CoIn: Comprehensive 2D-3D Inpainting with Gaussian Splatting Guidance

Hana Kim[1,2], Minje Kim[2], and Tae-Kyun Kim[2]

[1] LG Electronics, Seoul, South Korea hana1106.kim@lge.com
[2] KAIST, Daejeon, South Korea {minjekim, kimtaekyun}@kaist.ac.kr

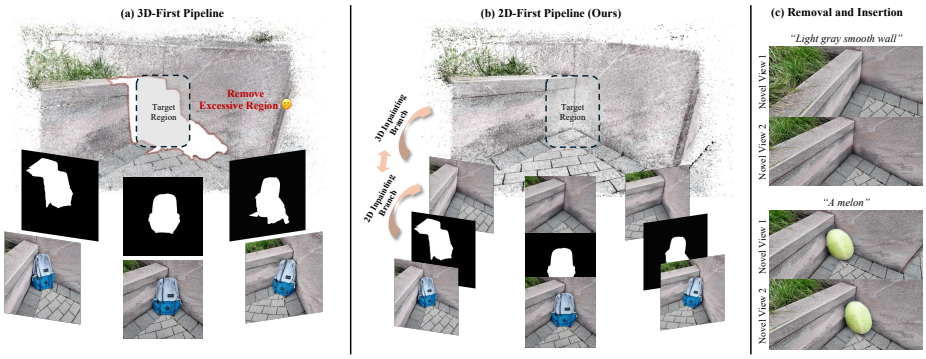

Fig. 1: Key impact of inconsistent masks on 3D-first vs. 2D-first pipelines. (a) Prior works, which typically adopt a 3D-first pipeline, often fail under inconsistent 2D segmentation masks across views, leading to excessive removal of regions and incorrect 3D segmentation. (b) In contrast, CoIn utilizes a 2D-first pipeline that comprehensively integrates 2D and 3D inpainting branches under Gaussian Splatting guidance. This approach ensures spatial and semantic consistency, enabling the use of arbitrary-shaped masks and supporting diverse tasks such as object removal and insertion (c).

Abstract. 3D scene inpainting is essential for reconstructing areas corrupted by occlusions or limited viewpoints. While recent methods leverage Gaussian Splatting (GS) for efficient 3D editing, they often depend on precise multi-view segmentation masks and are inherently constrained to object removal tasks. We propose CoIn, a novel framework that bridges 2D inpainting models and 3DGS through a multi-stage consistency pipeline. Our approach first generates initial inpainted images using a diffusion model, enabling the use of arbitrary-shaped masks and diverse tasks like object insertion. We then introduce Reference Adaptive GS with Feature Attention to reconstruct a coarse 3D scene by adaptively weighing towards a reference view (2D $\rightarrow$ 3D). This 3D representation provides geometric guidance to the diffusion process via GS-based Reference Feature Warping, ensuring multi-view consistency (3D $\rightarrow$ 2D). Finally, a Texture-Enhancing Discriminator refines the 3D scene to achieve high photometric realism (2D $\rightarrow$ 3D). Experiments show that CoIn, effectively leveraging bidirectional information flow, achieves state-of-the-art performance and effectively handles both object removal and object insertion with flexible mask input.

Keywords: 3D Inpainting · 3D Gaussian Splatting · Diffusion Guidance

1 Introduction

Inpainting 3D scene is essential for reconstructing incomplete or corrupted scenes arising from occlusions, sensor noise, or limited viewpoints. Recent advances in generative models have expanded this scope from simple object removal to editing and object insertion. To achieve this, various methods directly optimize Neural Radiance Fields (NeRF) [27] using priors such as RGB and depth [28] or diffusion-based [6]. However, the implicit representation of NeRF hinders direct geometry manipulation, necessitating complex volumetric optimization for precise removal, insertion, or shape editing.

To perform more geometric and efficient editing, subsequent methods [14, 37, 44, 46, 49] adopt the '3D-first' pipeline, which denotes the strategy of first reconstructing a 3D Gaussian Splatting (3DGS) [19] scene from the original images to provide a structural basis before performing inpainting. However, this sequence requires accurate segmentation masks (e.g., obtained from SAM2 [31]) across multiple views to precisely isolate target regions in 3D space. Furthermore, methods that initiate inpainting by pruning these reconstructed Gaussians, such as IMFine [37], 3DGIC [14], and AuraFusion [46], are inherently restricted to removal tasks. Although GScream [44] avoids initial pruning, it is based only on a single 2D inpainted reference image, which limits its ability to maintain consistency across large viewpoint changes.

By contrast, the '2D-first' pipeline denotes a strategy that prioritizes performing 2D inpainting across multiple views, followed by leveraging 3D information, such as optical flow [4] or meshes [3], to ensure consistency among the generated images. Although pipelines like MVInpainter [4] or Instant3dit [3] benefit from the diverse editing capabilities of 2D generative models, they often rely on fine-tuning with refined datasets or require additional accurate 3D labels.

We propose a novel framework that combines the flexibility of 2D-first pipelines with the efficient optimization of 3DGS. By integrating these approaches, our method supports arbitrary-shaped masks and handles diverse inpainting tasks with high efficiency. Multi-view consistency is maintained by guiding the 2D inpainter with 3D priors from Reference Adaptive GS, while a Texture Enhancing Discriminator further refines photometric realism. We demonstrate our approach through both quantitative and qualitative experiments for object removal and insertion tasks. In summary, our contributions are as follows:

- We propose CoIn, a novel framework that seamlessly bridges generative 2D synthesis and explicit 3D reconstruction via a multi-stage pipeline.
- We introduce Reference Adaptive Gaussian Splatting with Feature Attention (Ref-GS), which assigns adaptive weights to each view to optimize 3DGS toward the reference viewpoint.
- We present Consistency Loss Guidance (CLG), which leverages GS-based Reference Feature Warping to enforce 3D consistency with the reference image during the denoising process.
- The Texture Enhancing Discriminator (TE-D) mitigates blurriness in generated patches by learning the distribution of real image patches.

2 Related Work

2.1 2D Image Inpainting

2D inpainting refers to the task of reconstructing missing or masked regions in an image by leveraging contextual cues from visible areas. Early approaches mainly filled missing areas by copying nearby content, often resulting in visually inconsistent completions [9]. With the advent of neural networks trained on datasets, recent methods [17, 26, 33, 41, 43] used combinations of convolution networks, such as GAN [51], Fourier convolution [39], or Wavelet decomposition [16].

Meanwhile, diffusion-based models are widely adopted for 2D inpainting due to their ability to produce diverse yet highly feasible results [26, 33]. RePaint [26] extends the concept of unconditional DDPMs using the known region of an image as a condition, allowing the denoising process to serve as inpainting for the unknown region. Recent approaches [17, 43] incorporate latent diffusion models [32] to inpainting tasks. Stable diffusion (SD) inpainting pipeline [32] takes the masked image as input and performs denoising directly in the latent space, achieving faster inference while producing plausible and realistic results.

However, using pure 2D image inpainters without any additional modification, stochastic sampling in the denoising process leads to severely inconsistent results given small viewpoint changes across multi-view images, even for the same images. These inconsistencies are not suitable for subsequent 3D reconstruction, causing inaccurate geometries and blurriness. While several methods [1,3,4,8,21,23,30,35,36,40,45,54] consider 3D consistency for 2D generation and editing, they often exhibit trade-offs between computational efficiency and structural flexibility. Specifically, MVInpainter [4] and NeRFiller [45] attempt to enforce consistency via flow supervision or shared grid priors; however, these approaches still encounter limitations in mask shape flexibility, output resolution, or the need for per-dataset adaptation.

2.2 Guidance for Latent Diffusion Model

There exist various strategies [12, 20, 42] to enable diffusion models to perform specific tasks or adapt to particular domains, specifically, fine-tuning [13], incorporating a pretrained adapter [29,48], or injecting guidance [2,38,50] to align the model with a desired task or dataset. However, fine-tuning or training adapters generally relies on task-specific training data, which limits the flexibility of per-scene adaptation.

Instead, guidance-based control methods [2, 38, 50] introduce additional control signals to the denoising process at inference time. FreeDoM [50] formulates an energy function that measures the distance between the given condition c and the noisy intermediate result x_t. By minimizing the value of the energy function during denoising steps, diffusion models can generate desirable results without additional training. Our method leverages guidance-based control to generate 3D-consistent inpainted images. We incorporate guidance from the trained 3DGS within the energy function for the inpainting diffusion model.

2.3 3D Scene Inpainting

NeRF-based 3D inpainting methods [6, 22, 28] have recently shown strong performance by optimizing an implicit radiance field that can synthesize missing geometry and appearance from multi-view observations. However, the implicit formulation typically involves heavy computation and long training/rendering time, and explicit local edits often require re-optimization or complex volumetric procedures, limiting practical adaptation.

With 3D Gaussian Splatting (3DGS) [19], many subsequent approaches have performed inpainting explicitly in 3D space using point-based scene representations [14, 37, 44, 46, 49]. They usually get one reference image as a guide for inpainting the 3D scene. However, GScream [44] relies only on a single reference view that does not cooperate with other viewpoint images, large deviations from the reference view cause 3D inpainting to fail. In contrast, 3D-first pipelines remove objects directly in 3D and complete geometry using cues such as Laplacian smoothing to warp the reference image, thereby handling large viewpoint changes [14, 37, 46]. Although efficient, they rely on highly accurate 2D segmentation masks to specify a target object consistently across dense views. As shown in Fig. 1, in the absence of 3D consistency in the masks, the segmented regions do not align in the 3D space. This leads to geometrically inconsistent completion or unintended deletion of non-target regions and degrades 3D stability. Moreover, these pipelines are designed primarily for object removal rather than general inpainting tasks, such as insertion.

To the best of our knowledge, CoIn represents a comprehensive framework that establishes a correlated inpainting pipeline by integrating the generative capabilities of 2D models with the explicit representation of 3D Gaussian Splatting. Our framework, CoIn, is designed to leverage both 2D and 3D strengths: a 2D inpainting branch enables semantically meaningful edits under arbitrary-shaped masks, while an explicit 3D inpainting branch ensures strict multi-view consistency. Unlike 3D-first pipelines, CoIn does not require accurate segmentation masks and supports both removal and insertion, while avoiding the cross-view inconsistency issues common in 2D multi-view inpainting.

3 Preliminary

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [19] is an explicit point-based scene representation in which each Gaussian primitive is parameterized by its center position μ, rotation matrix R, scale S, color c and opacity α. The covariance matrix of each Gaussian is defined as $\Sigma = RSS^TR^\top$, and Gaussians are represented by

$$\mathcal{G}(x) = \exp\big(-\tfrac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\big) \tag{1}$$

Training is conducted by minimizing a combination of $\mathcal{L}_1$ and $\mathcal{L}_{SSIM}$ between the rendered image R_n and the ground-truth RGB image I_n:

$$\mathcal{L}_R(R_n, I_n) = (1-\lambda)\mathcal{L}_1 + \lambda(1-\mathcal{L}_{SSIM}) \tag{2}$$

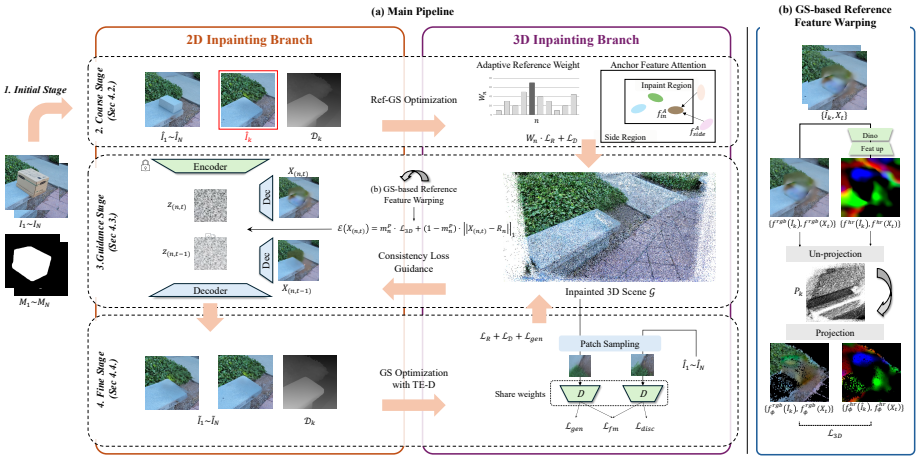

Fig. 2: Overview of CoIn. We begin with initial 2D inpainting results and apply Reference GS with Feature Attention to obtain a coarse inpainted 3D scene. We then use Consistency Loss Guidance for the frozen latent diffusion inpainting model with GS-based Reference Feature Warping, and the consistency-preserved results are finally used to fine-tune the 3D Gaussian Splatting scene $\mathcal{G}$ with a Texture-Enhancing Discriminator for realistic details.

Scaffold-GS. Unlike the original 3DGS, which directly optimizes the parameters of distributed gaussians, Scaffold-GS [25] adopts a lightweight anchor-based structure. Each anchor has a learnable feature embedding (anchor feature), and compact MLPs decode this embedding into neural gaussian attributes within its voxel-defined local region. This hierarchical design reduces redundancy by performing densification at the anchor level, achieving comparable rendering quality to the original 3DGS. The sparse anchor structure also enables efficient point-based guidance to maintain geometric correspondence across views, while anchor features permit attention to be applied directly in 3D, thereby facilitating 3D inpainting. We adopt it as our base model for representing the 3D scene, and propose an efficient object handling solution.

4 Method

4.1 Coherent 2D-3D Inpainting

Overview. Our method aims to achieve 3D-consistent inpainting from a set of object-present input images $\{I_n\}_{n=1}^{N}$ and their corresponding masks $\{M_n\}_{n=1}^{N}$. We begin with initial inpainting results $\{\hat{I}_n\}_{n=1}^{N}$ obtained from a SD inpainting model, which produces visually plausible completions for each image but fails to achieve consistency across views. To address this issue, we construct a 3D Gaussian scene from these initial results and refine it to enforce multi-view coherence while preserving fine details.

Fig. 2a illustrates our pipeline. We first apply a coarse stage, the Reference Adaptive GS with Feature Attention up-weights a selected reference view $\hat{I}_k$ and down-weights the others via per-view weights, while regularizing anchor features in the inpainting region by attending to neighboring anchors. It then enforces multi-view geometric and appearance consistency by warping reference features on the GS-derived point cloud and injecting them as guidance into the latent diffusion inpainting model. With the consistency-guided inpainted images, we refine the GS scene for high-frequency textures and photometric realism via an adversarial patch discriminator in a fine stage.

By integrating these components, our framework produces a 3D inpainted scene that is geometrically consistent, visually coherent, and seamlessly completed across all views. We introduce the Reference Adaptive GS with Feature Attention (Ref-GS) in Sec. 4.2, Consistency Loss Guidance (CLG) in Sec. 4.3, and Texture Enhancing Discriminator (TE-D) in Sec. 4.4.

4.2 Reference Adaptive GS with Feature Attention

We first construct a 3D Gaussian Splatting (3DGS) scene to provide 3D guidance to the 2D inpainting branch. However, employing vanilla GS often yields view-dependent inconsistencies and overly blurred renders, which are not suitable for multi-view inpainting. To stabilize the scene, we applied a per-view weight to the rendering loss, encouraging the GS scene to align with the reference view $\hat{I}_k$ while suppressing views that are inconsistent. Before optimization, we prune points whose projections fall inside the masks, thereby eliminating the influence of the residual object from the COLMAP initialization [34].

For each view n, we compute a weight W_n that up-weights the reference view $\hat{I}_k$ and down-weights views inconsistent with the reference view:

$$W_n = \begin{cases} \lambda_r, & \text{if } n = k \\ \dfrac{1}{1 + \exp(\lambda_r(\mathcal{L}_R^{(n,t)} - \mathcal{L}_R^{(n,t-1)}))}, & \text{if } n \neq k \end{cases} \tag{3}$$

where $\mathcal{L}_R^{(n,t)}$ represents the current photometric loss of the n-th view of t-th iteration, $\mathcal{L}_R^{(n,t-1)}$ is from the previous iteration of the same view, and λ_r is a weight parameter to the reference view. This fuction leads the GS scene to better reflect the reference image.

Although GScream [44] employs cross-attention to propagate texture from the surrounding region into the inpainted region, it is sensitive to variations in view range since it is applied only to the reference view. To address this limitation, we use unidirectional attention on anchor features for every training view. We update the features f_{in}^A via $\tilde{f_{\text{in}}^A} = \text{Attn}(Q = f_{\text{in}}^A,\ K = f_{\text{side}}^A,\ V = f_{\text{side}}^A)$ where f_{in}^A and f_{side}^A denote anchor features inside the inpainting region and in its neighborhood, respectively. This update enhances the consistency between the inpainted region and the surrounding areas, resulting in more coherent textures.

In addition, we apply a depth loss in the reference view using the estimated depth, making the GS-derived point cloud P_k more stable and better aligned with

the reference view geometry. We begin by using a monocular depth estimator [47] to estimate depth $\mathcal{D}_k$ from $\hat{I}_k$. With a linear transformation, we transform the rendering depth $R^{\mathcal{D}}$ to $\bar{R^{\mathcal{D}}} = A * R^{\mathcal{D}} + B$. A and B are obtained by solving a least-squares problem with respect to $\mathcal{D}_k$ [18, 44, 52]. Then the total GS optimization loss is computed as follows:

$$\mathcal{L} = W_n \cdot \mathcal{L}_R(R_n, \hat{I}_n) + \lambda_{\mathcal{D}} \mathcal{L}_{\mathcal{D}} \tag{4}$$

$$\mathcal{L}_{\mathcal{D}} = \frac{1}{HW} \sum ||\bar{R^{\mathcal{D}}} - \mathcal{D}_k||_1 \tag{5}$$

where $\lambda_{\mathcal{D}}$ denotes the weight for the depth loss, and H and W are height and width of the image, respectively.

4.3 Consistency Loss Guidance

We enforce cross-view consistency among inpainted images with an energy-based guidance inspired by FreeDoM [50]. During diffusion denoising at step t, we update the sample to minimize an energy function built from the GS scene (Sec. 4.2), leveraging both its renderings and the reconstructed point cloud P_k. We improve the consistency between the reference image $\hat{I}_k$ and the current view decoded image $X_{(n,t)} = Dec(z_{(n,t)})$ from the denoising diffusion latent $z_{(n,t)}$ by introducing GS-based Reference Feature Warping (Fig. 2b). Given the point cloud P_k and the camera parameters, we unproject per-view features into 3D and reproject them into an intermediate viewpoint ϕ. This viewpoint is established by interpolating between the reference ($\mathbf{P}_k$) and current ($\mathbf{P}_n$) camera poses; specifically, we apply linear interpolation for translation, $\mathbf{t}_\phi = (1-\alpha)\mathbf{t}_k + \alpha \mathbf{t}_n$, and Spherical Linear Interpolation (Slerp) for rotation, $\mathbf{q}_\phi = \text{Slerp}(\mathbf{q}_k, \mathbf{q}_n; \alpha)$, with $\alpha = 0.5$. From each image, we extract (i) high-frequency features with DINO [5] upscaled via Feat-Up [10], and (ii) appearance (RGB) features. Let $f_\phi(\cdot; P_k)$ denote the warp operator that projects the features to the intermediate view ϕ with P_k. We then form a 3D consistency loss by applying a cosine similarity term to the high-frequency features and a $L2$ similarity term to the RGB features:

$$\mathcal{L}_{3D} = 1 - cos(f_\phi^{hr}(\hat{I}_k), f_\phi^{hr}(X_{(n,t)})) + ||f_\phi^{rgb}(\hat{I}_k) - f_\phi^{rgb}(X_{(n,t)})||_2^2 \tag{6}$$

This GS-based Reference Feature Warping provides geometry-aware guidance at step t to enforce cross-view consistency. To specifically account for non-overlapping regions between the current view and the reference view, we introduce the point-cloud support mask m_n^P, which identifies areas where valid 3D correspondences are available. Furthermore, to compensate for imperfections in the reconstructed point cloud P_k, we incorporate an $L1$ loss against the GS rendering R_n as a complementary guidance.

Consequently, the final energy function is formulated as:

$$\mathcal{E}(X_{(n,t)}) = m_n^P \cdot \mathcal{L}_{3D} + (1 - m_n^P) \cdot ||X_{(n,t)} - R_n||_1 \tag{7}$$

By minimizing the value of $\mathcal{E}(X_{(n,t)})$, our framework encourages multi-view consistency and geometry-aware inpainting by leveraging correspondences from the GS scene within the mask.

4.4 Texture Enhancing Discriminator

Although Sec. 4.3 produces per-view inpainting results, they still exhibit residual artifacts and blurriness due to imperfect geometry and strong guidance. We therefore introduce a Texture-Enhancing Discriminator (TE-D) that transfers only texture information from the initial inpainting output $\hat{I}_n$ to the GS renderings R_n. Specifically, TE-D is trained jointly with the GS model on small image patches, treating patches from $\hat{I}_n$ as real and patches from the current GS renderings as fake. The GS model (generator) is trained adversarially against the discriminator, enhancing high-frequency texture details while preserving the overall geometric structure. Operating in local patches [15] near the inpainted region, it emphasizes texture fidelity rather than geometry, helping remove guidance-induced blur (see Fig. 6A(c)). Simultaneously, the depth loss and the rendering loss used in the GS optimization are computed with the CLG inpainting images $\tilde{I}_n$.

$$\mathcal{L} = \mathcal{L}_R(R_n, \tilde{I}_n) + \lambda_D \mathcal{L}_{\mathcal{D}} + \lambda_{gen} \mathcal{L}_{gen}(R_n, \hat{I}_n) \tag{8}$$

where λ_{gen} denotes the weight for the adversarial loss. The combination of adversarial texture matching to $\hat{I}_n$ and photometric agreement with $\tilde{I}_n$ fine-tunes the inpainted 3D scene $\mathcal{G}$, yielding high fidelity while maintaining the multi-view consistency established in the previous stage.

5 Experiments

5.1 Experimental Setup

Datasets. Following prior works, we perform experiments on the SPIn-NeRF dataset [28]. The dataset contains 10 scenes, each providing 100 calibrated images, 60 object-present and 40 object-absent. For evaluation, we inpaint the object-present images using the provided segmentation masks or bounding boxes derived from them. After inpainting, we synthesize novel views at the object-absent viewpoints to compute the metrics. We also evaluated on the IMFine dataset [37], which comprises 20 scenes grouped into four coverage cases (90°, 120°, 180°, 360°) having broader view changes than SPIn-NeRF. Each scene contains 125 object-present and 75 object-absent images, along with binary masks and calibrated camera parameters. To better show the effect of the reference image, we evaluated ten scenes with 90 °, 120 °and 180 °viewpoint coverage. The 360 °scenes are shown in the supplementary materials.

Evaluation Metrics. We report LPIPS [53], PSNR, and FID [11] on the full images. To show the performance fidelity in masked regions more effectively, we also report the masked variants, m-LPIPS, m-PSNR, and m-FID, computed

Table 1: Quantitative evaluation on the SPIn-NeRF dataset [28]. Note that m-LPIPS [53] and m-FID [11] represent the LPIPS and FID scores within the ground truth segmentation masks. BB mask denotes a bounding-box mask derived from the segmentation mask. The top three results are highlighted in red, orange, and yellow, respectively.

3D Representation	Mask Type		m-LPIPS (↓)	LPIPS (↓)	m-FID (↓)	FID (↓)
NeRF	Seg mask	SPIn-NeRF [28]	0.053	0.31	153.4	49.6
		MVIP-NeRF [6]	0.050	0.31	173.4	50.5
		MALD-NeRF [22]	0.031	0.30	113.5	44.7
Gaussian Splatting	Seg mask	Gaussian Grouping [49]	0.037	0.26	132.5	44.9
		3DGIC [14]	0.028	0.26	96.3	36.4
		GScream [44]	0.032	0.26	86.1	31.2
		Ours	0.032	0.23	80.4	28.9
	BB mask	3DGIC	0.047	0.38	168.0	104.1
		GScream	0.043	0.29	104.6	34.2
		Ours	0.033	0.24	95.1	26.6

only within the ground-truth segmentation mask. For the object insertion task, we follow established practices by calculating the $CLIP_{dir}$ (CLIP Text-Image Directional Similarity) to assess how well the generated 3D content aligns with the provided text instructions. Additionally, we conduct a user study to evaluate and compare our method with state-of-the-art baselines. Specifically, we utilize a comparative voting process to assess four key dimensions: multi-view consistency, overall visual quality, alignment with text prompts and absence of visual artifacts. Detailed experimental setups and the specific questions used in the user study are provided in the supplementary material.

Implementation Details. All experiments were conducted on a single NVIDIA RTX 4090 (24GB) GPU. We utilize Stable Diffusion 2.0 [32] for 2D inpainting, processing 512×512 mask-centered crops. The regions outside the mask are combined with the ground-truth image to preserve originality when training the GS model. Following prior works [37, 44], we empirically select a single reference view $\hat{I}_k$ from the initial 2D inpainting results to serve as a basis for maintaining consistency across the remaining views. For memory efficiency during the Consistency Loss Guidance (CLG) stage, we employ normalized DINO (dino16) [5] and FeatUp [10] features extracted from 256×256 resized images. The TE-D module is trained in two distinct stages: first, we perform 10k iterations of fine-tuning the generator $\mathcal{G}$ on the CLG-inpainted images $\tilde{I}_n$, followed by 10k iterations of joint adversarial training. During this training, 64×64 patches are extracted via mask-based probabilistic sampling to focus on the inpainted regions. Detailed hyperparameter settings, including the loss weights λ_D, λ_r, and λ_{gen} are provided in the supplementary material. The entire pipeline requires approximately 2.5 hours per scene, which is more efficient than typical learning-based 3D inpainting approaches.

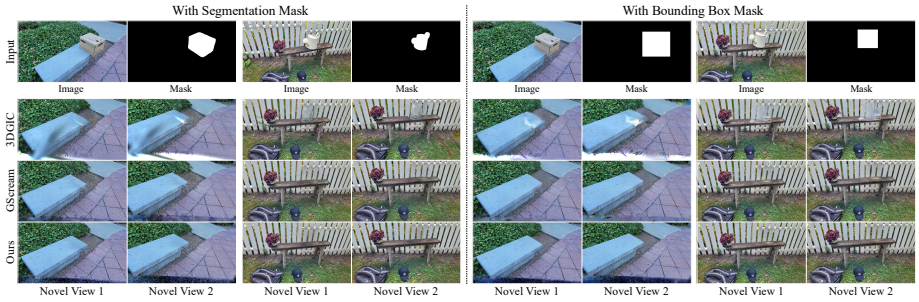

Fig. 3: Qualitative results on the SPIn-NeRF dataset [28]. The first row shows the input images and corresponding masks for each scene, followed by the inpainting results of the baseline methods 3DGIC [14], GScream [44], and Ours in subsequent rows. By comparing the four columns on the left with the four columns on the right, we can observe the difference between using the segmentation masks and the bounding-box masks for inpainting.

5.2 Quantitative Results

Tab. 1 presents quantitative results on the SPIn-NeRF dataset [28] compared to baseline methods. For the setting using segmentation masks, we use the numbers reported in 3DGIC [14] for SPIn-NeRF [28], MVIP-NeRF [6], MALD-NeRF [22], Gaussian Grouping [49], and 3DGIC [14]. For fairness, we reproduce GScream [44] (with segmentation and bounding-box masks) and 3DGIC (with bounding-box masks) using official implementations. Since both GScream and 3DGIC require a single reference image, we follow the reference preparation protocol specified in each official implementation. Our method achieves state-of-the-art performance on most entries under both the segmentation and bounding-box settings. Notably, the performance gap between the two types of masks is small for our method, indicating that our method is less sensitive to mask shapes.

Tab. 2 compares our method with 3DGIC [14], GScream [44], and IMFine [37] on the IMFine dataset. For IMFine, we compute the metrics with officially released result images, while 3DGIC and GScream are evaluated through our reproduction using their official implementations. Our method outperforms baselines across LPIPS [53], PSNR, and FID [11] under the segmentation mask setting.

In Tab. 3, we evaluate the object insertion task using $CLIP_{dir}$ and a user study. The results indicate that our method also exhibits certain advantages compared to other methods in terms of both semantic alignment and human preference. In particular, the total sum of votes in the user study may not be equal to 100% because a 'None of them' option was provided to ensure an unbiased evaluation, allowing participants to reject all candidates if none were satisfactory. Detailed information on the setup of the user study, including the specific questions and the number of participants, is provided in the supplementary material.

Table 2: Quantitative results on the IMFine dataset [37]. Bold numbers indicate the best performance, and underlined numbers represent the second-best results.

	LPIPS (↓)	PSNR (↑)	FID (↓)
3DGIC [14]	0.3386	20.29	200.99
GScream [44]	0.2000	22.68	112.91
IMFine [37]	<u>0.1747</u>	<u>23.59</u>	<u>57.53</u>
Ours	**0.1685**	**23.88**	**53.37**

Table 3: Quantitative evaluation of object insertion. We compare our method against baseline models using the $CLIP_{dir}$ and a user study across four dimensions. Values in the user study columns represent the percentage of user preference votes. Bold numbers indicate the best performance. $CLIP_{dir}$: CLIP Text-Image Direction Similarity; Align with T.P. : Align with Text Prompt

	$CLIP_{dir}$ ↑	Consistency (%)	Visual (%)	Align with T.P. (%)	w/o Artifacts (%)
Gaussian Editor [7]	0.0222	20.825	23.625	13.900	17.000
Infusion [24]	0.1589	31.950	31.950	26.975	30.550
Ours	**0.1628**	**47.225**	**44.425**	**59.125**	**45.400**

5.3 Qualitative Results

In Fig. 3, we present comparisons on the SPIn-NeRF dataset [28] using both segmentation and bounding-box masks. The first row shows the input image and its corresponding mask used for training. In the second row, we observe that 3DGIC [14] fails to preserve the non-inpainted regions and generates white artifacts inside the inpainted regions. For GScream [44], the generated content does not align with nearby regions (e.g., the wooden bench in the third and fourth columns). In contrast, our method generates high-fidelity images with multi-view consistency while preserving the non-inpainted regions. Moreover, the inpainting quality is maintained when changing from the segmentation to the bounding-box settings, indicating that our approach is robust to mask shapes.

We further compare our method against 3DGIC [14], GScream [44], and IMFine [37] on the IMFine dataset, as shown in Fig. 4. 3DGIC fails to inpaint removed regions when the geometry is complex, and GScream shows inconsistency when the viewpoint variation is large. Especially in second scene, while IMFine produces high-fidelity images, residual object traces remain after inpainting.

Beyond removal, we also perform object insertion across several scenes. As shown in Fig. 5, Gaussian Editor [7] fails to synthesize new objects, merely altering colors without handling the required geometric changes, especially in the *"Jansport"* scene. Infusion [24] produces a distorted geometry for the apple in the *"Apples2"* scene and leaves artifacts behind the melon in the *"Jansport"* scene. By contrast, our approach synthesizes high-fidelity objects that are seamlessly integrated into the scene while maintaining strict multi-view consistency across varying viewpoints.

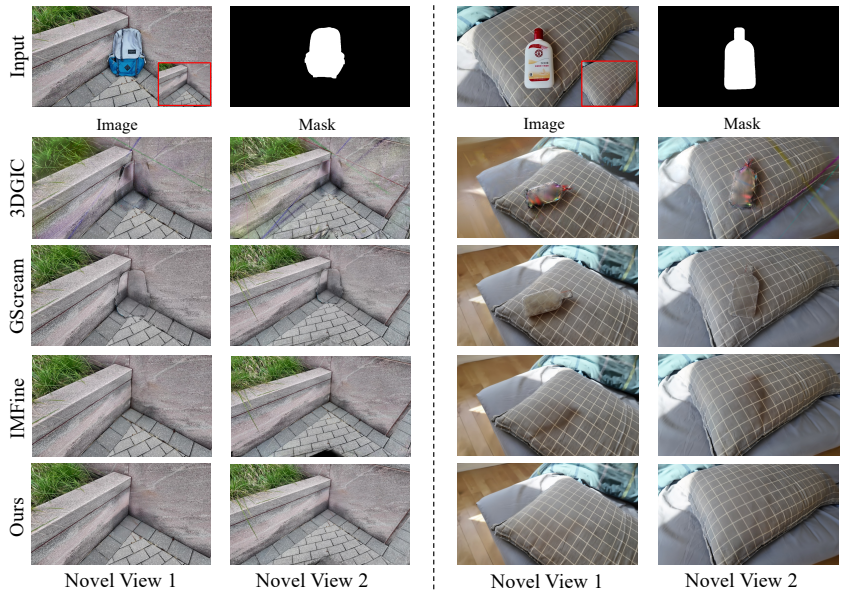

Fig. 4: Qualitative results on the IMFine dataset [37]. Same as in Fig. 3, the first row shows the input images and corresponding masks. We compare the rendering results with 3DGIC [14], GScream [44], IMFine [37]. Red box in the input image indicates the reference image for 3DGIC, GScream and Ours.

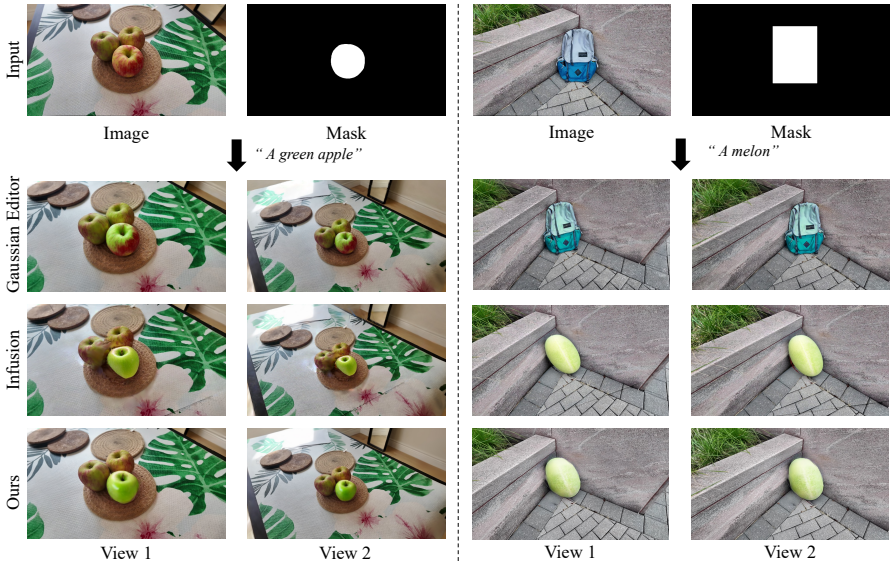

Fig. 5: Qualitative results for object insertion task. We perform object insertion tasks on the *"Apples2"* and *"Jansport"* scenes of IMFine dataset [37]. We compare the rendering results with Gaussian Editor [7] and Infusion [24].

Table 4: Quantitative results for ablation studies on the SPIn-NeRF dataset [28]. The full model with all key components achieves the best performance.

Methods (↓)	m-LPIPS	LPIPS	m-FID	FID
w/o Ref-GS	0.0359	0.2534	87.05	30.47
w/o Adaptive Weight	0.0356	0.2571	82.07	30.76
w/o Feature Attention	0.0355	0.2559	88.15	31.78
w/o CLG	0.0428	0.3057	97.04	36.47
w/o TE-D	0.0535	0.2902	132.55	39.31
Full model (Ours)	**0.0317**	**0.2349**	**80.44**	**28.92**

5.4 Ablation Study

Effectiveness of key components. We conduct ablation studies on the SPIn-NeRF dataset [28] to evaluate the effectiveness of our three key components: Reference Adaptive Gaussian Splatting with Feature Attention (Ref-GS), Consistency Loss Guidance (CLG), and Texture-Enhancing Discriminator (TE-D). As shown in Tab. 4, removing any component leads to a noticeable decrease in performance in all metrics, indicating that each module is essential for high-quality, consistent inpainting. Specifically, we further examine the individual components of Ref-GS by separately ablating Adaptive Weight and Feature Attention. Excluding either component leads to performance degradation, confirming that both components are vital for aligning the 3DGS scene toward the reference view while maintaining overall structure. Specifically, removing TE-D significantly decreases accuracies, especially in the m-FID/FID [11], which indicates that TE-D improves rendering quality with increasing the photometric reality.

Fig. 6A shows the qualitative comparisons of the ablation study. Rows (a)–(d) correspond to without Ref-GS, without CLG, without TE-D, and the Full model, respectively. Column (a) fails to generate cross-view consistent results (see the yellow circle), and column (b) shows irregular artifacts. Although column (c) achieves consistent results, it produces overly smooth textures compared to the full model. This over-smoothing behavior corresponds to the lowest quantitative scores on all metrics except LPIPS [53]. For LPIPS, the jittering textures observed in the results without CLG lead to even worse scores.

Effectiveness of consistent 2D images on 3D geometry. To assess the effect of enforcing 2D consistency during inpainting on the recovered 3D geometry, we present RGB renderings and depth maps in Fig. 6B. The figure reports 3DGS results obtained by training with three types of inputs: occluded 2D images, 2D inpainting outputs without guidance, and our guided images. In the first column, the occluded regions remain visible in both the RGB rendering and the depth map. Although training with unguided 2D inpainting outputs appears to fill the occlusions in the RGB renderings, these renderings remain blurry and the depth maps lack geometric stability. In contrast, ours shows more plausible and consistent RGB rendering and depth map. The results indicate that training

3DGS with consistent 2D images improves not only photometric fidelity but also underlying geometric quality.

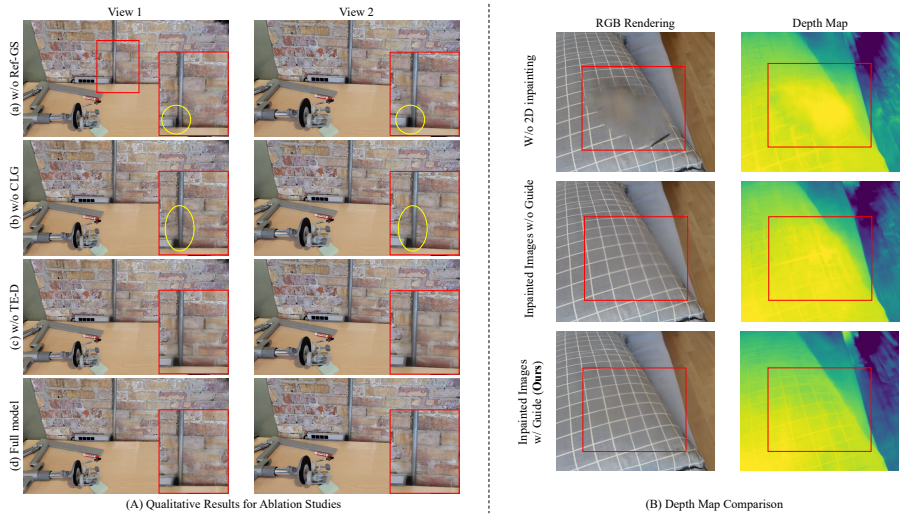

Fig. 6: Results of ablation studies (A) Qualitative results for ablation studies on the *"Book"* scene of SPIn-NeRF dataset [28]. We mark inpainted regions with red boxes and especially highlight inconsistencies with yellow circles. Rows (a)–(d) show w/o Ref-GS, w/o CLG, w/o TE-D, and the full model, respectively. (B) Depth map comparison on the *"Dabao"* scene from the IMFine dataset [37]. We highlight the inpainted regions with red boxes. For each row, we present the RGB rendering and the depth map from the same viewpoint.

6 Conclusions

In this paper, we present CoIn, a comprehensive 2D–3D inpainting framework. By combining the strengths of 2D inpainting and 3D inpainting, our method supports arbitrary-shaped masks while maintaining cross-view geometric and appearance consistency. We achieve this by optimizing the GS scene toward a reference view with adaptive weights (Ref-GS) and using it to guide the 2D inpainter via GS-based Reference Feature Warping. With the guided inpainting images and Texture-Enhancing Discriminator, we fine-tune the inpainted GS scene to obtain multi-view consistency and recover high-frequency textures. In the experiments, we achieve state-of-the-art performance in both quantitative and qualitative aspects. Extensive experiments show that CoIn handles multiple inpainting tasks (e.g., removal and insertion) under both segmentation and bounding-box masks, outperforming existing 3D inpainting approaches.

Acknowledgements

This work was supported by NST grant (CRC 21015, MSIT), IITP grant (RS-2023-00228996, RS-2024-00459749, RS-2025-25443318, RS-2025-25441313, RS-2026-25526850, RS-2026-25522885, MSIT), KOCCA grant (RS-2024-00442308, MCST) and InnoCORE program (N10260110, MSIT).

References

1. Ai, H., Cao, Z., Lu, H., Chen, C., Ma, J., Zhou, P., Kim, T.K., Hui, P., Wang, L.: Dream360: Diverse and immersive outdoor virtual scene creation via transformer-based 360 image outpainting. IEEE Transactions on Visualization and Computer Graphics **30**(5), 2734–2744 (2024), presented at IEEE Virtual Reality (VR) 2024
2. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 843–852 (2023)
3. Barda, A., Gadelha, M., Kim, V.G., Aigerman, N., Bermano, A.H., Groueix, T.: Instant3dit: Multiview inpainting for fast editing of 3d objects. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16273–16282 (2025)
4. Cao, C., Yu, C., Wang, F., Xue, X., Fu, Y.: Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing. arXiv preprint arXiv:2408.08000 (2024)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
6. Chen, H., Loy, C.C., Pan, X.: 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. pp. 5344–5353 (2024)
7. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21476–21485 (2024)
8. Deng, Z., He, X., Peng, Y., Zhu, X., Cheng, L.: Mv-diffusion: Motion-aware video diffusion model. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 7255–7263 (2023)
9. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1033–1038. IEEE (1999)
10. Fu, S., Hamilton, M., Brandt, L., Feldman, A., Zhang, Z., Freeman, W.T.: Featup: A model-agnostic framework for features at any resolution. arXiv preprint arXiv:2403.10516 (2024)
11. 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. Advances in neural information processing systems **30** (2017)
12. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
13. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)

14. Huang, S.Y., Chou, Z.T., Wang, Y.C.F.: 3d gaussian inpainting with depth-guided cross-view consistency. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26704–26713 (2025)
15. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
16. Jeevan, P., Kumar, D.S., Sethi, A.: Wavepaint: Resource-efficient token-mixer for self-supervised inpainting. arXiv preprint arXiv:2307.00407 (2023)
17. Ju, X., Liu, X., Wang, X., Bian, Y., Shan, Y., Xu, Q.: Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In: European Conference on Computer Vision. pp. 150–168. Springer (2024)
18. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9492–9502 (2024)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
20. Kim, J., Kim, T.K.: Arbitrary-scale image generation and upsampling using latent diffusion model and implicit neural decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
21. Kim, M., Kim, T.K.: Srhand: Super-resolving hand images and 3d shapes via view/pose-aware nueral image representations and explicit 3d meshes. In: Advances in Neural Information Processing Systems (NIPS) (2025)
22. Lin, C.H., Kim, C., Huang, J.B., Li, Q., Ma, C.Y., Kopf, J., Yang, M.H., Tseng, H.Y.: Taming latent diffusion model for neural radiance field inpainting. In: European Conference on Computer Vision. pp. 149–165. Springer (2024)
23. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9298–9309 (2023)
24. Liu, Z., Ouyang, H., Wang, Q., Cheng, K.L., Xiao, J., Zhu, K., Xue, N., Liu, Y., Shen, Y., Cao, Y.: Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. arXiv preprint arXiv:2404.11613 (2024)
25. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)
26. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
28. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20669–20679 (2023)
29. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 4296–4304 (2024)

30. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
31. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
33. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–10 (2022)
34. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
35. Shen, Q., Yang, X., Wang, X.: Anything-3d: Towards single-view anything reconstruction in the wild. arXiv preprint arXiv:2304.10261 (2023)
36. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
37. Shi, Z., Huo, D., Zhou, Y., Min, Y., Lu, J., Zuo, X.: Imfine: 3d inpainting via geometry-guided multi-view refinement. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26694–26703 (2025)
38. Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.Y., Kautz, J., Chen, Y., Vahdat, A.: Loss-guided diffusion models for plug-and-play controllable generation. In: International Conference on Machine Learning. pp. 32483–32498. PMLR (2023)
39. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
40. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: European Conference on Computer Vision. pp. 1–18. Springer (2024)
41. Wang, T., Zhang, K., Shao, Z., Luo, W., Stenger, B., Lu, T., Kim, T.K., Liu, W., Li, H.: Gridformer: Residual dense transformer with grid structure for image restoration in adverse weather conditions. International Journal of Computer Vision (IJCV) **132**, 4541–4563 (2024)
42. Wang, T., Zhang, K., Zhang, Y., Luo, W., Stenger, B., Lu, T., Kim, T.K., Liu, W.: Lldiffusion: Learning degradation representations in diffusion models for low-light image enhancement. Pattern Recognition **166**, 111628 (2025)
43. Wang, Y., Cao, C., Fu, K.F.X.X.Y.: Towards context-stable and visual-consistent image inpainting. arXiv preprint arXiv:2312.04831 (2023)
44. Wang, Y., Wu, Q., Zhang, G., Xu, D.: Learning 3d geometry and feature consistent gaussian splatting for object removal. In: European Conference on Computer Vision. pp. 1–17. Springer (2024)
45. Weber, E., Holynski, A., Jampani, V., Saxena, S., Snavely, N., Kar, A., Kanazawa, A.: Nerfiller: Completing scenes via generative 3d inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20731–20741 (2024)
46. Wu, C.H., Chen, Y.J., Chen, Y.H., Lee, J.Y., Ke, B.H., Mu, C.W.T., Huang, Y.C., Lin, C.Y., Chen, M.H., Lin, Y.Y., et al.: Aurafusion360: Augmented unseen region alignment for reference-based 360deg unbounded scene inpainting. In: Proceedings

of the Computer Vision and Pattern Recognition Conference. pp. 16366–16376 (2025)
47. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)
48. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
49. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. In: European conference on computer vision. pp. 162–179. Springer (2024)
50. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: Freedom: Training-free energy-guided conditional diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23174–23184 (2023)
51. Yu, Y., Zhang, L., Fan, H., Luo, T.: High-fidelity image inpainting with gan inversion. In: European Conference on Computer Vision. pp. 242–258. Springer (2022)
52. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in neural information processing systems **35**, 25018–25032 (2022)
53. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
54. Zhuang, J., Kang, D., Cao, Y.P., Li, G., Lin, L., Shan, Y.: Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. ACM Transactions on Graphics (ToG) **43**(4), 1–12 (2024)