Learning 3D Geometry and Feature Consistent Gaussian Splatting for Object Removal

Yuxin Wang[1], Qianyi Wu[2], Guofeng Zhang[3], and Dan Xu[1]✉

[1] Hong Kong University of Science and Technology
[2] Monash University [3] Zhejiang University
ywangom@cse.ust.hk, qianyi.wu@monash.edu,
zhangguofeng@zju.edu.cn, danxu@cse.ust.hk

Abstract. This paper tackles the intricate challenge of object removal to update the radiance field using the 3D Gaussian Splatting. The main challenges of this task lie in the preservation of geometric consistency and the maintenance of texture coherence in the presence of the substantial discrete nature of Gaussian primitives. We introduce a robust framework specifically designed to overcome these obstacles. The key insight of our approach is the enhancement of information exchange among visible and invisible areas, facilitating content restoration in terms of both geometry and texture. Our methodology begins with optimizing the positioning of Gaussian primitives to improve geometric consistency across both removed and visible areas, guided by an online registration process informed by monocular depth estimation. Following this, we employ a novel feature propagation mechanism to bolster texture coherence, leveraging a cross-attention design that bridges sampling Gaussians from both uncertain and certain areas. This innovative approach significantly refines the texture coherence within the final radiance field. Extensive experiments validate that our method not only elevates the quality of novel view synthesis for scenes undergoing object removal but also showcases notable efficiency gains in training and rendering speeds. Project Page: https://w-ted.github.io/publications/gscream

Keywords: Object Removal · 3D Scene In-painting · Gaussian Splatting

1 Introduction

3D object removal from pre-captured scenes stands as a complex yet pivotal challenge in the realm of 3D vision, garnering significant attention in computer vision and graphics, particularly for its applications in virtual reality and content generation. This task extends beyond the scope of its 2D counterpart, *i.e.* image in-painting [3], which primarily focuses on *texture filling*. In 3D object removal, the intricacies of *geometry completion* become equally crucial, and the choice of 3D representation plays a significant role in the effectiveness of the model and rendering quality. [9–11, 13, 17, 31, 36].

✉ Corresponding author.

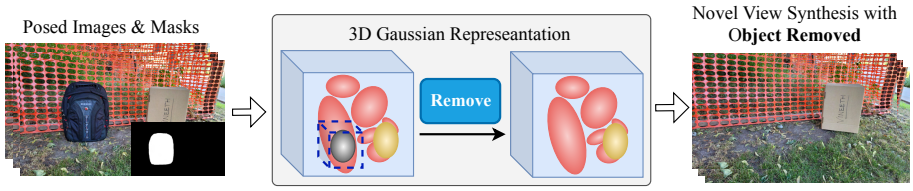

Fig. 1: Illustration of the Object Removal using 3D Gaussian Representations. Given a set of multi-view posed images and object masks, our goal is to learn a 3D consistent Gaussian representation modeling the scene with the object removed, which enables the consistent novel view synthesis without the specific object.

Recently, the radiance field representation has revolutionized the community due to the superior quality of scene representation and novel view synthesis. Among these, the Neural Radiance Field (NeRF) [21] has emerged as the groundbreaking implicit 3D representation approach, offering photo-realistic view synthesis quality. The high-quality rendering capabilities of NeRF have spurred further development in 3D object removal techniques based on it [17, 22, 23, 40, 41]. However, the intrinsic drawbacks of implicit representation, particularly its slow training and rendering speeds, pose severe limitations for practical applications based on object removal. For instance, it is highly expected that the system can quickly model the scene given any object mask condition for object removal, which enforces a straight requirement in terms of training efficiency. Another critical issue is that the object removal task relies on a flexible scene representation that can learn effective multi-view consistency to synthesize high-quality scene images with objects masked.

To effectively address the dual challenges of producing an enhanced radiance field for object removal, we introduce a pioneering strategy leveraging 3D Gaussian Splatting (3DGS) [15]. Unlike implicit representations, 3DGS explicitly models the 3D scene using tons of Gaussian primitives. This approach has demonstrated notable advances in rendering efficiency and quality, surpassing traditional NeRF-based methods [1, 24]. However, applying 3DGS to object removal presents unique challenges, primarily from two aspects: 1) *Geometry Accuracy*: The inherently discrete nature of a significant number of Gaussians can result in an inaccurate representation of the underlying geometry in the standard 3DGS model. This inaccuracy poses a considerable challenge in executing geometry completion and ensuring geometric consistency in the object removal areas within a 3D space. 2) *Texture Coherence*: Filling the region behind the removed object with consistent textures under the 3DGS framework represents another unexplored challenge. Achieving texture coherence across various viewing angles is essential, yet the methodologies to realize this goal within the 3DGS paradigm are currently underdeveloped.

The cornerstone of our approach lies in *augmenting the interaction between Gaussians in both the in-painted and visible regions, encompassing geometry and appearance enhancements*. Initially, to bolster geometric consistency across the

removal and visible areas, our method incorporates monocular depth estimation from multi-view images as a supplementary geometric constraint. This enhances the precision of 3D Gaussian Splatting (3DGS) placements. Employing a novel online depth alignment strategy, we refine the spatial arrangement of Gaussians within the removal area, ensuring improved alignment with adjacent regions. In terms of texture synthesis, our goal is to achieve a seamless blend between the visible and in-painted regions. Distinct from approaches tailored for implicit representations, which predominantly rely on image domain guidance for supervision, such as generating multi-view in-painted images [23, 40, 41] or simulating pseudo-view-dependent effects from NeRF [22], the explicit characteristic of Gaussian representations opens the door to innovative solutions. We introduce a novel method that facilitates feature interactions between Gaussian clusters from both visible and in-painted regions. This is achieved through a meticulously designed attention mechanism, which significantly improves the alignment of apparent and in-painted appearances. By sampling Gaussians positioned within both masked and unmasked areas, we refine their features via cross-attention in preparation for the final rendering. This self-interaction strategy capitalizes on the explicit nature of Gaussians to fine-tune the feature distribution in 3D spaces, culminating in enhanced coherence in the rendered outcomes. Furthermore, to mitigate the computational burden associated with directly manipulating millions of diminutive Gaussians, we implement a lightweight Gaussian Splatting architecture, Scaffold-GS [19], as our base model. Scaffold-GS introduces a novel paradigm that organizes Gaussians around anchor points, using the features associated with these anchors to decode attributes for the respective Gaussians. This approach not only streamlines the processing of Gaussian data but also significantly enhances the efficiency and effectiveness of our rendering process.

To the end, we propose a holistic solution coined **GScream** for object **re**moval from **G**aussian **S**platting while maintaining the geo**m**etry and fe**a**ture **c**onsistency. The contribution of our paper is threefold summarized below:

- We introduce GScream, a model that employs 3D Gaussian Splatting for object removal, specifically targeting and mitigating issues related to geometric inconsistencies and texture incoherence. This approach not only achieves significant efficiency but also ensures superior rendering quality when compared to traditional NeRF-based methods.
- To overcome the geometry inconsistency in the removal area, we incorporate multiview monocular depth estimation as an extra constraint. This aids in the precise optimization of Gaussian placements. Through an online depth alignment process, we enhance the geometric consistency between the removed area and the surrounding visible areas.
- Addressing the challenge of appearance incoherence, we exploit the explicit representation capability of 3DGS. We propose a unique feature regularization strategy that fosters improved interaction between Gaussian clusters in both the in-painted and visible sections of the scene. This method ensures coherence and elevates the appearance quality of the final rendered images.

2 Related Works

2.1 Radiance Field for Novel View Synthesis

Photo realistic view synthesis is a long-standing problem in computer vision and computer graphics [16,18,30,32]. Recently, the radiance field approaches [21] revolutionized this task by only capturing scenes with multiple photos and brought the reconstruction quality to a new level with the help of neural implicit representations [25, 33] and effective positional encoding [21, 34]. While the implicit representation benefits the optimization, the extensive queries of the network along the ray for rendering make the entire rendering speed costly and time-consuming [1,2]. Recently, there have been several attempts to facilitate the rendering speeds [7,15,24,26]. Among all of them, the 3D Gaussian splitting (3DGS) representation [15,19] stands as the most representative one which reaches a real-time rendering with state-of-the-art visual quality. 3DGS represents the radiance field as a collection of learnable 3D Gaussian. Each Gaussian blob includes information describing its 3D position, opacity, anisotropic covariance, and color features. With the dedicated design of a tiled-based splatting solution for training, the rendering of 3DGS is real-time with high quality. However, 3DGS is only proposed for novel view synthesis. It remains challenging to tame it if we want to remove objects from the pre-captured images [5].

2.2 Object Removal from Radiance Field

As the fidelity of 3D scene reconstruction advances, the ability to edit pre-captured 3D scenes becomes increasingly vital. Object removal, a key application in content generation, has garnered significant interest, particularly within the realm of radiance field representation. Several methods have been proposed to tackle this challenge [17, 22, 23, 40, 41]. For instance, NeRF-in [17] and SPIn-NeRF [23] utilize 2D in-painting models to fill gaps in training views and rendered depths. However, these approaches often result in inconsistent in-painted images across different views, leading to "ghost" effects in the removed object regions. View-Subtitude [22] offers an alternative by in-painting a single reference image and designing depth-guided warping and bilateral filtering techniques to guide the generation in other views. Despite these innovations, the underlying issue of slow training and rendering speed persists in these NeRF-based methods. The recent 3DGS-based general editing framework, GaussianEditor [6], includes the operation of deleting objects. However, despite its faster editing efficiency compared to NeRF-based methods, it still lacks specific constraints in the 3D domain. For the object removal task, purely fitting the 2D priors provided by the image in-painting model can also result in discontinuities in the 3D domain.

In response to these limitations, our work proposes a novel solution utilizing the 3D Gaussian Splatting (3DGS) [15] representation to achieve efficient object removal. The 3DGS method offers a more rapid training and rendering process, making it a suitable candidate for this application. However, 3DGS, in its standard form, primarily focuses on RGB reconstruction loss, leading to

less accurate underlying geometry for complex scenes. To make it suitable for recovering a scene without a selected object, we approach the problem in two stages: depth completion followed by texture propagation. We first enhance the geometric accuracy of 3DGS using monocular depth supervision. With a more refined geometric base, we then employ this improved structure to propagate 3D information outside the in-painted region to refine the texture in the in-painted region. These processes ensure not only the efficient removal of objects but also the maintenance of the scene's visual and geometric integrity.

3 The Propose Framework: GScream

As illustrated in Fig. 1, given N multi-view posed images $\{I_i|i = 0, \ldots, N\}$ of a static real-world scene with the corresponding binary masks specifying the object $\{M_i|i = 0, \ldots, N\}$. The object mask M_i is a binary mask with the object region set as 1 and the background set as 0. We assume these masks are provided for training, which can be obtained trivially by video segmentation [8, 17] or a straightforward 3D annotation [4, 41]. Our goal is to learn a 3D Gaussian representation to model the real-world scene with the object removed. To address this problem, we propose a novel framework named GScream, and the overview of it can be found in Fig. 2. First, we select one view as the reference view and perform the 2D in-painting [27,28] to complete the content by the corresponding mask. Without loss of generality, we denote the selected view with index 0 and the in-painted image as $\bar{I}$. We use the in-painted one single image to train the final 3DGS. The overview of our proposed GScream is shown in Fig. 2.

The organization of this section is presented as follows: we will introduce the preliminary about 3D Gaussian Splatting and its variants in Sec. 3.1, and then dive into the details about the core design of our framework in terms of geometry consistency and appearance coherence in the following subsection.

3.1 Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting We use the 3D Gaussian representation as our underlying modeling structure. Each Gaussian blob has the following attributes: 3D coordinates $\boldsymbol{\mu}$, scale matrix S, rotation matrix R, color features $\boldsymbol{c}$, and its opacity. With these attributes, the Gaussians are defined by the covariance matrix $\Sigma = RSS^TR^T$ centered at point $\boldsymbol{\mu}$:

$$G(x) = \exp^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \tag{1}$$

This Gaussian is multiplied by the opacity in the rendering process. By projecting the covariance onto the 2D plane following Zwicker *et al.* [44], we can obtain the projected Gaussian and adopt the volume rendering (α-blending) [20] to render the color in the image plane.

$$\hat{C} = \sum_{k=1}^{K} \boldsymbol{c}_k \alpha_k \prod_{j=1}^{k-1}(1-\alpha_j), \tag{2}$$

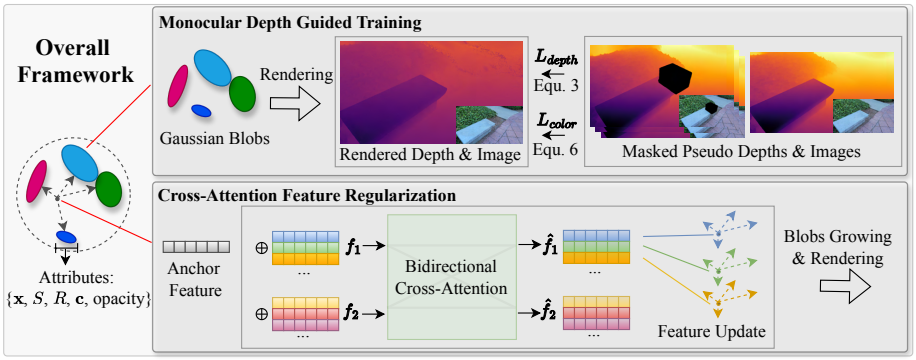

Fig. 2: Illustration of our GScream framework. It consists of two novel components, which are monocular depth guided training and cross-attention feature regularization. Our 3D Gaussian splatting (3DGS) representation is initialized by the 3D SfM points and supervised by both images and multi-view monocular pseudo depth estimation. The additional depth losses help refine the geometry accuracy within the 3DGS framework. The following 3D feature regularization performs texture propagation to refine the appearance within the 3D in-painted region.

where K means the number of sampling points along the ray and α is given by evaluating the projected Gaussian of $G(x)$ and the corresponding opacity. The initial 3D coordinates of each 3D Gaussian blob are initialized as the coordinates of the SfM points [29]. All the attributes of Gaussians are optimized by the reconstruction loss of the image. More details can be found in [15].

Scaffold-GS While the sparse initial points are insufficient to model the entire scene, 3DGS designs a densification operation to split and merge Gaussians to capture more details. It will result in better rendering quality while leading to a heavy storage burden. Therefore, we adopt a lightweight Gaussian Splatting structure, Scaffold-GS [19]. The key contribution of it is to use anchors to generate new Gaussian attributes with several decoders. There will be a learnable feature embedding attached to each anchor, and all the new Gaussian attributes can be extracted from the anchor features. With the densification performed in the anchor points, the storage requirement of Scaffold-GS can be significantly reduced and benefit the modeling of the radiance field. We adopt it as our base model to propose an efficient object removal solution for Gaussian Splatting. More details can be found in [19].

3.2 Improve Geometry Consistency by Monocular Depth Guidance

One of the challenges to performing object removal upon 3DGS is the underlying geometry is too noisy [15], which further leads to difficulty when performing geometry completion for the removal region. To improve the quality, we propose to leverage the guidance from estimated monocular depth as extra supervision. Concretely, we use the depth estimation model [14] to extract the depth $\mathcal{D} =$

$\{D_i|i = 0, \ldots, N\}$ of each image from the in-painted image $\bar{I}$ and other views $\mathcal{I}$. Here D_0 corresponds to the estimated depth of $\bar{I}$.

Online Depth Alignment and Supervision The monocular depth estimation is not a metric depth [14]. Therefore, we propose an online depth alignment design to utilize the depth guidance. However, the inconsistent depth estimation of $\bar{I}$ and $\mathcal{I}$ brings an additional issue. The $\mathcal{I}$ contains the object that we want to remove, while $\bar{I}$ depicts an image without the object. Therefore, we propose the following weighted depth loss to solve this problem:

$$\mathcal{L}_{\text{depth}} = \frac{1}{HW} \sum M'_i \|(w\hat{D}_i + q) - D_i\|, \tag{3}$$

$$M'_i = \begin{cases} \lambda_1 M_i + \lambda_2(1 - M_i), & \text{if } i = 0 \\ \lambda_3(1 - M_i), & \text{if } i \neq 0 \end{cases}. \tag{4}$$

Where $\hat{D}_i$ is the rendered depth map from 3D Gaussian Splatting calculated similar to the Equ. 2 by:

$$\hat{D} = \sum_{k=1}^{K} t_k \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \tag{5}$$

where t_k is the z-coordinates of Gaussian mean μ_k in the corresponding camera coordinate system. The depth obtained from the monocular estimator D_i and the rendered depth $\hat{D}_i$ by the 3D Gaussians have different numerical scales, so we cannot directly calculate the loss. We employ an online alignment method to address the scale issue. Specifically, we align the rendered depth using scale and shift parameters, denoted as w and q, to match the scale of the monocular depth before calculating the loss. The scale and shift are obtained by solving a least-squares problem [14,42]. For the image in $\mathcal{I}$, we only use the points outside the mask region, and the resulting scale and shift are applied to the entire depth map. We design different weights to calculate the depth loss as in Equ. 4. With this design, the depth supervision is applied to the entire depth map D_0 for the reference view, while it is applied on the background region for other views' depth $\{D_i|i = 1, \ldots, N\}$. The $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters to balance the influence of mask weights. In addition to the point-wise L1 loss, we also enforce a total variation loss to enforce smoothness in the depth difference as follows:

$$\mathcal{L}_{\text{tv}} = \frac{1}{N} \sum M'_i \|\nabla((w\hat{D}_i + q) - D_i))\| \tag{6}$$

Color Loss Following [15,19], we also apply the multi-view color reconstruction loss for both the training:

$$\begin{aligned} \mathcal{L}_{\text{color}} = \frac{1}{HW} \sum M'_i((1 - \lambda_{ssim})\|\hat{C}_i - I_i\| \\ + \lambda_{ssim} SSIM(\hat{C}_i, I_i)), \end{aligned} \tag{7}$$

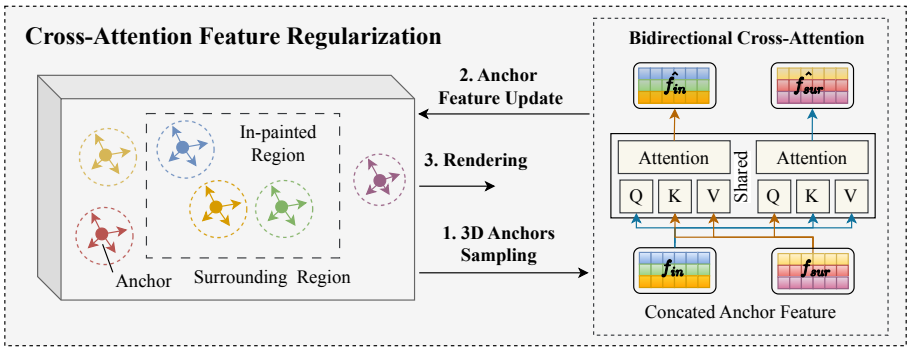

Fig. 3: Illustration of the Cross-attention Feature Regularization. Our regularization module consists of 3D Gaussian Sampling and a Bidirectional Cross-Attention Module, propagating the 3D feature from surrounding blobs to the in-painted region. As a complement to the 2D prior, the cross-attention mechanism enables the transmission of information among 3D Gaussian blobs, further ensuring the similarity of appearance between the in-painted region and its surroundings.

where $\hat{C}$ is the rendered image from 3DGS. Thanks to the rendering efficiency of 3DGS, we can render the entire image and perform a structural image reconstruction loss [39] SSIM to constrain the RGB image reconstruction. The overall training loss is the weighted sum of depth and color loss:

$$\mathcal{L}_{\text{total}} = \lambda_{depth}\mathcal{L}_{\text{depth}} + \lambda_{tv}\mathcal{L}_{\text{tv}} + \mathcal{L}_{\text{color}} \tag{8}$$

3.3 Cross-Attention Feature Regularization

Through monocular depth-guided training, we enhance the geometry of the 3D Gaussian representation. The following question is how we can refine the texture in the missing region from the surrounding environment.

Prior approaches in the realm of 3D object removal commonly employ a strategy that involves generating pseudo-RGB guidance to refresh the scene's information. This typically relies on leveraging multi-view in-painted images to update NeRF/3DGS models [6, 23, 41], or on producing view-dependent effects as a form of guidance [22]. However, these methods tend to be sensitive to the quality of the pseudo-ground truth and often overlook the intrinsic relationships between the in-painted regions and their visible counterparts.

The key insight of our model is *to propagate the accurate texture in the surrounding region into the in-painted region in a certain manner*. The explicit nature of 3DGS provides us the possibility to use the information from visible parts to update the content in the in-painted region. We expect this propagation can provide reliable information for the in-painted region in 3D space and ensure the propagated content is consistent across multiple viewpoints. Specifically, as

shown in Fig. 3, we perform a two-stage procedure to achieve texture propagation, *i.e.*, 3D anchors sampling, and subsequent bidirectional cross-attention.

3D Gaussian Sampling First, for each view i, we sample the patch that can simultaneously cover both the inside and the outside of the mask M_i. Then, we project the center coordinates of the 3D Gaussian anchors to the current view, to determine which anchor's 2D projection falls within the sampled 2D patch. After we identify the clusters of Gaussian anchors whose projections fall within the patch, we can easily categorize them into two groups based on whether their 2D projections are inside or outside the 2D mask. In this way, we sample 3D Gaussian anchors in both the in-painted and surrounding regions. Our goal is to sample 3D points in both the in-painted region and the surrounding region, as shown in the left part of Fig. 3. Although there are alternative sampling methods such as using depth for point back-projection, we believe that our approach based on 2D mask back-projection is sufficient to achieve our objectives.

Bidirectional Cross-Attention After obtaining the 3D Gaussian anchors from both regions, we perform bidirectional cross-attention between the two sets of Gaussian features to propagate information between the anchors. Specifically, we concatenate the two sets of Gaussian features as two tokens and take them as input to a bidirectional cross-attention structure following the classical definition [35] $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}$, where d_k is the token length.

The output of the cross-attention structure, which represents the updated features, is then assigned back to the corresponding Gaussian anchors. The bidirectional structure of the cross-attention is designed to facilitate bidirectional information propagation between the features inside and outside the in-painted regions. It can be seen as two sets of shared-parameter cross-attention modules, enabling information exchange between the two sets of features. As shown in Fig. 3, let us assume that the sampled tokens in the in-painted and surrounding regions are represented by the f_{in} and f_{sur}. After passing them through the cross-attention module, the updated features can be denoted as $\hat{f}_{in}$ and $\hat{f}_{sur}$:

$$\begin{aligned} \hat{f}_{in} &= \text{Attention}(\mathbf{Q} = f_{in}, \mathbf{K} = f_{sur}, \mathbf{V} = f_{sur}) \\ \hat{f}_{sur} &= \text{Attention}(\mathbf{Q} = f_{sur}, \mathbf{K} = f_{in}, \mathbf{V} = f_{in}) \end{aligned} \tag{9}$$

As shown in Fig. 2, when the sampled anchors complete the feature updates, all anchors undergo neural blobs growing and differentiable rendering as usual in [19]. The rendered depth map and image under the current viewpoint are then supervised by the total loss introduced in 8.

The 3D Gaussian sampling strategy together with the shared bidirectional cross-attention augments the anchor feature with similarity towards higher consistency. Through the gradients backpropagated to the anchors' features in the visible region, the similar anchors in the unpainted region can also be updated due to the attention mechanism. This design improves the consistency between the in-painted region and visible certain areas, which leads to better texture coherence in our experiments.

4 Experiments

4.1 Experimental Setup

Dataset Following previous methods, we conducted experiments for object removal on the SPIn-NeRF [23] and IBRNet dataset [37]. SPIn-NeRF dataset consists of 10 forward-facing in-the-wild scenes. Each scene has 100 multi-view images with annotated foreground object masks. To ensure a fair comparison, we directly utilize the camera parameters from the dataset instead of re-performing the sparse construction as [41]. IBRNet dataset is constructed for novel view synthesis, including selected scenes from existing datasets and 102 scenes collected by mobile phones. We use five captured scenes from IBRNet for experimentation.
Baselines We compare our methods with three recent baseline methods: SPIn-NeRF [23], OR-NeRF [41], and View-Sub [22]. We re-train and test the model using their open-source code to compare the first two baselines. We borrow the reported quantitative and qualitative results directly from the paper [22] due to the unavailable of open-source code.
Evaluation Metric We calculate the PSNR, SSIM [38], and LPIPS [43] on the full image and within the mask region. We also calculate the Frechet Inception Distance (FID) [12], which measures the distribution similarity between the generated and real images. We record the training time to evaluate the efficiency. Please note that the IBRNet scenes do not have ground truth images with objects removed, so quantitative metrics such as PSNR cannot be calculated. We only showcase partial quantitative results for these scenes.

4.2 Comparison with the State-of-the-art Methods

We present quantitative and qualitative comparisons between our method and three baseline methods in Tab. 1 and Fig. 4, respectively.
Quantitative Comparison As detailed in Tab. 1, our method either matches or surpasses SPIn-NeRF or OR-NeRF across all evaluated metrics. Notably, our approach yields superior similarity metrics, such as SSIM and LPIPS, suggesting that the images rendered by our method bear a closer resemblance to the ground truth in the test set. It is worth mentioning that SPIn-NeRF and OR-NeRF both utilize patch-based LPIPS loss in their optimization objective, which we did not employ. Despite this, our results still show an advantage in LPIPS, demonstrating the effectiveness of our method. Our method also performs better in terms of FID, indicating that the feature distribution of our rendered images is more consistent with real images without objects. Moreover, thanks to the efficiency of 3DGS representation rendering and optimization, our method achieves training times that are 1.5× and 4.0× faster than SPIn-NeRF and OR-NeRF, respectively. Regarding the View-sub method, due to the unavailability of its code, our comparison was limited to the masked LPIPS as reported in their paper, where our results were comparable. However, our method shows promise for an even more significant advantage in training efficiency.

Table 1: Quantitative comparison on novel view synthesis with the object removed. We compared our method with three baselines: SPIn-NeRF [23], OR-NeRF [41], and View-Sub [22]. '-' indicates the metrics are not reported by the authors in the paper. '*' indicates the metrics are directly borrowed from the corresponding paper.

Methods	PSNR ↑	masked PSNR ↑	SSIM ↑	masked SSIM ↑	LPIPS↓	masked LPIPS ↓	FID ↓	Training Time ↓
SPIn-NeRF [23]	20.18	15.80	0.46	**0.21**	0.47	0.58	58.78	∼ 3.0h
OR-NeRF [41]	20.32	15.74	0.54	**0.21**	0.35	0.56	38.69	∼ 6.0h
View-Sub [22]	-	-	-	-	-	**0.45***	-	-
GScream (Ours)	**20.49**	**15.84**	**0.58**	**0.21**	**0.28**	0.54	**36.72**	∼ **1.2h**

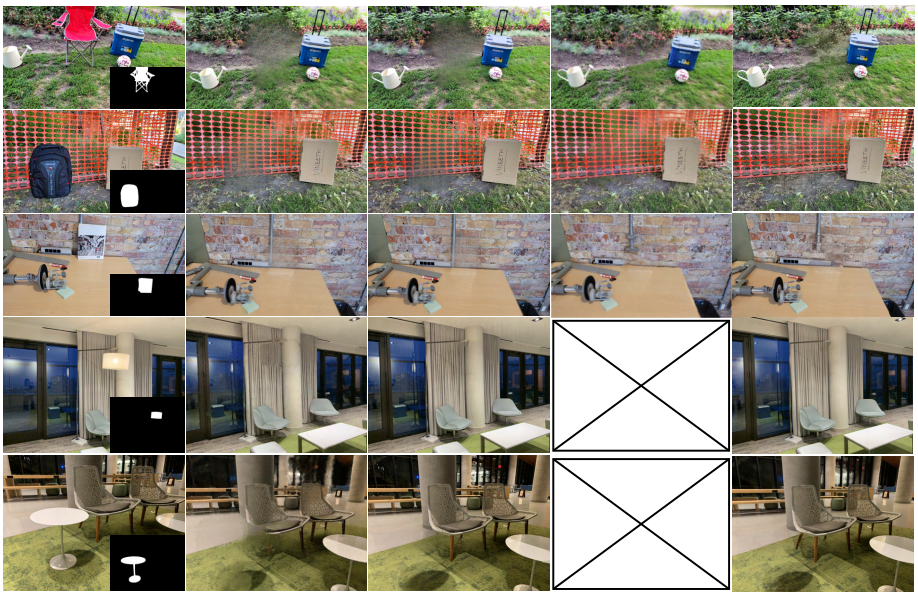

Fig. 4: Qualitative results compared with the most representative object-removal approaches. Illustration of the rendered qualitative images with object removed, compared with SPIn-NeRF [23], OR-NeRF [41], and View-Sub [22]. Our approach can synthesize high-quality images with natural removal effect.

Qualitative Comparison Fig. 4 presents a qualitative comparison across five different scenes. For the first three scenes, we select of the nearest neighboring viewpoints based on the View-sub paper, enabling a coherent rendering comparison among all approaches. Despite slight camera pose differences, we believe these variations are negligible concerning the overall assessment of rendering quality. The leftmost column shows randomly selected scene images and their corresponding mask. Upon analysis, it is evident that while all methods exhibit competence in completing mask regions across certain scenarios, such as the regular wall depicted in the third row and the simple textured fence in the fourth

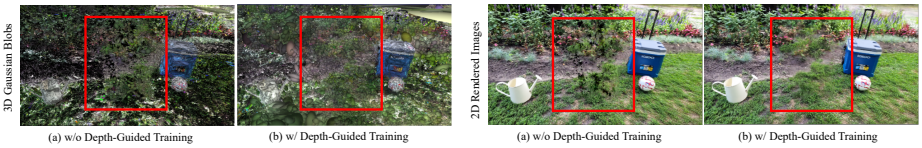

Fig. 5: Qualitative results of the effective depth-guided training. We visualize the scene in 3D Gaussian Splatting format and 2D rendered image by ablating the depth-guide training. The geometry guidance provides more information to fill the missing area with Gaussian blobs. Please zoom in for a better view.

row, SPIn-NeRF and OR-NeRF occasionally struggle with more complex regions. For instance, in the scenarios requiring the completion of both soil and bush textures (as seen in the first row), these methods often resort to inserting repetitive, unrealistic gray textures. In contrast, both the View-Sub method and our results can complete appropriate grass and plants. Similarly, in the second row, our completed railing appears more reasonable. While minor discrepancies in viewpoint exist between the results of the View-sub and ours, the fidelity of the completed textures in the first three scenes remains notably comparable.

Further analysis of the last two rows in Fig. 4, which shows two indoor scenes with more complex depth from the IBRNet dataset, reveals our method's proficiency. For instance, in the case of lamp removal, our method naturally completes the curtain behind the lamp compared to baselines. In the case of table removal, our method reconstructs the chair legs and carpet more accurately.

4.3 Ablation Study

We conduct ablation experiments on mono-depth supervision and cross-attention feature regularization and present the quantitative and qualitative results in Tab. 2, Fig. 5, and Fig. 6.

Analysis of Depth Supervision We first analyze our first contribution: introducing multi-view depth maps to aid in the 3D geometry learning of the inpainted area. Fig. 5 (a) and (b) show the results supervised by using Equ. 7 and Equ. 8 based on the original 3DGS. Note that the former does not have mono-depth supervision while the latter has mono-depth supervision. We visualize the learned Gaussian blobs and 2D images before and after incorporating depth supervision (all visualized in novel views). From Fig. 5, we can observe that in (a), where depth supervision is lacking, the positions of the Gaussian blobs within the red box are floating in the air, with noticeable holes interspersed in between. The corresponding 2D rendered images also exhibit noticeable texture floating. However, the involvement of depth supervision in (b) leads to more plausible positions of the 3D Gaussian blobs: most blobs are located within areas with objects (grass and bushes) rather than floating in the air as in (a). The corresponding 2D rendered images are noticeably more realistic and plausible. This demonstrates that our depth supervision significantly constrains the position of

Table 2: Quantitative comparison of different variants of our proposed method. We remove one or both of the Mono-Depth Supervision and Cross-Attn (Cross-Attention) Regularization components and compare the quantitative results.

Variants	PSNR ↑	masked-PSNR ↑	SSIM ↑	masked-SSIM ↑	LPIPS↓	masked-LPIPS ↓
GScream w/o Cross-Attn & Mono-Depth	20.12	14.87	0.58	0.19	**0.26**	0.56
GScream w/o Cross-Attn	20.47	15.63	0.58	0.20	**0.26**	**0.50**
GScream (Our Full Model)	**20.49**	**15.84**	**0.58**	**0.21**	0.28	0.54

Gaussian blobs and improves the geometric accuracy of 3DGS, which enables the realism of the 2D renderings in novel views.

Quantitative Analysis of Key Components We further disable Mono-Depth Supervision and Cross-Attention Feature Regularization modules individually based on the full model GScream, and present more quantitative and qualitative results of these ablation experiments in Tab. 2 and Fig. 6. Disabling Cross-Attention Feature Regularization means training only with Equ. 8, without performing 3D Gaussian sampling and bidirectional cross-attention. Disabling both means only retaining the color loss term in Equ. 8.

From the Tab. 2, we can observe that removing the cross-attention feature regularization modules leads to a degradation in the metrics PSNR and SSIM. For instance, the masked PSNR decreases from 15.84 to 15.63, indicating that the content filled in the masked regions becomes less reasonable. This suggests that improvements in depth accuracy and feature propagation are beneficial for the results. Furthermore, if both modules are disabled, the metrics become even worse. Compared to the full model, the masked PSNR decreases to 14.87, and the masked-SSIM further decreases to 0.19, suggesting poorer depth and no 3D regularization in masked regions lead to worse results.

Qualitative Analysis of the Mono-Depth Module. The label (a)(b)(c) in Fig. 6 represent (a) GScream w/o Cross-Attention & Mono-Depth; (b) GScream w/o Cross-Attention Regularization and (c) Our Full Method (GScream), respectively. For both Scene-1 and Scene-2, by comparing Fig. 6 (a) with (b)(c), we can observe that removing depth supervision results in poor depth prediction, with significant noise present in the red box region and along the image edges. The texture quality of scene (a) suffers notably due to the absence of depth supervision, resulting in texture holes when viewed from novel perspectives.

Qualitative Analysis of the Cross-Attention Module. While our experiments revealed a marginal reduction in the LPIPS upon deactivating the cross-attention module, we are poised to showcase this module's substantial role in enhancing our results in Fig. 6. While Fig. 6 (b) benefits from incorporating monocular depth supervision, leading to improved texture filling and depth accuracy, the outcomes still fall short of naturalness due to the absence of 3D feature regularization. In Scene-1 (b) of Fig. 6, when the perspective shifts to the left side of the tree trunk, black holes become visible in areas distanced from the frontal view, as indicated by the red arrow (zooming in is recommended for clarity). This scenario underscores the limitations of solely relying on 2D priors for supervi-

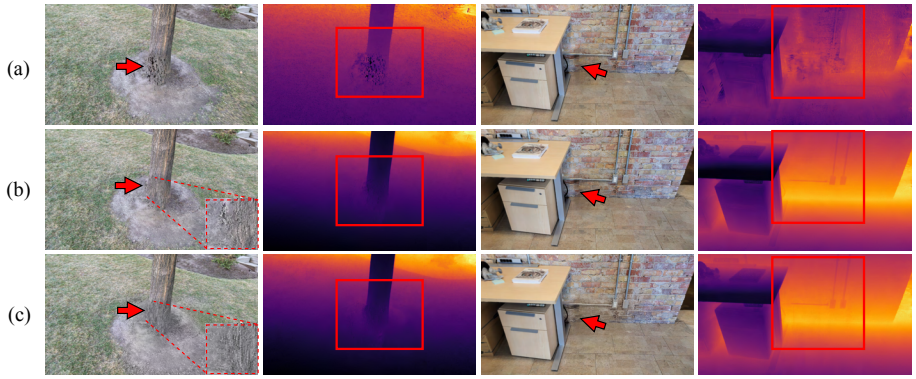

Fig. 6: Qualitative results of the ablation study. We provide the visualization of different variants of our method. From the top to bottom, **(a)** GScream w/o Cross-Attention & Mono-Depth; **(b)** GScream w/o Cross-Attention Feature Regularization; **(c)** Our Full Method GScream. We visualize the rendered RGB and depth to verify the effectiveness of our proposed components. Our full model produces a more reasonable depth and RGB image. Please zoom in for a better view.

sion, which are unable to remediate texture gaps in unseen regions. However, the introduction of 3D feature regularization in (c) effectively addresses these shortcomings by filling the previously observed holes. This enhancement reveals the critical role of 3D feature interactions in supplementing 2D priors, enabling the propagation of appropriate textures to obscured areas and thereby ensuring more cohesive rendering in novel views. In Scene-2, a side-by-side comparison of (a) and (b) reveals that, while (b) demonstrates depth enhancements over (a), both still exhibit a pronounced sharp boundary, as indicated by the red arrow, which detracts from naturalism. However, integrating feature cross-attention in (c) significantly mitigates this issue. The previously stark gap softens, eliminating the noticeable boundary. This transformation suggests that facilitating feature information exchange can harmonize originally disjointed textures at boundaries, ensuring a more seamless and consistent texture transition.

5 Conclusion

In conclusion, our innovative framework for object removal, which leverages 3D Gaussian Splatting, has proven to be both effective and more efficient than traditional NeRF-based approaches. Through the integration of monocular-depth guided training and cross-attention feature regularization techniques, our method facilitates rapid training speeds while simultaneously preserving multi-view geometric and texture consistency in the inpainted textures. Experimental validations confirm that our approach outperforms existing NeRF-based methods in terms of both efficiency and effectiveness.

Acknowledgements

This research is supported in part by the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321, SAIL Research Project, HKUST-Zeekr Collaborative Research Fund, HKUST-WeBank Joint Lab Project, and Tencent Rhino-Bird Focused Research Program.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV (2023)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques (2000)
4. Cen, J., Zhou, Z., Fang, J., Shen, W., Xie, L., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. In: NeurIPS (2023)
5. Chen, G., Wang, W.: A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 (2024)
6. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: CVPR (2024)
7. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: CVPR (2023)
8. Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: NeurIPS (2021)
9. Dai, A., Diller, C., Nießner, M.: Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In: CVPR (2020)
10. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: CVPR (2017)
11. Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M.: Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In: CVPR (2018)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
13. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing (2006)
14. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: CVPR (2024)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ToG (2023)
16. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. IJCV (2000)
17. Liu, H.K., Shen, I., Chen, B.Y., et al.: Nerf-in: Free-form nerf inpainting with rgb-d priors. arXiv preprint arXiv:2206.04901 (2022)

18. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. TOG (2019)
19. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: CVPR (2024)
20. Max, N.: Optical models for direct volume rendering. TVCG (1995)
21. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM (2021)
22. Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinshtein, A., Derpanis, K.G., Gilitschenski, I.: Reference-guided controllable inpainting of neural radiance fields. In: ICCV (2023)
23. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinshtein, A.: Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023)
24. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ToG (2022)
25. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019)
26. Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P.P., Mildenhall, B., Geiger, A., Barron, J.T., Hedman, P.: Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. TOG (2023)
27. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
28. RunwayML: Stable diffusion. `https://huggingface.co/runwayml/stable-diffusion-inpainting` (2021)
29. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
30. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. IJCV (1999)
31. Shih, M.L., Su, S.Y., Kopf, J., Huang, J.B.: 3d photography using context-aware layered depth inpainting. In: CVPR (2020)
32. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: CVPR (2019)
33. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019)
34. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: NeurIPS (2020)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
36. Wang, D., Zhang, T., Abboud, A., Süsstrunk, S.: Inpaintnerf360: Text-guided 3d inpainting on unbounded neural radiance fields. In: CVPR (2024)
37. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021)
38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004)
39. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003 (2003)

40. Weder, S., Garcia-Hernando, G., Monszpart, A., Pollefeys, M., Brostow, G.J., Firman, M., Vicente, S.: Removing objects from neural radiance fields. In: CVPR (2023)
41. Yin, Y., Fu, Z., Yang, F., Lin, G.: Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. arXiv preprint arXiv:2305.10503 (2023)
42. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In: NeurIPS (2022)
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
44. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: VIS (2001)