusion models with limited user-provided data for instance-level control. For structure-aware editing tasks such as inpainting, models like SDEdit [25] and DDIM Inversion [26], [27] inject noise into images and iteratively denoise to allow semantic modifications while preserving global structure. These techniques enable faithful reconstruction and content preservation, making them particularly effective for realistic 3D object removal scenarios.

B. 3D Inpainting meets Gaussian Splatting

Early approaches for 3D inpainting primarily focused on geometric completion alone [28], [29]. Recently, several works have utilized CLIP-based optimization [30], [31], Score Distillation Sampling [13], [32], and Iterative Dataset Update strategies [33] to distill the 2D generative prior into 3D inpainting techniques. These techniques are commonly applied across different representations, including meshes [34], NeRFs [35], [36], and SDFs [37]. With the development of 3DGS [2], Gaussian-based 3D inpainting methods [11], [8], [38] have emerged as a promising path in filling missing regions within the GS framework, taking advantage of its superior rendering efficiency and high-quality reconstruction. GScream [39] enhances information flow between visible and occluded areas, promoting both geometric consistency and texture coherence in complex regions. MVInPainter [40] leverages reference-guided multi-view inpainting, significantly simplifying in-the-wild novel view synthesis by utilizing unmasked clues instead of explicit pose inputs. GaussianEditor [8] introduces Hierarchical Gaussian Splatting and Gaussian semantic tracing to improve the precision and stability of diffusion-guided reconstructions. Infusion [11] performs 3D inpainting by learning depth completion from diffusion priors, while Gaussian Grouping [38] often suffers from inaccurate unseen region segmentation during mask generation, adversely affecting inpainting quality. In this work, we further advance 3D inpainting within the Gaussian Splatting framework by improving both visual fidelity and computational efficiency.

III. Preliminary

A. Gaussian Splatting

We use 3DGS [2] to model 3D scenes, representing them as anisotropic Gaussian primitives. Each primitive $G_i = (x_i, r_i, \alpha_i, c_i)$ includes a center $x_i \in \mathbb{R}^3$, rotation $r_i \in \mathbb{R}^3$, opacity $\alpha_i \in [0, 1]$, and RGB color $c_i \in \mathbb{R}^3$ via spherical harmonics. The Gaussian density is:

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

with covariance $\Sigma = \mathbf{RSS}^T\mathbf{R}^T$, where $\mathbf{R}$ is rotation and $\mathbf{S}$ is scaling. For rendering, 3D Gaussians are projected to 2D using the camera matrix, processed in parallel as pixel blocks [41]. The final color $\hat{\mathbf{C}}$ is computed via alpha blending [42]:

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} c_i \cdot \alpha_i \cdot \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where c_i and α_i are the color and opacity of the i-th Gaussian, ordered by depth.

B. 2D Diffusion Models

Diffusion models operate via two key processes: a forward diffusion step $q(\mathbf{z}_\tau|\mathbf{z}_0)$ that incrementally corrupts a data sample $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{z})$ with Gaussian noise, and a learned reverse process that gradually denoises a pure Gaussian sample $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ back to a clean image. At a particular timestep $\tau \in [0, T]$, the noisy latent $\mathbf{z}_\tau$ can be expressed as:

$$\mathbf{z}_\tau = \sqrt{\bar{\alpha}_\tau}\,\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_\tau}\,\boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ denotes the injected noise and $\bar{\alpha}_\tau$ is the cumulative product of noise schedule coefficients. This forward process transitions into the reverse diffusion phase, where a neural network $\boldsymbol{\epsilon}_\theta$ learns to predict the noise $\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\theta(\mathbf{z}_\tau, \tau, \mathbf{c})$, starting from a pure Gaussian sample $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$. The training objective minimizes the denoising error

$$\mathcal{L}_{\text{diff}} = \left\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\right\|^2, \tag{4}$$

and using this predicted noise, the denoised latent at the previous timestep is computed as $\mathbf{z}_{\tau-1} = \mathbf{z}_\tau - \hat{\boldsymbol{\epsilon}}$ (with scaling omitted for brevity). This procedure is repeated iteratively to reconstruct a clean sample $\mathbf{z}_0$. For latent diffusion models such as Stable Diffusion [18], $\mathbf{z}$ resides in a compressed latent space where $\mathbf{z} = \mathcal{E}(\mathbf{x})$ maps pixels to latents and a decoder $\mathcal{D}(\cdot)$ generates the final image $\mathbf{x}$.

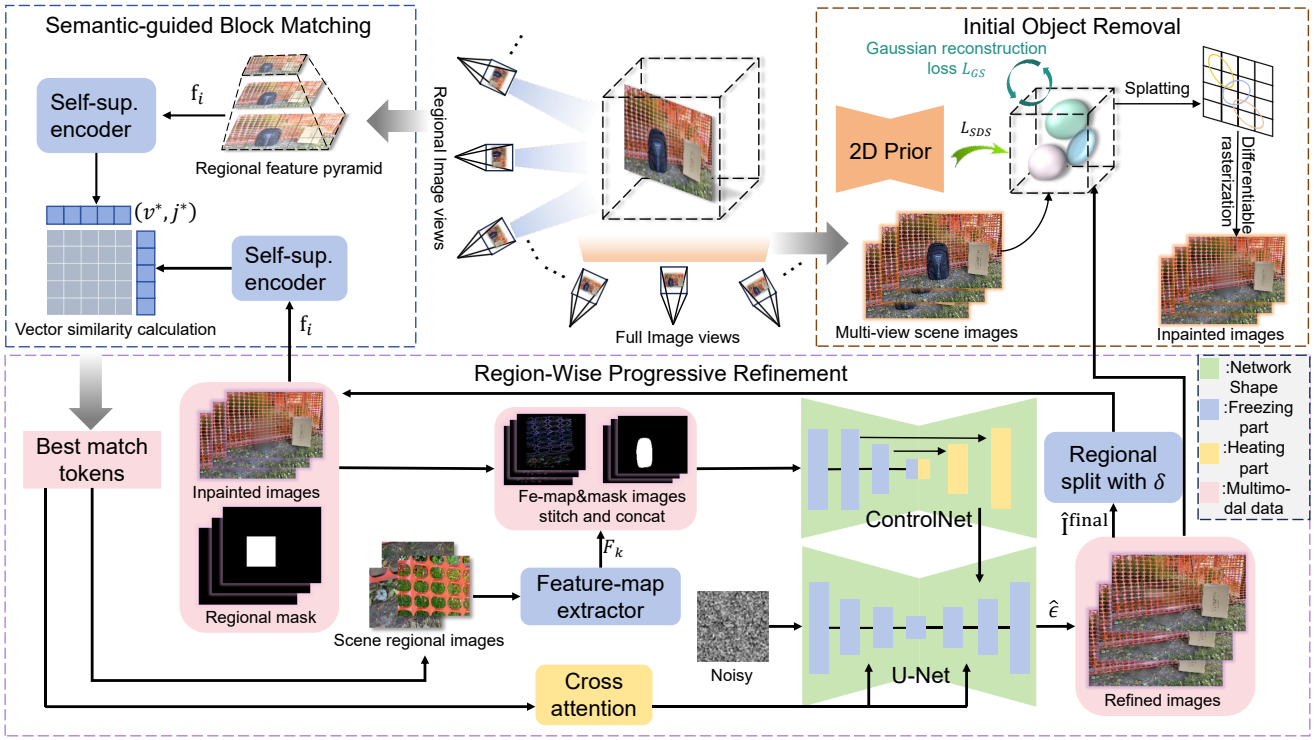

Fig. 1. **Overview of our pipeline.** The key insight of our method is to iteratively refine all region blocks in a few steps, guided by the high-frequency information from the semantic-guided block. Specifically, the semantic-guided block matching leverages multi-view observations to guide the completion of missing regions with content-aligned semantics. Next, the high-frequency feature extractor provides auxiliary supervision signals from image components to enhance detail preservation during diffusion-based generation. Finally, we utilize a RPR strategy that decomposes the target mask into subregions and iteratively refines visually suboptimal areas for improved local and global coherence.

C. Diffusion Priors for 3D Generation

Beyond 2D synthesis, diffusion models serve as powerful generative priors for optimizing implicit 3D representations like NeRF [1] and 3DGS [2], unlike GANs or deterministic models [43]. In this work, Score Distillation Sampling (SDS) [13] is utilized to update Gaussian primitives G_θ such that the rendered image $\mathbf{x}_r$ aligns with the distribution of real images under the learned diffusion model. By adding noise $\boldsymbol{\epsilon}$ to obtain $\mathbf{z}_\tau$, and minimizing:

$$\mathcal{L}_{\text{SDS}} = \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_\tau, \tau, \mathbf{c})\|^2 . \tag{5}$$

We backpropagate this gradient between the predicted and actual noise to refine θ, guiding the 3DGS to produce photorealistic renderings consistent with 2D diffusion prior.

IV. Methodology

In this section, we detail the design of SBM, the High-frequency Feature Extractor and RPR in Fig. 1. Our goal is to achieve high-fidelity 3D object removal with cross-view consistency and fine-grained visual quality.

A. Multi-view Representation and Initialization

Formally, 3D scenes can be represented by 3D Gaussians θ, given a collection of multi-view images $\mathcal{I} = \{I_i\}_{i=1}^n$, accompanied by respective camera poses $\{\pi_i\}_{i=1}^n$. To initialize the 3D Gaussian set, we first estimate a sparse point cloud from Structure-from-Motion (SfM) [44]. The initial positions $\boldsymbol{x_i}$ of the Gaussians are seeded from these points, while their scales and opacities are set to default or random values. To increase scene fidelity, we perform adaptive densification through primitive splitting and cloning strategies, dynamically increasing the density of Gaussians in regions with high reconstruction error.

The 2D object masks $\mathcal{M} = \{M_i\}_{i=1}^n$ are utilized to indicate the target 3D object. We remove the Gaussians in the masked region according to object masks and replace them with the same amount of randomly initialized Gaussians. Then, the Score Distillation Sampling [13] loss is calculated by distilling knowledge from the inpainting backbone [18] to improve the rendering result of 3DGS with multi-step noise prediction. The initial inpainted images $\hat{\mathcal{I}} = \{\hat{I}_i\}_{i=1}^n$ are formulated as:

$$\hat{I}_k = G_\theta(I_k, \pi_k), \tag{6}$$

where G_θ denotes the 3DGS model, I_k is the input view, and π_k is its corresponding camera pose.

To update the parameters θ, Gaussian noise ϵ is added to the rendered images $\hat{\mathbf{I}}$ and the following SDS loss is applied, using predicted noise $\hat{\epsilon}_\psi$ from the latent diffusion model ψ:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\psi, \hat{\mathbf{I}}) = \mathbb{E}_{\epsilon,t}\left[w(t)\big(\hat{\epsilon}_\psi(\hat{\mathbf{I}}_t; y, t) - \epsilon\big) \frac{\partial \hat{\mathbf{I}}}{\partial \theta} \right], \tag{7}$$

where y would be the input text and t is the level of noise.

B. Semantic-Guided Block Matching

To fully utilize cross-view contextual cues, we propose a *semantic-guided block matching strategy* to optimize the ambiguity of target regions by retrieving semantically matched content across views and enforcing high-level alignment through diffusion-based generation. It serves as a key component to improve multi-view consistency and fine-grained realism in removal regions.

1) Block Decomposition and Semantic Extraction: The initial inpainted image $\hat{\mathbf{I}} \in \mathbb{R}^{H\times W\times 3}$ with its corresponding binary mask $\mathbf{M} \in \{0,1\}^{H\times W}$ indicating the removal object, we first divide both $\mathbf{I}$ and $\mathbf{M}$ into N non-overlapping square blocks $\mathbf{B}_i$ with size $s \times s$. Only blocks sufficiently occluded, satisfying $\mathbf{B}_i \cap \mathbf{M} > \tau$, are selected as target blocks for retrieving. The occlusion ratio threshold τ is empirically set to 0.1 in our experiments. Specifically, a smaller τ increases the number of selected blocks, capturing more occluded areas but reducing performance due to higher computational demands. Conversely, a larger τ selects fewer blocks, improving performance but potentially lowering accuracy by ignoring moderately occluded blocks.

For each block $\mathbf{B}_i$, we extract high-level semantic embeddings using a pre-trained DINOv2 encoder $\mathcal{F}_{\text{DINOv2}}$:

$$\mathbf{f}_i = \mathcal{F}_{\text{DINOv2}}(\mathbf{B}_i \odot M) \in \mathbb{R}^d, \tag{8}$$

where d is the feature dimension.

2) Cross-View Retrieval: The original images $\{\mathbf{I}^v\}_{v=1}^V$ partitioned into blocks $\{\mathbf{B}_j^v\}$, we compute semantic features $\mathbf{f}_j^v = \mathcal{F}_{\text{sem}}(\mathbf{B}_j^v)$ for all visible blocks. For each target block $\mathbf{B}_i^t$, we perform semantic matching by identifying the best matching source block $\mathbf{B}_{j^*}^{v^*}$ via cosine similarity:

$$(v^*, j^*) = \arg\max_{v,j} \text{sim}(\mathbf{f}_i^t, \mathbf{f}_j^v), \tag{9}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity between two feature vectors.

3) Semantic-guided Inpainting: The core idea of this part is to encode the selected blocks that are located with relevant features from other viewpoints and inject these features into a pre-trained diffusion model to refine the inpainted images. Once matched, the selected source block $\mathbf{B}_{j^*}^{v^*}$ is used as a semantic reference to guide the denoising process of the diffusion model [18] for generating the corresponding object region. Specifically, the matched source block $\mathbf{B}_{j^*}^{v^*}$ is projected into the latent space, where probabilistic sampling is performed using a UNet-based denoising network. To incorporate semantic priors, we replace the original text embedding $\mathbf{c}$ with a semantic-aware token $\mathbf{c}_i$, derived from the matched reference block. These tokens are injected into each layer of the UNet via cross-attention mechanisms, enabling fine-grained, semantic-driven conditioning.

Formally, let $\mathbf{z}_i^t$ denote the latent of the occluded target block at timestep T. The predicted noise is computed as:

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\theta(\mathbf{z}_i^t, T, \mathbf{c}_i). \tag{10}$$

This guidance mechanism ensures that the generated content within each masked block is semantically aligned with the most relevant information from other views, resulting in more coherent and visually plausible inpainting results.

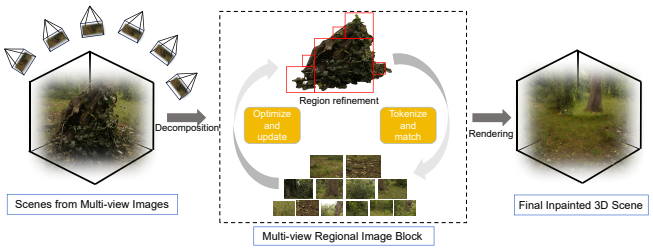

Fig. 2. Given a 3D scene, our method produces accurate object removal with region refinement and outputs a visually consistent 3D scene.

C. Region-Wise Progressive Refinement Strategy

Although semantic alignment guides the initial inpainting, the results may still exhibit blurry textures or geometric inconsistencies, especially in complex regions. To address this, we introduce an RPR strategy, which selectively refines low-quality blocks using a frequency-aware diffusion generator. The RPR process is illustrated in Fig. 2.

1) High-Frequency Feature Extraction: We first identify low-quality inpainted regions based on their lack of High-frequency details. Specifically, we adopt our High-frequency Feature Extractor, which is inspired by edge detection and combines horizontal and vertical gradient responses of the grayscale image. Given a reconstructed RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, we first convert it to grayscale $\mathbf{I}_{\text{gray}}$, then apply high-pass filtering using horizontal and vertical Sobel kernels [45] $\mathbf{K}_h$ and $\mathbf{K}_v$, respectively. The high-frequency response $\mathbf{F}_k$ is computed as:

$$\mathbf{F}_k = (\mathbf{I}_{\text{gray}} \otimes \mathbf{K}_h + \mathbf{I}_{\text{gray}} \otimes \mathbf{K}_v) \odot \mathbf{I} \odot \mathbf{M}_{\text{erode}}, \tag{11}$$

where $\otimes$ denotes convolution and $\odot$ is the Hadamard product. $\mathbf{M}_{\text{erode}}$ is an eroded version of the original mask to exclude noisy boundaries. A block is marked for refinement if its average high-frequency magnitude is below a predefined threshold δ. A block is labeled "low-quality" if its high-frequency amplitude is below 15% of the surrounding area, a threshold chosen empirically based on frequency statistics across scenes.

2) Re-Inpainting with Frequency Guidance: For each $\mathbf{B}_k \in \mathcal{B}_{\text{low}}$, we re-infer the block using a diffusion process based on the pre-trained UNet. Unlike the initial inpainting step, here we concat both semantic token guidance and a frequency-aware embedding to better preserve high-frequency textures.

Given a latent representation $\mathbf{z}_k$ and its semantic token $\mathbf{c}_k$ obtained via DINOv2 matching, we also encode the block's high-frequency map $\mathbf{F}_k$ into a compact feature vector $\mathbf{f}_k$ using a lightweight convolutional encoder $\mathcal{E}_{\text{freq}}$:

$$\mathbf{f}_k = \mathcal{E}_{\text{freq}}(\mathbf{F}_k), \quad \mathbf{F}_k = |\nabla_x \mathbf{I}_k| + |\nabla_y \mathbf{I}_k|, \tag{12}$$

where ∇_x and ∇_y denote Sobel-filter gradients in the horizontal and vertical directions, respectively.

We then condition the denoising UNet on both $\mathbf{c}_k$ and $\mathbf{f}_k$:

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_{\theta'}(\mathbf{z}_k, T, \mathbf{c}_k, \mathbf{f}_k), \tag{13}$$

where $\epsilon_{\theta'}$ denotes the fine-tuned UNet, and T is the diffusion timestep. The frequency embedding $\mathbf{f}_k$ is injected through a cross-attention mechanism or MLP fusion block into intermediate UNet layers, enabling the network to recover sharper textures guided by both global semantics and local detail priors.

After generating the refined result $\hat{\mathbf{B}}_k$ for each low-quality block $\mathbf{B}_k \in \mathcal{B}_{\text{low}}$, we selectively replace the corresponding region in the initially completed image:

$$\hat{\mathbf{I}}^{\text{final}} = \text{Replace}(\hat{\mathbf{I}}, \mathbf{B}_k \leftarrow \hat{\mathbf{B}}_k), \quad \forall \mathbf{B}_k \in \mathcal{B}_{\text{low}}. \tag{14}$$

This progressive and targeted replacement strategy ensures that only perceptually suboptimal regions are refined, thereby reducing unnecessary computation.

D. Integration with Gaussian Splatting

Our object removal framework is ultimately integrated into a 3DGS pipeline to enable consistent reflection in the underlying 3D geometry and appearance. The Gaussian representation is updated by optimizing both color and spatial parameters across views, guided by a composite loss function that promotes both local realism and global multi-view consistency. The total loss combines the Gaussian reconstruction loss with SDS loss terms, weighted to balance multi-view coherence and visual consistency.

Gaussian reconstruction loss combines a mean absolute error (MAE) term and a differentiable SSIM loss [46]:

$$\mathcal{L}_{\text{GS}} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}, \tag{15}$$

where $\mathcal{L}_1 = \|\hat{\mathbf{I}} - \mathbf{I}\|_1$ measures pixel-level differences, and $\mathcal{L}_{\text{D-SSIM}}$ emphasizes perceptual structure similarity. The balance factor λ is consistent with the original 3DGS.

The overall loss function used to supervise the optimization of the Gaussian parameters θ is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GS}} + \lambda_{\text{SDS}}\mathcal{L}_{\text{SDS}}, \tag{16}$$

where λ_{SDS} is a hyperparameter that control the influence of semantic-aware guidance. To maintain efficiency, we limit the number of Gaussians and employ a coarse-to-fine strategy, initializing with sparse Gaussians and progressively refining them. As a result, our method enables realistic object removal, high-fidelity novel view synthesis, and reliable downstream scene understanding within an efficient and lightweight 3DGS framework.

V. Experiments

In this section, we evaluate our proposed approach for 3D object removal in several benchmark datasets. We first describe our implementation details, then introduce the datasets, baselines, and evaluation metrics. We present both quantitative and qualitative results, followed by ablation studies to analyze the contribution of block-matching strategy and progressive refinement strategy.

A. Experimental Setup

1) Implementation Details: Our method is implemented on top of the official 3D Gaussian Splatting (3DGS) PyTorch CUDA extension. To ensure stable convergence, we apply Gaussian densification and pruning between iterations 100 and 2500. The 3D scene is initialized using sparse point clouds obtained via Structure-from-Motion (SfM), and progressively refined throughout the training. All experiments are conducted on an NVIDIA RTX 3090 GPU (24GB memory). We adopt the Adam optimizer with an initial learning rate of 1×10^{-4} and s is set to 15. Following the SDS strategy from DreamFusion [13], we perform 1000 iterations of Score Distillation Sampling to obtain an initial diffusion prior. During this process, latent features are perturbed using a pre-defined noise schedule, and a frozen conditional Imagen model predicts the added noise to guide parameter updates. We employ a pre-trained Variational Autoencoder to encode both the original RGB image and its binary mask into a 4-channel latent space, serving as the input to the diffusion model. Stable Diffusion v2.1 [18] is adopted as the base generator throughout all experiments.

2) Datasets and Baselines: To comprehensively evaluate the effectiveness of our proposed method, we conduct experiments on both public datasets [47], [48], [7] and self-captured datasets, covering a wide range of scene complexities and camera configurations. We compare our method against several state-of-the-art approaches for 3D scene object removal, including both NeRF-based and Gaussian-based methods.

3) Evaluation Metrics: To quantitatively evaluate the effectiveness of our method, we adopt Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46], and Learned Perceptual Image Patch Similarity (LPIPS) [49] to quantitatively assess synthesis results. These metrics are calculated solely for the pixels within removal regions identified in each view to evaluate the image quality.

B. Performance Comparisons

1) Results on Forward-facing Scenes: We compare our method against two representative 3D inpainting approaches: SPIn-NeRF [7] and GaussianEdit [8]. The evaluated scenes are performed on *Rednet* from SPIn-NeRF, *Orchids* from LLFF, *Chuyin* from self-captured dataset and *Kitchen* from Mip-NeRF. We assess the performance in terms of maintaining scene coherence and visual quality. In the left part of Fig. 3, SPIn-NeRF struggles with texture discontinuities and blurry regions in texture complex areas, such as the geometric structure of grass and net in *Rednet* and *Orchids*. For the right part scene of *Chuyin* and *Kitchen*, GaussianEdit demonstrates slightly better texture fidelity but often suffers from ghosting artifacts due to its reliance on the removal of Gaussian primitives. In contrast, our method leverages multi-view semantic alignment and high-frequency supervision to produce sharper and semantically plausible results. Tab. I reports PSNR, SSIM, and LPIPS metrics computed over masked inpainting regions. Our method consistently outperforms the SPIn-NeRF and GaussianEdit across all metrics,

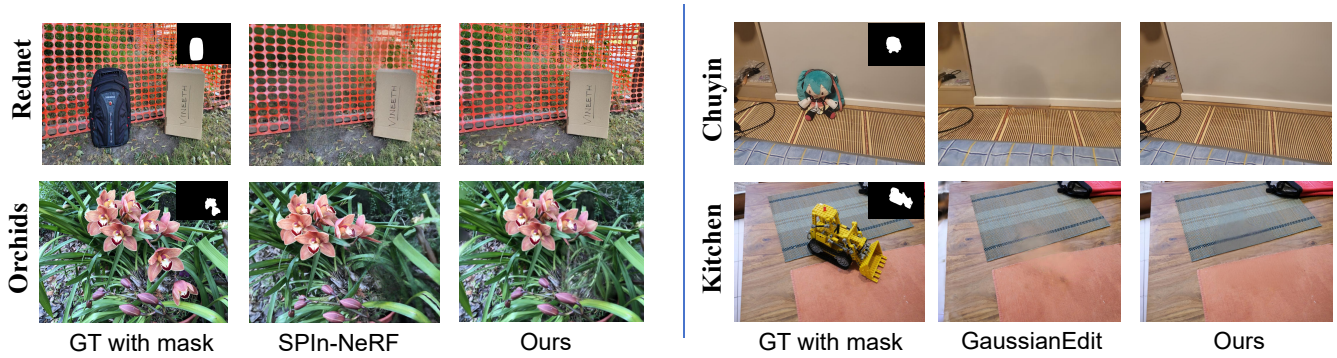

Fig. 3. **Qualitative results.** We compare our method with SPIn-NeRF and GaussianEdit in a consistent view.

TABLE I

QUANTITATIVE COMPARISONS WITH GAUSSIAN-BASED METHODS ON SPIN-NERF, SELF-CAPTURED, AND MIP-NERF 360 DATASETS.

Method	*SPIn-NeRF dataset*			*Self-captured dataset*			*Mip-NeRF 360 dataset*		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SPIn-NeRF	26.8	0.901	0.176	21.7	0.824	0.231	23.0	0.860	0.204
GaussianEdit	22.5	0.841	0.217	25.2	0.879	0.192	24.9	0.871	0.208
GaussianGroup	21.7	0.815	0.174	19.1	0.693	0.188	20.8	0.684	0.193
MVInpainter	27.5	0.915	0.160	26.1	0.891	0.179	25.8	0.884	0.191
InFusion	27.1	0.912	0.164	26.4	0.893	0.174	25.5	0.881	0.186
Ours	**28.7**	**0.929**	**0.139**	**29.6**	**0.936**	**0.131**	**27.4**	**0.899**	**0.161**

indicating superior reconstruction quality and perceptual realism.

2) Comparison with Gaussian-based Methods: We further evaluate our method against recent state-of-the-art Gaussian-based methods, including MVInpainter [40], GaussianGroup [38], and InFusion [11]. These methods leverage 3D Gaussian Splatting representations to enable photorealistic object removal. Our comparisons focus on multi-view consistency across challenging scenes. As shown in Fig. 4, our method delivers cleaner and more semantically aligned inpainting in the *Bear* and *Bicycle* scenes. MVInpainter produces visually smooth results but struggles with detailed restoration and sometimes introduces over-smoothed patches. GaussianGroup introduces semantically irrelevant textures or severely inconsistent shading, particularly in highly cluttered backgrounds. InFusion achieves promising results guided by 2D diffusion priors but lacks precise structural alignment in object removal areas. In contrast, our method benefits from semantic-aware token guidance and progressive region-wise refinement, achieving visually realistic and structurally coherent results across views. Tab. I presents a quantitative comparison of our proposed method with existing Gaussian-based approaches across PSNR, SSIM, and LPIPS metrics computed on masked regions. Our approach achieves the highest performance across all evaluation metrics, demonstrating its effectiveness in 3D object removal. Specifically, for the *Rednet* and *Chuyin* scenes, our method achieves a peak PSNR of 28.7, significantly outperforming other baselines. In the more challenging *Kitchen* scene, our approach continues to outperform all competitors, with a PSNR of 27.4, SSIM of 0.899, and LPIPS of 0.162—outclassing the previous best method InFusion. Overall, our method consistently surpasses other baselines across all scenes and metrics, highlighting its strengths in perceptual similarity, structural fidelity, and view-consistent content completion.

TABLE II

EFFICIENCY COMPARISON. WE REPORT TRAINING TIME, INFERENCE TIME, AND GPU MEMORY CONSUMPTION.

Method	**Training Time**	**Inference Time**	**GPU Memory**
SPIn-NeRF	20h	5h	-
GaussianEditor	2h	10min	12GB
GaussianGroup	2.6h	20min	15GB
MVInpainter	120h	5min	20GB
InFusion	24h	2min	20GB
Ours	7.9h	13min	10GB

3) Comparison on Runtime and Memory: We compare the training time, inference time, and GPU memory usage of various methods on our self-captured dataset using an NVIDIA RTX 3090 GPU. As shown in Tab. II, our method achieves a favorable trade-off between performance and efficiency. Specifically, MVInpainter and InFusion require 20GB memory and longer training durations , while GaussianEditor and GaussianGroup offer faster training but with limited accuracy. In contrast, our method requires only 7.9 hours of training and 13 minutes of inference time, with a modest GPU memory footprint of 10GB, outperforming most baselines in efficiency without compromising quality. These results demonstrate the practicality of our approach for real-world deployment scenarios, where both runtime and memory are critical considerations.

C. Ablation Studies

SBM module enhances geometric alignment across views by leveraging semantic features to guide correspondence

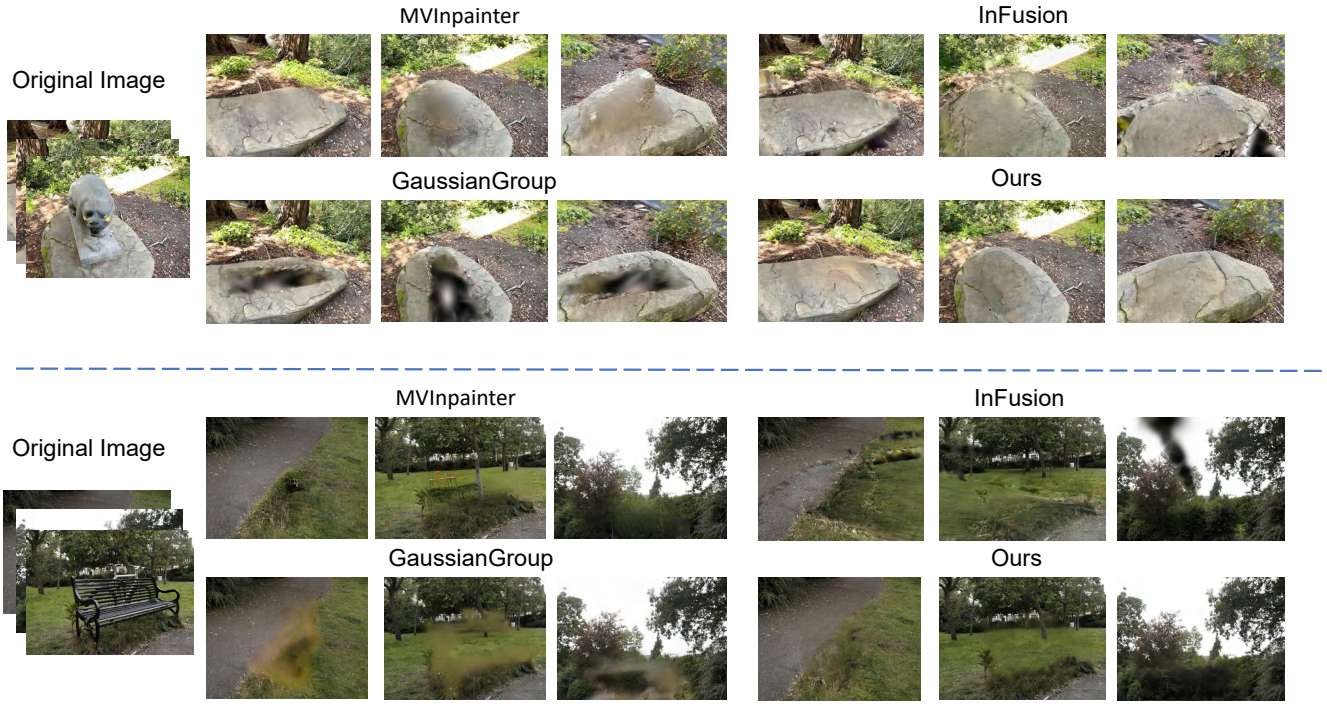

Fig. 4. **Qualitative results on *Bear* and *Bicycle* scenes.** Given the input images, we visualize the removal results of MVInpainter, GaussianGroup, InFusion and our method in a consistent view.

TABLE III

ABLATION STUDY. WE REPORT THE AVERAGE METRICS ON THE SELF-CAPTURED SCENE AND THE PUBLIC SCENE.

Method	*Self-captured scene*			*Public scene*		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o SBM	24.58	0.681	0.312	24.18	0.605	0.369
w/o RPR	29.37	0.761	0.143	26.43	0.841	0.198
Full Model	**30.15**	**0.886**	**0.126**	**28.62**	**0.892**	**0.151**

during multi-view inpainting. As shown in Tab. III, removing SBM causes a substantial degradation in performance, with the PSNR dropping by 5.57 and 4.44 on the self-captured and public scenes, respectively. SSIM also decreases notably, while LPIPS increases by 0.186 and 0.218, indicating a significant loss in perceptual quality. RPR module progressively refines inpainting results from coarse to fine across semantic regions, ensuring detail preservation and boundary smoothness. As shown in Tab. III, when RPR is removed, PSNR drops by 0.78 and 2.19, while LPIPS increases by 0.017 and 0.047 for the self-captured and public scenes, respectively. Although the SSIM degradation is less pronounced, the increased LPIPS reflects reduced perceptual realism. These results highlight the importance of both SBM and RPR, which improves boundary fidelity.

VI. CONCLUSION

We propose a novel framework for 3D object removal and scene completion that combines semantic-guided inpainting and RPR within 3DGS pipeline. By leveraging multi-view semantic features extracted via DINOv2, our method performs block-level matching to guide the completion of occluded regions while preserving cross-view consistency. A progressive refinement strategy further enhances visual quality by selectively updating low-fidelity regions based on perceptual feedback. Experiments show that our method outperforms existing Gaussian-based baselines in perceptual quality.

VII. ACKNOWLEDGEMENT

This project was supported by the National Natural Science Foundation of China under Grant No. 62472415.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] 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.

[3] H. Mao, Z. Xu, S. Wei, Y. Quan, N. Deng, and X. Yang, "Live-gs: Llm powers interactive vr by enhancing gaussian splatting," *arXiv preprint arXiv:2412.09176*, 2024.

[4] M. Liu, D. Fan, H. Que, H. Gao, X. Liu, S. Peng, M. Lin, S. Gu, R. Ye, W. Qiu, *et al.*, "Mace: Mixture-of-experts accelerated coordinate encoding for large-scale scene localization and rendering," *arXiv preprint arXiv:2510.14251*, 2025.

[5] R. Zhang, M. Zhang, J. Zhou, Z. Guo, X. Liu, Z. Xu, Z. Zhong, P. Yan, H. Luo, and X. Li, "Mind-v: Hierarchical video generation for long-horizon robotic manipulation with rl-based physical alignment," *arXiv preprint arXiv:2512.06628*, 2025.

[6] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 21 634–21 643.

[7] A. Mirzaei, T. Aumentado-Armstrong, K. G. Derpanis, J. Kelly, M. A. Brubaker, I. Gilitschenski, and A. Levinshtein, "Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields," *arXiv preprint arXiv:2211.12254*, 2022.

[8] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin, "Gaussianeditor: Swift and controllable 3d editing with gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 476–21 485.

[9] X. Huang, Z. Zhong, S. Chen, Y. Xu, J. Guan, and S. Zhou, "Nerf-mir: Toward high-quality restoration of masked images with neural radiance fields," *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[10] H. Chen, C. C. Loy, and X. Pan, "Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5344–5353.

[11] Z. Liu, H. Ouyang, Q. Wang, K. L. Cheng, J. Xiao, K. Zhu, N. Xue, Y. Liu, Y. Shen, and Y. Cao, "Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior," *arXiv preprint arXiv:2404.11613*, 2024.

[12] H. Zhong, C. Wang, J. Zhang, and J. Liao, "Generative object insertion in gaussian splatting with a multi-view diffusion model," *Visual Informatics*, p. 100238, 2025.

[13] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[14] S. Zhou, J. Pan, J. Shi, D. Chen, L. Qu, and J. Yang, "Seeing the unseen: A frequency prompt guided transformer for image restoration," in *European Conference on Computer Vision*. Springer, 2024, pp. 246–264.

[15] S. Zhou, D. Li, J. Pan, J. Zhou, J. Shi, and J. Yang, "Devil is in the uniformity: Exploring diverse learners within transformer for image restoration," in *Proc. Int. Conf. Comput. Vis.*, 2025.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[17] X. Huang, J. Gou, S. Chen, Z. Zhong, J. Guan, and S. Zhou, "Iddr-ngp: Incorporating detectors for distractors removal with instant neural radiance field," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1343–1351.

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[19] S. Zhou, J. Pan, and J. Yang, "Learning an adaptive sparse transformer for efficient image restoration," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 47, no. 11, pp. 10 344–10 360, 2025.

[20] Y. Lin, Y.-W. Chen, Y.-H. Tsai, L. Jiang, and M.-H. Yang, "Text-driven image editing via learnable regions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7059–7068.

[21] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023.

[22] O. Avrahami, T. Hayes, O. Gafni, S. Gupta, Y. Taigman, D. Parikh, D. Lischinski, O. Fried, and X. Yin, "Spatext: Spatio-textual representation for controllable image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 370–18 380.

[23] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 500–22 510.

[24] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618*, 2022.

[25] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[26] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[27] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[28] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[29] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3d generative adversarial network and recurrent convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2298–2306.

[30] W. Gao, N. Aigerman, T. Groueix, V. Kim, and R. Hanocka, "Textdeformer: Geometry manipulation using text guidance," in *ACM SIGGRAPH 2023 conference proceedings*, 2023, pp. 1–11.

[31] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.

[32] J. Zhuang, C. Wang, L. Lin, L. Liu, and G. Li, "Dreameditor: Text-driven 3d scene editing with neural fields," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–10.

[33] X. Huang, S. Chen, Z. Zhong, J. Gou, J. Guan, and S. Zhou, "Hi-nerf: Hybridizing 2d inpainting with neural radiance fields for 3d scene inpainting," in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 2855–2871.

[34] O. Michel, R. Bar-On, R. Liu, S. Benaim, and R. Hanocka, "Text2mesh: Text-driven neural stylization for meshes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 492–13 502.

[35] E. Weber, A. Holynski, V. Jampani, S. Saxena, N. Snavely, A. Kar, and A. Kanazawa, "Nerfiller: Completing scenes via generative 3d inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 731–20 741.

[36] A. Barda, M. Gadelha, V. G. Kim, N. Aigerman, A. H. Bermano, and T. Groueix, "Instant3dit: Multiview inpainting for fast editing of 3d objects," *arXiv preprint arXiv:2412.00518*, 2024.

[37] X. Cheng, T. Yang, J. Wang, Y. Li, L. Zhang, J. Zhang, and L. Yuan, "Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts," *arXiv preprint arXiv:2310.11784*, 2023.

[38] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian grouping: Segment and edit anything in 3d scenes," in *European Conference on Computer Vision*. Springer, 2025, pp. 162–179.

[39] Y. Wang, Q. Wu, G. Zhang, and D. Xu, "Learning 3d geometry and feature consistent gaussian splatting for object removal," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–17.

[40] C. Cao, C. Yu, F. Wang, X. Xue, and Y. Fu, "Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing," *arXiv preprint arXiv:2408.08000*, 2024.

[41] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 29–538.

[42] N. Max, "Optical models for direct volume rendering," *TVCG*, vol. 1, no. 2, pp. 99–108, 1995.

[43] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2149–2159.

[44] 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.

[45] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

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

[47] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, 2019.

[48] 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.

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