Pose-Free Feed-Forward 3D Inpainting via Learnable Mask Attention and Support Token Refinement

Jingyi Pan[1] **Dan Xu**[2*] **Qiong Luo**[1,2*]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology

jpan305@connect.hkust-gz.edu.cn
{danxu,luo}@cse.ust.hk

Abstract

3D scene inpainting aims to recover missing or occluded regions in edited 3D scenes, while ensuring geometric and textural consistency. Existing approaches, however, typically require accurately calibrated camera poses, which restricts their applicability in casual, in-the-wild scenarios and introduces additional preprocessing overhead. To overcome this limitation, we present **FreeInpaint**, a novel feed-forward framework that generates complete and 3D-consistent scenes directly from unposed multi-view images with masked regions. At its core, FreeInpaint extends a 3D foundation model to propagate masked regions from a reference view to other unposed views, bridging 3D reconstruction and scene inpainting while preserving the model's native ability to recover camera poses and scene geometry. Our method addresses two key challenges in adapting feed-forward 3D foundation models to masked inputs. First, masked regions can corrupt cross-view correspondence reasoning, degrading pose estimation and geometry recovery. To address this, we introduce a Learnable Mask Attention mechanism that preserves the spatial anchoring of reliable observations while allowing masked regions to progressively absorb useful context in deeper layers. Second, under severe occlusions, a single forward pass often lacks sufficient appearance evidence for high-fidelity completion. Therefore, we propose a Support Token Refinement strategy, which injects diffusion-generated support evidence as confidence-weighted auxiliary tokens to refine under-observed regions while preserving the original spatial anchor. Extensive experiments across diverse datasets demonstrate that FreeInpaint achieves superior inpainting quality, eliminating the reliance on precomputed camera poses while keeping a fast inference speed. The project page is https://rorisis.github.io/FreeInpaint/.

1 Introduction

The increasing demand for 3D content creation in virtual reality, augmented reality, and interactive gaming has stimulated growing interest in 3D scene editing. Among various editing tasks, 3D scene inpainting aims to remove unwanted objects or complete missing regions while preserving geometric and textural consistency across multiple views. Compared with 2D inpainting, which operates on a single image, 3D inpainting must maintain multi-view consistency in both appearance and geometry, making it substantially more challenging.

Despite recent progress, existing 3D scene inpainting approaches [27, 26, 39, 21, 17, 7, 29, 41, 10, 46] suffer from a challenge: they rely on precisely calibrated camera poses. Typically, obtaining these

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

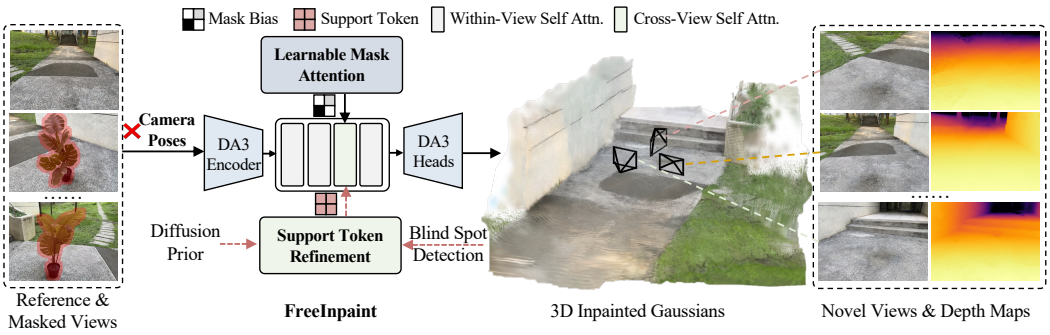


Figure 1: **FreeInpaint** takes a single reference image and multiple unposed masked source images as inputs. It jointly estimates the camera parameters and reconstructs a high-fidelity, multi-view consistent 3D Gaussian field.

poses requires running off-the-shelf Structure-from-Motion (SfM) [33], such as COLMAP, as a pre-processing step. However, this multi-stage approach is inherently time-consuming, restricting the efficiency and interactivity expected in user-facing inpainting applications. Furthermore, the reliance on SfM introduces a practical bottleneck for casual, in-the-wild captures. SfM pipelines are brittle and frequently fail when presented with common real-world conditions, such as sparse view collections, large untextured regions, or masked observations. If this initial pose estimation step fails to register cameras or yields inaccurate calibrations, the downstream 3D inpainting methods are fundamentally compromised, inevitably producing visual artifacts or distorted geometry.

To address this limitation, we present **FreeInpaint**, a novel feed-forward, reference-based framework that generates complete, 3D-consistent scenes directly from unposed image collections containing user-defined masked regions, as shown in Fig. 1. At its core, our method builds upon the architecture of a 3D foundation model [18], and adapts it for inpainting by conditioning on masked, unposed views while preserving the backbone's strong geometric priors.

However, adapting a reconstruction-oriented foundation model for inpainting introduces two primary technical challenges. First, masked regions can disrupt the multi-view correspondences essential for accurate pose and geometry estimation. If processed without constraints, these masked areas inevitably corrupt the spatial anchoring of the surrounding valid context. Second, a single feed-forward pass is often insufficient for handling severe occlusions where masked regions lack visual cues from the reference view, i.e., blind spots. In such cases, forcing the network to inpaint highly ambiguous regions in one step typically results in blurry or degraded textures.

To tackle these challenges, FreeInpaint introduces two complementary components. First, to prevent the corruption of pose estimation and geometry recovery under masked inputs, we introduce a **Learnable Mask Attention** mechanism. In the early layers of the network, masked regions carry little meaningful information; thus, valid tokens should not attend to them during the self-attention process. However, as the network deepens, these masked regions gradually aggregate contextual information from the reference view and surrounding tokens, eventually becoming useful for global reasoning. To model this dynamic transition, we apply a learnable mask bias term to the attention logits associated with masked tokens within each self-attention block, allowing the network to adaptively regulate how masked regions participate in cross-view reasoning at different depths. Second, we propose a **Support Token Refinement** strategy to improve blind-spot completion at inference time. When a single feed-forward pass remains under-confident in severely occluded regions, we inject generated support evidence as confidence-weighted support tokens. This allows the model to refine under-observed regions with complementary appearance cues while preserving the original reference-driven structure and multi-view consistency. Extensive experiments on SPIn-NeRF [27], 360-USID [41], and LLFF [24] datasets demonstrate that FreeInpaint achieves superior performance in both forward-facing and 360-degree scenes, all without requiring camera poses.

In summary, our key contributions are: (**i**) We propose FreeInpaint, a feed-forward framework capable of performing 3D scene inpainting directly from unposed, masked image collections. (**ii**) We introduce a Learnable Mask Attention mechanism that prevents masked regions from corrupting the foundation model's native pose and geometry estimation. (**iii**) We introduce a Support Token Refinement strategy that injects diffusion-generated support evidence as auxiliary tokens for blind-spot inpainting. (**iv**) We demonstrate the effectiveness of our method through extensive experiments on diverse datasets, highlighting its ability to eliminate the reliance on pre-computed camera poses.

2 Related Work

3D Scene Inpainting. The rapid progress in neural scene representations [25, 37, 12] has catalyzed significant advancements in 3D scene inpainting [27, 7, 40, 21, 5, 10]. The primary aim of 3D inpainting is to fill missing regions within a 3D scene, which includes tasks such as object removal and generating realistic textures and geometries to complete affected areas. Existing approaches can generally be classified into non-reference-based and reference-based methods. Non-reference-based techniques often leverage 2D foundation models like CLIP [31] or DINO [6] to capture 3D semantics for localized editing [43, 30, 49], or employ diffusion priors and Score Distillation Sampling (SDS) to iteratively inpaint missing content [27, 7, 42]. In contrast, reference-based methods [26, 39, 22, 21, 34, 41, 29, 46] utilize a provided reference image to propagate the desired texture and geometry across all viewpoints. However, most existing methods rely on computationally intensive per-scene optimization and accurate camera poses, typically obtained via Structure-from-Motion (SfM) [33], limiting their generalization to casual, sparse inputs. Unlike these methods, we propose FreeInpaint, a feed-forward and pose-free framework for 3D scene inpainting.

3D Feed-Forward Models. Recent years have witnessed a paradigm shift toward feed-forward 3D reconstruction models, which directly map 2D images to 3D representations without relying on per-scene optimization [45, 4]. A pioneering work in this domain is DUSt3R [38], which introduces a transformer-based architecture to process unposed image pairs and predict dense 3D pointmaps. Subsequent works [2, 15, 9, 47] have explored broader and more complex scenarios. Furthermore, VGGT [36] presents a unified architecture utilizing alternating intra-frame and inter-frame attention mechanisms across the entire sequence, enabling the joint prediction of camera poses, depth maps, and point correspondences. Following this trajectory, recent foundation models [11, 18], particularly Depth Anything 3 (DA3), employ a GS-DPT head to further extend the capability of feed-forward 3D reconstruction to high-fidelity novel view synthesis. However, these models inherently assume a fully observed scene and lack the generative capacity required for dense inpainting in severely occluded areas. In our work, we exploit the robust geometric priors of these feed-forward foundation models and adapt them to pose-free 3D scene inpainting.

3 Method

As illustrated in Fig. 2, given a single reference image I_{ref} and a set of source images $\{I^i_{\text{src}}\}_{i=1}^N$ with corresponding binary masks $\{M^i\}_{i=1}^N$ indicating regions to be inpainted, our goal is to reconstruct a comprehensive 3D scene representation $\mathcal{G}$ (parameterized as 3D Gaussians). Notably, we assume the camera poses for all inputs are unknown. The model must implicitly estimate relative poses while simultaneously inpainting geometry and texture within the masked regions of I_{src} to ensure consistency with I_{ref}, enabling high-fidelity novel view synthesis from arbitrary viewpoints.

This section is organized as follows: Sec. 3.1 introduces the preliminaries of Depth Anything 3 [18] and 3D Gaussian Splatting [12]. Sec. 3.2 details our learnable mask attention mechanism, which prevents masked regions from corrupting pose and geometry estimation. Sec. 3.3 describes the support token refinement for handling occlusions. Finally, Sec. 3.4 outlines the training objectives and optimization strategy.

3.1 Preliminary

Depth Anything 3. Our framework builds upon Depth Anything 3 (DA3) [18], a recent visual geometry foundation model designed to recover consistent 3D spatial representations from an arbitrary number of images. Given an input image set $\mathcal{I} = \{I_i\}_{i=1}^N$, DA3 employs a Vision Transformer backbone (*i.e.*, DINOv2 [28]) where each view is prepended with a learnable camera placeholder token to handle unposed scenarios. These tokens are processed via alternative attention blocks that switch between within-view and cross-view self-attention to aggregate multi-view consistency. For output, DA3 utilizes specialized heads to decode these refined features. A Dual-DPT head predicts dense depth and ray maps, while a lightweight camera head regresses explicit camera intrinsics and extrinsics from the refined camera tokens. In addition, a GS-DPT head predicts pixel-aligned 3D Gaussian parameters (i.e., opacity, rotation, scale and colors). This architecture enables robust feed-forward reconstruction and novel view synthesis from unposed images.

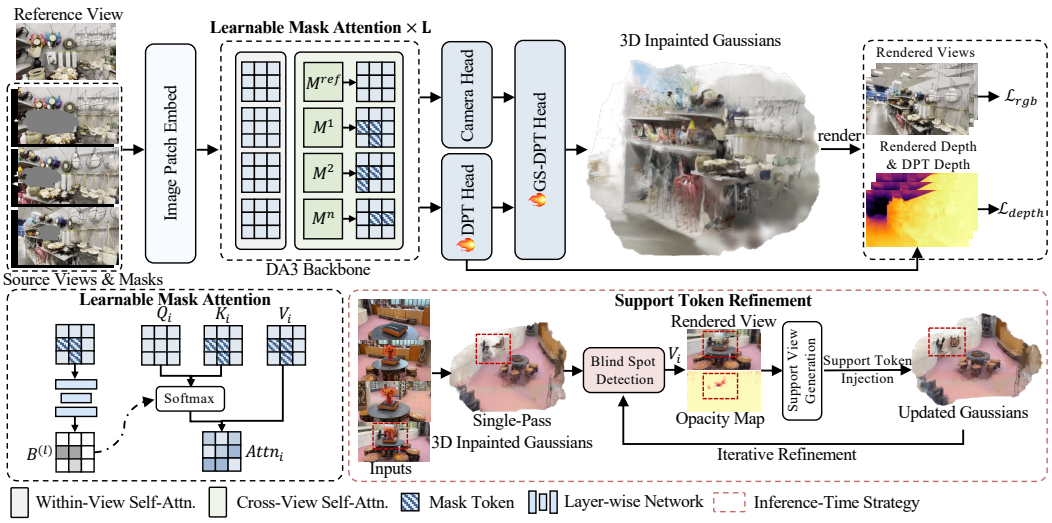


Figure 2: **The pipeline of our FreeInpaint.** Our method adapts DA3 for pose-free 3D inpainting by introducing learnable mask attention, which prevents masked regions from corrupting both pose and geometry estimation (Sec. 3.2). Additionally, we propose an inference-time support token refinement strategy to progressively inpaint content in occluded blind spots lacking visual cues (Sec. 3.3), while fine-tuning the DPT heads to adapt the model for inpainting tasks (Sec. 3.4).

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [12] represents a scene as a set of explicit anisotropic Gaussians, each parameterized by a center μ, opacity α, color c, rotation R, and scale S. Unlike implicit radiance fields [25, 37], 3DGS renders novel views by projecting Gaussians onto the image plane and alpha-compositing the projected primitives.

3.2 Learnable Mask Attention

A central challenge in pose-free 3D scene inpainting is that, although current 3D foundation models excel at jointly predicting camera poses and 3D geometry, they are highly vulnerable to masked regions. These regions contain no valid information and severely degrade the accuracy of pose and geometry estimation if processed naively.

A straightforward strategy is to apply a hard binary attention mask that permanently blocks masked regions. However, such a formulation is overly restrictive for inpainting. As information propagates through the network, some masked tokens may gradually become informative due to context aggregation from neighboring valid regions and other views. Permanently suppressing them prevents the model from exploiting this evolving evidence. Previous works [1, 16, 20] have explored dynamic or learnable masking mechanisms to modulate information propagation in image inpainting or more general vision tasks. However, MAT [16] relies on heuristic mask-update rules, MLLAM [1] predicts independent token weights at each layer, and Swin Transformer [20] employs relative-position biases and structural attention masks for shifted-window computation.

To address these, we design a learnable mask attention mechanism to encourage the model to preserve the spatial anchoring of reliable observations while limiting the corruption introduced by masked tokens. Specifically, it introduces a per-token, key-side attention bias initialized from explicit input occlusion masks and progressively updated across layers through learnable residuals. Let M^i denote the binary mask of source view i. We first downsample each mask to the patch resolution of the DA3 transformer and flatten it into a token-level vector $\tilde{M}^i$. Rather than treating $\tilde{M}^i$ as a fixed masking rule, we use it to initialize a sequence of learnable soft masks that evolve across the cross-view self-attention layers.

For each cross-view attention layer l, we derive a token-level mask bias $B^{(l)}$ from the corresponding source tokens and binary masks. This bias is initialized from the binary mask and then progressively

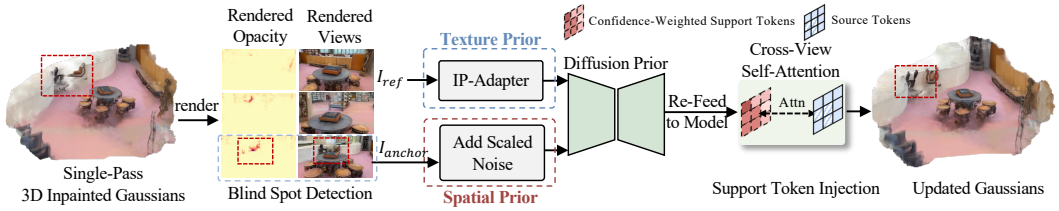


Figure 3: **Support Token Refinement.** We dynamically detect unobserved blind spots and designate them as anchor views. A diffusion prior is then employed to rapidly inpaint the missing content, while spatial and texture priors ensure multi-view consistency during the generation process.

refined through a lightweight layer-wise network g_l conditioned on the current source tokens:

$$P^{(0)} = \text{logit}\left(\epsilon + (1-2\epsilon)\tilde{M}\right), \tag{1}$$

$$P^{(l)} = P^{(l-1)} + \alpha g_l\left(F^{(l)}\right), \tag{2}$$

$$B^{(l)} = -\beta\sigma\left(P^{(l)}\right), \tag{3}$$

where $P^{(l)}$ denotes the token-level suppression state at layer l, $F^{(l)}$ denotes the current source tokens, $\sigma(\cdot)$ denotes the sigmoid function, $\text{logit}(p) = \log\frac{p}{1-p}$, and ϵ avoids infinite logits for binary mask values. Furthermore, given the query, key, and value Q, K, and V, the cross-view attention is computed as:

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}} + B^{(l)}\right)V, \tag{4}$$

where d is the key dimension. In this way, the model learns not only *where* masked regions should be suppressed, but also *how strongly* they should participate in cross-view reasoning at different depths.

3.3 Support Token Refinement

While the learnable mask attention mechanism effectively mitigates invalid context propagation, a single feed-forward pass remains insufficient for handling severe occlusions where source views lack visual cues from the reference image. We term these unobserved regions as blind spots. To address this challenge, we introduce a support token refinement strategy at inference time, as shown in Fig. 3. The key idea is to identify under-observed regions from the current 3D prediction, generate auxiliary support evidence for those regions, and re-inject this evidence into the model as confidence-weighted support tokens.

Blind Spot Detection. After an initial feed-forward pass, the predicted 3D Gaussian field is rendered back to the masked source views. For each masked source view i, we compute a visibility score over the masked region:

$$V_i = \frac{\sum\left(\alpha^i_{\text{render}} \odot M^i\right)}{\sum M^i + \epsilon}, \tag{5}$$

where α^i_{render} denotes the rendered opacity map and M^i is the corresponding binary mask. Low opacity inside the masked region indicates that the current 3D reconstruction lacks sufficient support in that area. We identify the view with the lowest visibility score as the anchor view for refinement.

Support View Generation. For the selected anchor view, we introduce a diffusion prior to generate a complementary support view conditioned on the current scene context. To preserve consistency with the existing reconstruction, we apply Stochastic Differential Editing (SDEdit) [23] using the previously rendered image I_{render} as the canvas. By adding noise proportional to a strength factor, we retain the underlying geometry, thereby introducing spatial prior. Simultaneously, we condition the inpainting on the original reference image I_{ref} via an IP-Adapter [44] to further inject the textural prior. The resulting support view generated by a diffusion model with both spatial and textural priors thus supplies complementary appearance cues for the blind spot while remaining consistent with the original reference image.

Support Token Injection. The generated support view is encoded into auxiliary tokens, which are then injected into the model in a confidence-weighted manner to refine under-observed regions. To

Table 1: Quantitative comparisons with per-scene optimization methods on SPIn-NeRF [27], 360-USID [41], and LLFF [24] datasets. Inference time is measured on a single A6000 GPU. We report the cumulative time for multi-stage pipelines, while our method requires only ∼0.4s for a single forward pass. *For scenes with severe occlusions, a dynamic Support Token Refinement is triggered, slightly increasing the time to ∼1.8s (see Sec. 4.2), which remains orders of magnitude faster than optimization baselines.

Methods	SPIn-NeRF [27]		360-USID [41]		LLFF [24]		Time↓
	PSNR ↑	LPIPS↓	PSNR↑	LPIPS↓	C-KID↓	C-FID ↓	
SPIn-NeRF [27]	16.32	0.4122	17.59	0.4126	0.5978	388.29	∼ 5h
NeRFiller [40]	<u>16.86</u>	0.4183	15.03	0.5036	<u>0.5835</u>	<u>360.02</u>	∼ 30m
InFusion [21]	13.99	0.5216	15.62	0.3716	0.6652	428.18	∼ 30m
GScream [39]	16.85	<u>0.3644</u>	16.29	0.4415	0.6300	409.16	∼ 2h
AuraFusion360 [41]	14.51	0.7077	<u>18.34</u>	<u>0.2982</u>	0.6378	402.18	∼ 1h
Ours	**17.79**	**0.2819**	**18.58**	**0.2526**	**0.5613**	**343.97**	**∼ 0.4s***

determine the reliability of these support tokens, we compute a confidence weight map. Instead of trusting the generated support view uniformly over the masked region, we prioritize true blind spots by combining the anchor mask with the rendered opacity:

$$C^{\text{sup}} = \text{Norm}(\mathcal{G}_\sigma(M^a \odot (1 - \alpha^a_{\text{render}})^\gamma)), \tag{6}$$

where M^a is the anchor mask, α^a_{render} is the rendered opacity map of the anchor view, $\mathcal{G}_\sigma$ denotes Gaussian smoothing, and $\text{Norm}(\cdot)$ rescales the response to $[0, 1]$. In this way, regions already explained by the current reconstruction receive low support confidence, while low-opacity blind spots receive high confidence. Finally, we resize to the token resolution and use it to modulate the attention given to the support tokens, implementing the confidence-weighted injection. Consequently, high-confidence support tokens from blind spots are effectively attended to, whereas low-confidence ones remain suppressed, allowing the model to refine blind spots without disturbing the original reference-driven structure.

3.4 Optimization

To effectively adapt the 3D foundation model from a reconstruction-based model to an inpainting-capable framework, we detail our fine-tuning strategy and training objectives.

Mask Generation. To simulate realistic occlusion patterns during training, we employ a hybrid masking strategy. For each training clip, we designate one frame as the reference view I_{ref}, while others serve as source views I_{src} or novel views I_{novel}. We generate masks M for I_{src} using two methods: a) *Random Box and Brush Masks*, which simulate arbitrary occlusions ranging from simple artifacts to large-scale missing regions; and b) *Geometric Masks*, where 2D masks are projected from the reference view to source views using the metric depth estimated by the original pre-trained DA3 [18]. This simulates physically consistent 3D occlusions, compelling the model to learn geometry-aware inpainting rather than simple 2D texture infilling.

Depth-Guided Self-Distillation. A major challenge in feed-forward 3D inpainting is recovering accurate geometry within occluded regions where photometric constraints are entirely absent. Without explicit geometric guidance, models tend to hallucinate textures onto incorrect depth planes. To mitigate this, we formulate the geometric reconstruction as a self-distillation process, leveraging the strong spatial priors embedded in the original foundation model. During training, we employ a frozen copy of the pre-trained DA3 as a teacher network. As the model processes a scene, we pass the unmasked source and novel views through this frozen teacher to generate high-quality depth maps, denoted as D_{prior}, on the fly. These serve as pseudo ground-truth. Simultaneously, our trainable model processes the masked images to predict the inpainted depth, D_{pred}. We enforce geometric consistency between the student's prediction and the teacher's prior via a masked distillation loss:

$$\mathcal{L}_{\text{depth}} = \|D_{\text{pred}} \odot M - D_{\text{prior}} \odot M\|_1, \tag{7}$$

where D denotes the normalized depth, and M is the binary mask.

Training Objectives. The overall training objective is formulated as a combination of a photometric loss and a multi-view depth consistency loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}, \tag{8}$$



Figure 4: Qualitative results with per-scene optimization methods on SPIn-NeRF [27], 360-USID [41], and LLFF [24] datasets. FreeInpaint successfully synthesizes high-fidelity and multi-view consistent novel views without requiring camera poses.

Table 2: Quantitative comparisons with feed-forward unposed methods on SPIn-NeRF [27], 360-USID [41], and LLFF [24] datasets.

Methods	SPIn-NeRF [27]			360-USID [41]			LLFF [24]	
	PSNR↑	LPIPS ↓	FID↓	PSNR↑	LPIPS↓	FID↓	C-KID↓	C-FID ↓
LAMA [35]+DA3	16.67	0.4034	196.01	17.76	0.3211	186.83	0.5629	361.75
MVInpainter [5]+DA3	16.79	0.3733	164.22	17.24	0.3891	275.06	0.5724	372.76
Ours	**17.79**	**0.2819**	**148.05**	**18.58**	**0.2526**	**199.61**	**0.5613**	**343.97**

where λ_{depth} is a balancing weight. The photometric loss $\mathcal{L}_{\text{rgb}}$ is computed using a MSE loss alongside an LPIPS perceptual penalty:

$$\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}. \tag{9}$$

To prevent the model from trivially overfitting input views, we compute the photometric loss $\mathcal{L}_{\text{rgb}}$ exclusively on held-out novel views. In contrast, the depth loss $\mathcal{L}_{\text{depth}}$ is evaluated on both the input views and the novel views to enforce 3D geometric consistency across the entire scene.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We utilize the DL3DV-10K dataset [19] for training, which offers large-scale real-world scenes with diverse camera trajectories. For evaluation, we employ three distinct benchmarks to assess performance across different scenarios: SPIn-NeRF [27], 360-USID [41], and LLFF [24]. SPIn-NeRF serves as a standard benchmark for object removal in forward-facing scenes with ground-

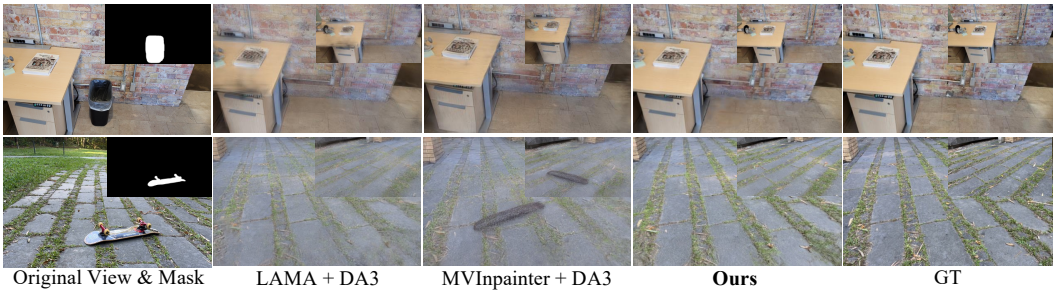


Figure 5: Qualitative results on SPIn-NeRF [27] and 360-USID [41] datasets.

Table 3: Quantitative comparisons of the object replacement task.

Methods	Category	$\text{CLIP}_{\text{dir}}\uparrow$	C-KID↓	C-FID ↓
DiGA3D [29]	Non-Ref.	0.1928	0.2607	169.08
NeRFiller [40]	Ref.	0.1917	0.2628	198.25
InFusion [21]	Ref.	0.1922	0.2647	236.75
Ours	Ref.	**0.1942**	**0.2589**	**153.25**

Table 4: Ablation study on each component on the 360-USID dataset [41].

Methods	PSNR↑	LPIPS↓	FID ↓
Baseline	15.58	0.7399	392.83
+ Fine-Tuning (Sec. 3.4)	16.96	0.2854	259.26
+ Mask Attn. (Sec. 3.2)	18.26	0.2628	209.86
+ Token Refine. (Sec. 3.3)	**18.58**	**0.2526**	**199.61**

truth backgrounds. The 360-USID dataset extends this to 360-degree scenes, providing reference images and ground truth to quantitatively evaluate inpainting quality behind removed objects. The LLFF dataset provides diverse real-world scenes with varying view numbers. Additional experiments on other datasets are provided in the Appendix.

Evaluation Metrics. Following prior works [41, 34, 46], we render novel views from the inpainted 3D representations for quantitative assessment. For datasets with available ground truth (SPIn-NeRF and 360-USID), we report PSNR and LPIPS [48]. For the LLFF dataset, which lacks ground-truth backgrounds for physically removed objects, we employ C-FID [8] and C-KID [3] to assess the visual quality and coherence of the inpainted regions. To ensure a focused evaluation, all metrics are computed exclusively within the bounding box defined by the object mask. Furthermore, for text-driven object replacement tasks, we additionally report the CLIP Text-Image Directional Similarity (CLIP_{dir}) [31] to evaluate semantic alignment.

4.2 Implementation Details

Training. Our framework is built on Depth Anything 3 (DA3-Giant) [18]. We optimize the model using the DL3DV-10K dataset [19], sampling 15-frame clips from long video sequences to form each training batch. We adopt a two-stage fine-tuning strategy: we first adapt the DPT and GS-DPT heads to masked scene inpainting, and then further optimize the Learnable Mask Attention modules together with cross-view attention layers. During training, we employ a hybrid masking strategy that mixes random masks and geometric masks at a ratio of 0.4:0.6. Optimization is performed on 8 NVIDIA RTX A6000 GPUs with a per-GPU batch size of 2 and a gradient accumulation step of 2. All input images are resized to a resolution of 280×504.

Inference. During inference, we leverage PowerPaint v1 [50] as a 2D diffusion prior to inpaint the reference image if a clean reference view is initially unavailable. We utilize the Language-based Segment Anything model (Lang-SAM) [14] to extract specific object masks. Our support token refinement is dynamically triggered when the rendered visibility score inside the masked region falls below 0.97, with a maximum of 3 total feed-forward passes.

4.3 Main Results

Comparison with Per-Scene Optimization Methods. Tab. 1 presents a quantitative comparison on the SPIn-NeRF [27], 360-USID [41], and LLFF [24] datasets. For our evaluation, FreeInpaint operates on highly sparse and uncalibrated data, utilizing only 4 unposed inputs (1 reference and 3 masked source images) for the SPIn-NeRF and LLFF datasets, and 8 unposed inputs (1 reference and 7 masked source images) for 360-USID dataset. Despite this challenging setting, FreeInpaint consistently outperforms prior methods across most metrics while maintaining rapid inference speeds. Crucially, our feed-forward design enables inference in just 0.4s via a single forward pass, a

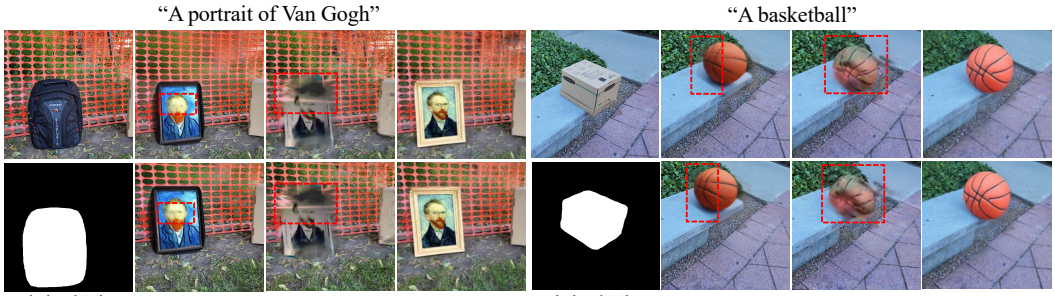


Figure 6: Qualitative results of the object replacement task. We compare FreeInpaint with DiGA3D [29] and InFusion [21] methods.

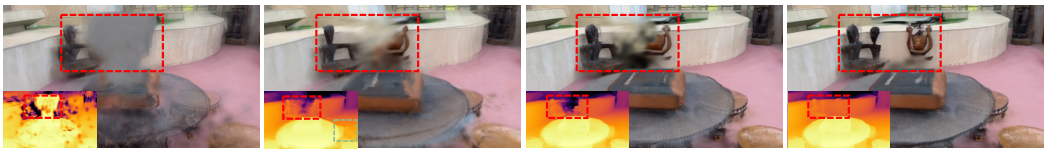


Figure 7: **Qualitative ablation study on the 360-USID dataset.** Fine-tuning (+ Fine-Tuning) alone still yields blurriness. Integrating our learnable mask attention (+ Mask Attn.) significantly sharpens structural details by suppressing invalid features. The support token refinement (+ Token Refine.) further recovers high-fidelity textures in occluded regions.

Table 5: Effect of Learnable Mask Attention. We compare different mask-aware attention strategies under masked inputs.

Mask Attention	PSNR↑	LPIPS↓	ATE↓	RPE_t↓	RPE_r↓
No mask	16.96	0.2854	0.0462	0.0803	0.7766
Hard mask	17.97	0.2581	0.0377	0.0764	0.7430
Learnable mask	**18.58**	**0.2526**	**0.0340**	**0.0644**	**0.7069**

significant speedup compared to the 30 minutes to 5 hours required by optimization-based approaches. Qualitative results in Fig. 4 further validate its ability to generate high-fidelity, consistent renderings under sparse, unposed conditions.

Comparison with Feed-Forward Unposed Methods. Given the absence of existing feed-forward, pose-free 3D inpainting frameworks, we formulate LAMA [35]+DA3 and MVInpainter [5]+DA3 for uncalibrated feed-forward comparisons. As reported in Tab. 2, our framework achieves superior performance across all evaluated metrics. As shown in Fig.5, LAMA+DA3 fails due to multi-view inconsistency in 2D inpainting, which corrupts pose estimation and subsequent 3D reconstruction. Even MVInpainter+DA3 exhibits noticeable pose misalignments. These results underscore the effectiveness of our unified approach, which jointly addresses inpainting and pose estimation with sparse and masked inputs, avoiding the error accumulation inherent in cascaded pipelines.

Object Replacement. In Tab. 3, FreeInpaint consistently outperforms both the reference-free approach [29] and the reference-based methods [40, 21]. As illustrated in Fig. 6, previous methods struggle with noticeable artifacts and blurriness within the inpainted regions. In contrast, FreeInpaint synthesizes clean, high-fidelity textures that blend seamlessly with the surrounding 3D context.

4.4 More Analysis

Ablation on Main Components. We conduct ablation studies on the 360-USID [41] dataset to validate the effectiveness of each proposed component. As shown in Tab. 4 and Fig. 7, directly applying the original pre-trained DA3 as a baseline produces poor results due to the domain gap between standard reconstruction and scene inpainting. Fine-tuning improves basic novel view synthesis, but the inpainted regions remain noticeably blurry, as masked regions can still corrupt cross-view pose and geometry estimation. Introducing learnable mask attention substantially improves all quantitative metrics and visibly reduces blurriness, confirming its ability to preserve spatial anchoring while adaptively incorporating useful context from masked regions. Finally, support token refinement

Table 6: View-count scalability on the SPIn-NeRF [27] and 360-USID [41] datasets. "Total Input Views" includes one reference view and the remaining masked source views.

Total Input Views	SPIn-NeRF			360-USID		
	PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓
4	17.79	0.2819	148.05	17.19	0.3148	226.30
8	18.32	0.2702	**143.88**	18.58	0.2526	199.61
16	18.33	0.2688	145.97	18.66	**0.2512**	196.67
32	**18.47**	**0.2683**	146.44	**18.69**	0.2519	**194.32**

Table 7: Sensitivity to the diffusion-generated reference and reference view selection on the SPIn-NeRF [27] dataset.

Metric	Diffusion-generated reference	Reference view selection
PSNR↑	17.68 ± 0.16	17.61 ± 0.24
LPIPS↓	0.2808 ± 0.0168	0.2867 ± 0.0144
FID↓	149.85 ± 7.55	152.18 ± 12.76

further improves inpainting quality, especially for blind spots lacking sufficient reference-view cues, by progressively injecting high-fidelity auxiliary evidence.

Effect of Learnable Mask Attention. A key contribution of our work is the learnable mask attention mechanism, which not only improves inpainting quality but also enhances pose estimation robustness. To quantify this, we evaluate the camera rotation and translation accuracy using Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) on the 360-USID dataset [41]. As shown in Tab. 5, both hard and learnable masking outperform the variant without mask attention, confirming the importance of suppressing invalid masked regions. Moreover, learnable mask attention further improves ATE, RPE_t, and RPE_r over fixed hard masking, showing that adaptive mask-aware routing better preserves spatial anchoring while allowing masked regions to absorb useful cross-view context.

Scalability to More Views. To evaluate view-count scalability, we vary the total number of input views from 4 to 32 on SPIn-NeRF [27] and 360-USID [41] datasets, while keeping all other evaluation settings unchanged. As shown in Tab. 6, FreeInpaint generally benefits from additional input views. The reconstruction quality improves as more unposed source observations are introduced, while the perceptual metrics remain stable with only minor fluctuations. These results show that FreeInpaint is not restricted to the original sparse-view setting and can effectively utilize denser collections of unposed inputs.

Sensitivity to the Reference View. We evaluate the sensitivity of FreeInpaint to variations in the reference input on the SPIn-NeRF [27] dataset. Specifically, we consider two sources of variation: 1) diffusion-inpainted references generated with different sampling seeds for a fixed reference viewpoint in each scene, and 2) different choices of the reference viewpoint, which provide varying scene coverage and visible context. We report the mean and standard deviation over ten diffusion seeds and four reference-view choices, respectively. As shown in Tab. 7, FreeInpaint remains relatively stable under both types of variation, with only moderate changes in reconstruction and perceptual quality. For each setting, we first compute the metrics over the complete test set and then report the mean and standard deviation across the corresponding diffusion seeds or reference-view choices.

5 Conclusion

In this paper, we present FreeInpaint, a feed-forward framework for pose-free 3D scene inpainting from uncalibrated and masked image collections. By adapting a 3D foundation model to masked inputs, FreeInpaint enables joint pose estimation, geometry recovery, and scene inpainting without requiring pre-computed camera poses. To prevent masked regions from corrupting spatial anchoring, we introduced a learnable mask attention mechanism that adaptively regulates the participation of masked tokens in cross-view reasoning. To further handle severe occlusions with insufficient reference cues, we proposed an inference-time support token refinement strategy that injects diffusion-generated evidence as auxiliary tokens for progressive inpainting. Extensive experiments demonstrate that FreeInpaint achieves high-quality, multi-view consistent inpainting with fast feed-forward inference, providing an efficient and robust solution for 3D scene inpainting in unposed settings.

Acknowledgments and Disclosure of Funding

The work of Qiong Luo and Jingyi Pan was supported by a startup grant from the Hong Kong University of Science and Technology (Guangzhou). The work of Dan Xu was supported by the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321, ITF PRP/046/24FX, Science & Technology Cooperation Program of Shandong under grant No. SDST26EG01, SAIL Research Project, and HKUST-Zeekr Collaborative Research Fund.

References

[1] Wayner Barrios and SouYoung Jin. Multi-layer learnable attention mask for multimodal tasks. *arXiv preprint arXiv:2406.02761*, 2024.

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[4] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

[5] Chenjie Cao, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing. In *NeurIPS*, 2024.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[7] Honghua Chen, Chen Change Loy, and Xingang Pan. Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior. In *CVPR*, 2024.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

[9] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[10] Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3d gaussian inpainting with depth-guided cross-view consistency. In *CVPR*, 2025.

[11] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):139–1, 2023.

[13] Umar Khalid, Hasan Iqbal, Azib Farooq, Jing Hua, and Chen Chen. 3dego: 3d editing on the go! In *ECCV*, 2024.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

[15] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024.

[16] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *CVPR*, 2022.

[17] Chieh Hubert Lin, Changil Kim, Jia-Bin Huang, Qinbo Li, Chih-Yao Ma, Johannes Kopf, Ming-Hsuan Yang, and Hung-Yu Tseng. Taming latent diffusion model for neural radiance field inpainting. In *ECCV*, 2024.

[18] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[19] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[21] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024.

[22] Yiren Lu, Jing Ma, and Yu Yin. View-consistent object removal in radiance fields. In *ACM MM*, 2024.

[23] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[24] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *TOG*, 2019.

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[26] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Reference-guided controllable inpainting of neural radiance fields. In *ICCV*, 2023.

[27] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023.

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[29] Jingyi Pan, Dan Xu, and Qiong Luo. Diga3d: Coarse-to-fine diffusional propagation of geometry and appearance for versatile 3d inpainting. In *ICCV*, 2025.

[30] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, 2024.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[32] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021.

[33] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[34] Zhihao Shi, Dong Huo, Yuhongze Zhou, Yan Min, Juwei Lu, and Xinxin Zuo. Imfine: 3d inpainting via geometry-guided multi-view refinement. In *CVPR*, 2025.

[35] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022.

[36] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

[37] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[38] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[39] Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3d geometry and feature consistent gaussian splatting for object removal. In *ECCV*, 2024.

[40] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *CVPR*, 2024.

[41] Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huang, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, et al. Aurafusion360: Augmented unseen region alignment for reference-based 360 unbounded scene inpainting. In *CVPR*, 2025.

[42] Hanyuan Xiao, Yingshu Chen, Huajian Huang, Haolin Xiong, Jing Yang, Pratusha Prasad, and Yajie Zhao. Localized gaussian splatting editing with contextual awareness. In *WACV*, 2025.

[43] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[44] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[45] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, 2023.

[46] Junqi You, Chieh Hubert Lin, Weijie Lyu, Zhengbo Zhang, and Ming-Hsuan Yang. Instainpaint: Instant 3d-scene inpainting with masked large reconstruction model. *arXiv preprint arXiv:2506.10980*, 2025.

[47] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024.

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[49] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, 2024.

[50] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *ECCV*, 2024.

A Technical Appendices and Supplementary Material

A.1 Demo Videos

To further demonstrate the multi-view consistency and geometric stability of our method, we have included demo video results in our project page.

A.2 Additional Dataset Details

In addition to the datasets discussed in the main paper, we incorporate the IMFine dataset [34] for extended ablation studies and qualitative evaluation in the appendix. IMFine provides complex scenes with challenging view variations, making it particularly well-suited for evaluating the robustness of our method.

A.3 Additional Implementation Details

We apply learnable mask attention only to the cross-view attention blocks of DA3. For each cross-view attention block, we instantiate an independent lightweight layer-wise network to predict a learnable mask bias that suppresses invalid masked information from corrupting pose and geometry estimation. The network consists of a 3×3 convolution, a depthwise 3×3 convolution, and a 1×1 projection layer, with GELU activations in between. The final projection layer is zero-initialized such that the learnable mask is initialized as the binary-mask prior and then gradually adapted during training. We set the hidden dimension to 8, the residual scaling factor α to 1.5, and the suppression strength β to 10.

A.4 Additional Details of Feed-Forward Baselines

For the LaMa+DA3 and MVInpainter+DA3 baselines in Tab. 2, we use the official pretrained checkpoints of LaMa [35] (`big-lama`) and MVInpainter [5] (`MVInpainter-F-512`) without any task-specific fine-tuning. LaMa is applied independently to each masked input view, whereas MVInpainter jointly processes the masked multi-view inputs using its official inference pipeline. The resulting completed views are then directly passed to the pretrained DA3-Giant model [18] to estimate the camera parameters and reconstruct 3D Gaussian scenes. All components of these baselines are kept frozen, and we use the officially recommended inference settings without tuning on the test sets.

For the comparison, all methods use the same input-view sets, masked inputs, image resolution, and pose-free evaluation protocol. Neither ground-truth nor externally estimated camera poses are provided as model inputs. These baselines instantiate two straightforward cascaded paradigms, namely "2D inpainting + pose-free 3D reconstruction" and "multi-view inpainting + pose-free 3D reconstruction." Their purpose is to examine whether combining an off-the-shelf image inpainter with a generic 3D foundation model is sufficient for this task. In contrast, FreeInpaint directly adapts the 3D foundation model to masked observations and performs mask-aware cross-view reasoning for camera estimation, geometry recovery, and 3D scene inpainting.

A.5 More Analysis

A.5.1 Discussion on Support Token Refinement

We further analyze the design choices and parameter settings of our support token refinement strategy. Since this mechanism is specifically triggered to handle severe occlusions where the reference view lacks sufficient visual cues, we conduct the following ablation studies on six challenging scenes selected from the 360-USID [41] and IMFine [34] datasets.

Ablation on Support View Generation. When refinement is applied, we find that leveraging the IP-Adapter without SDEdit repairs the basic geometry of the bottle but introduces structural artifacts due to multi-view inconsistencies in the generated anchor view. In addition, using SDEdit without the IP-Adapter maintains spatial structure but generates incorrect textures that fail to match the original reference. By combining both priors, our full method successfully repairs geometric distortions while maintaining textural identity without introducing extra artifacts.



Figure 8: **Ablation on the Support View Generation.** Our full method effectively reconstructs the occluded regions with high fidelity, preventing the geometric or textural inconsistencies seen in the ablated baselines.

Table 8: **Ablation studies on parameters of support token refinement.** We evaluate the impact of different iteration numbers and visibility score thresholds (τ) on the 360-USID [41] and IMFine [34] datasets. **Note that the reported metrics are averaged across both datasets.** "Iters." refers to the total number of feed-forward passes through our model (e.g., 3 Iters. equals 1 initial pass plus up to 2 refinement loops). The default settings used in our main experiments are highlighted in gray.

Max Iters.	PSNR ↑	LPIPS ↓	Time ↓
1 (Initial pass)	18.02	0.2738	0.41
2	18.29	0.2703	1.12
3	18.46	0.2623	1.83
4	18.48	0.2620	2.54

(a) Impact of maximum iteration numbers.

Threshold τ	PSNR ↑	LPIPS ↓	Avg. Iters ↓
0.93	18.12	0.2702	1.33
0.95	18.25	0.2689	1.50
0.97	**18.46**	**0.2623**	2.67
0.99	18.30	0.2629	4.33

(b) Impact of visibility score threshold.

Ablations on Iterative Numbers. As shown in Tab. 8 (a), relying purely on a single-pass inference (1 iter) without support token refinement yields relatively lower metrics, as the feed-forward model inherently struggles with complete blind spots in scenes with large view variations. Increasing the maximum iterations to 2 or 3 (i.e., triggering 1 or 2 refinement loops) significantly improves PSNR and LPIPS, demonstrating the effectiveness of injecting generative support evidence for unseen geometries. However, beyond 3 iterations, the performance yields marginal gains (e.g., only $+0.02$ dB in PSNR from iter 3 to 4) while incurring a linear increase in computational overhead. Therefore, we set 3 as the maximum iteration limit to offer the optimal trade-off between high-fidelity inpainting quality and inference efficiency.

Ablations on Threshold of Visibility Score. In Tab. 8 (b), we further analyze the sensitivity of the visibility score threshold τ. A lower threshold fails to trigger the refinement for severe blind spots, leading to under-completion. In contrast, an excessively high threshold (e.g., $\tau = 0.99$) forces the model to heavily rely on 2D diffusion outputs, overwriting well-reconstructed regions. This over-refinement inevitably breaks the inherent 3D consistency of the foundation model and unnecessarily inflates the average iteration count (increasing to 4.33 iters). Therefore, we empirically set $\tau = 0.97$, which yields the optimal balance by accurately locating true geometric holes without disrupting reliable multi-view observations.

Effect of the LCM-based Diffusion Prior on Support Token Refinement. As shown in Tab. 9, we evaluate the impact of the LCM-based diffusion prior on three scenes from the 360-USID dataset [41] that require additional confidence-based support token refinement. We compare our 4-step LCM pipeline against a standard 50-step diffusion baseline. For the runtime analysis, we measure a single refinement iteration, which encompasses two forward passes of FreeInpaint. Notably, the LCM prior reduces the diffusion process time to just 0.3s, demonstrating a significant speedup over the 1.6s required by the standard baseline. Furthermore, the LCM-based approach yields better PSNR and LPIPS scores on the tested scenes. These results demonstrate the effectiveness of integrating an LCM-based prior into the support token refinement strategy.

Discussion on Fine-Tuning Strategy. We further analyze the impact of depth-guided self-distillation within our fine-tuning strategy on the SPIn-NeRF [27] dataset. As illustrated in Fig. 9, fine-tuning without depth-guided self-distillation leads to noticeable geometric artifacts, such as unnatural holes or flattened surfaces within the inpainted regions. In contrast, introducing depth-guided self-distillation effectively constrains the 3D geometry. This not only yields artifact-free geometric inpainting but also significantly improves the texture quality. Because accurate underlying geometry provides a robust foundation for optimizing the 3D Gaussians, it naturally translates to higher-fidelity texture rendering. The fine-tuning with depth-guided self-distillation demonstrates better overall visual quality compared to the baseline lacking depth constraints.

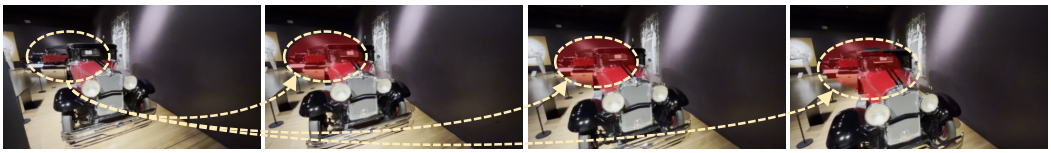


Baseline w/o depth-guided w/ depth-guided GT

Figure 9: Visualization of Fine-Tuning Strategy on the SPIn-NeRF [27] dataset.



a) Random Box and Brush Masks



b) Geometric Masks

Figure 10: Visualization of Mask Generation Strategy.

Effect of Different Mask-Aware Attention Strategies. We further extend the ablation study in Tab. 5 by implementing MAT-style [16] and MLLAM-style [1] masking strategies under the same DA3 backbone, training protocol, and Support Token Refinement, while varying only the mask-attention mechanism. The MAT-style variant uses rule-based dynamic binary mask propagation, whereas the MLLAM-style variant applies layer-wise attention modulation without initialization from the input binary mask. As shown in Tab. 11, both variants outperform the baseline without mask attention, while our mask-initialized, feature-conditioned learnable mask attention strategy achieves the best performance in both scene inpainting and camera pose estimation.

Table 9: **Effect of the LCM-based diffusion prior** on support view generation. $(\cdot)$ denotes the time required for two forward passes of FreeInpaint.

Methods	PSNR ↑	LPIPS ↓	Time ↓
w/o LCM (50 steps)	18.22	0.2736	1.6s (+0.8s)
w/ LCM (4 steps)	**18.67**	**0.2689**	**0.3s (+0.8s)**

Table 10: **Ablation studies on Mask Generation Strategy**, evaluated on the SPIn-NeRF dataset [27].

Mask Strategy	PSNR ↑	SSIM ↑	LPIPS ↓
Random Box and Brush Only	17.18	0.3652	0.3022
Geometric Only	17.71	0.3835	0.2877
Hybrid (Ours)	**17.79**	**0.3869**	**0.2819**

A.6 Limitations and Future Work

While FreeInpaint establishes a robust framework for pose-free 3D scene inpainting, it has some limitations. Primarily, our support token refinement relies on 2D diffusion models to inpaint blind spots, making the final 3D quality inherently bounded by the performance of the 2D prior. Future research could integrate native 3D or multi-view consistent video diffusion priors to ensure more stable geometric consistency. Furthermore, while our method excels with sparse, unposed collections, extreme cases involving only 1 or 2 restricted views of a complex 360-degree scene remain challenging (In Fig. 11) due to error accumulation during support token refinement. To address this, future iterations could incorporate a persistent spatial memory mechanism to maintain long-term geometric consistency across arbitrary refinement steps. In addition, support token refinement could be further extended to sparse-view reconstruction and novel view synthesis, where insufficiently observed regions could be identified and refined using video diffusion priors.

A.7 More Results

Additional Comparisons. We additionally compare FreeInpaint with InstaInpaint [46], a feed-forward 3D scene inpainting method that requires camera poses as input. As shown in Tab. 12,

Table 11: Effect of different mask-aware attention strategies under the same DA3 backbone and training protocol.

Mask Attention	PSNR↑	LPIPS↓	ATE↓	RPE_t↓	RPE_r↓
MAT-style mask [16]	18.13	0.2620	0.0361	0.0743	0.7459
MLLAM-style mask [1]	18.06	0.2651	0.0374	0.0775	0.7468
Our Learnable mask	**18.58**	**0.2526**	**0.0340**	**0.0644**	**0.7069**

Table 12: View-count scalability comparison with InstaInpaint [46] on SPIn-NeRF [27] and 360-USID [41]. "Total Input Views" includes one reference view and the remaining masked source views.

Total Input Views	Method	SPIn-NeRF PSNR↑	SPIn-NeRF LPIPS↓	SPIn-NeRF FID↓	360-USID PSNR↑	360-USID LPIPS↓	360-USID FID↓
4	InstaInpaint	16.89	0.2844	149.11	14.66	0.4934	259.39
	FreeInpaint (Ours)	**17.79**	**0.2819**	**148.05**	**17.19**	**0.3148**	**226.30**
8	InstaInpaint	16.72	0.3013	144.31	15.60	0.4707	228.08
	FreeInpaint (Ours)	**18.32**	**0.2702**	**143.88**	**18.58**	**0.2526**	**199.61**
16	InstaInpaint	17.04	0.3630	168.35	15.51	0.5012	265.28
	FreeInpaint (Ours)	**18.33**	**0.2688**	**145.97**	**18.66**	**0.2512**	**196.67**
32	InstaInpaint	17.12	0.4186	239.05	14.58	0.5330	268.26
	FreeInpaint (Ours)	**18.47**	**0.2683**	**146.44**	**18.69**	**0.2519**	**194.32**

Table 13: Quantitative evaluation on unseen in-the-wild scenes from GS25 [13] and Co3D [32].

Method	Pose Input	GS25 [13] C-KID↓	GS25 [13] C-FID↓	Co3D [32] C-KID↓	Co3D [32] C-FID↓
LaMa [35]+DA3	✗	0.3030	305.23	0.3870	360.72
MVInpainter [5]+DA3	✗	0.3624	356.07	0.3975	361.37
InstaInpaint [46]	✓	0.3406	330.84	0.3568	330.41
FreeInpaint (Ours)	✗	**0.2889**	**292.28**	**0.3295**	**325.44**



Figure 11: **Additional Visualization on the NeRFiller [40] dataset.** As illustrated, in extreme scenarios where highly sparse input views are combined with large-scale masks, the model struggles to retrieve sufficient contextual information. This occasionally degrades the inpainting quality and introduces some visual artifacts.

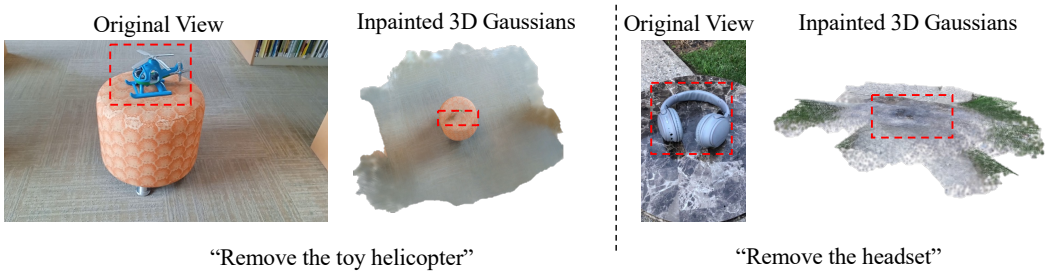


Figure 12: Additional Qualitative Results on the GS25 dataset [13] under unposed 8-view in-the-wild settings.

FreeInpaint consistently outperforms InstaInpaint across both datasets and all evaluated metrics. As further illustrated in Fig. 13, FreeInpaint benefits from additional input views, with reconstruction quality improving while perceptual quality remains stable. In contrast, InstaInpaint does not consistently benefit from denser inputs, with LPIPS and FID degrading notably at 16 and 32 views.

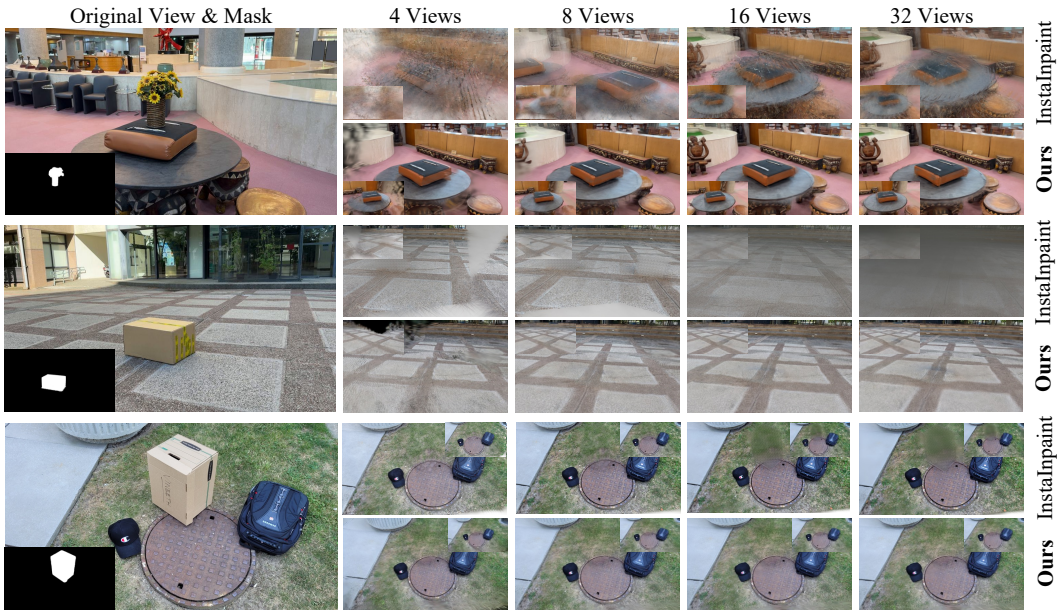


Figure 13: Additional qualitative results on view-count scalability.

These results demonstrate that FreeInpaint can more robustly exploit denser collections of unposed observations without requiring camera poses as model inputs.

To further evaluate generalization to in-the-wild scenes, we compare LaMa [35]+DA3, MVInpainter [5]+DA3, InstaInpaint [46], and FreeInpaint on 10 GS25 [13] scenes and 15 Co3D [32] scenes. For each scene, all methods use the same eight input views, consisting of one diffusion-inpainted reference image and seven masked source views. Since these datasets do not provide ground-truth backgrounds after object removal, we adopt the same no-reference evaluation protocol used for LLFF and compute C-KID and C-FID within the object-mask bounding boxes across all evaluation views. As shown in Tab. 13, FreeInpaint achieves the best results across all four metrics, providing further evidence of its generalization to unseen casual captures.

More Visual Results. In this section, we present additional visual results on the SPIn-NeRF [27], LLFF [24], and IMFine [34] datasets. As illustrated in Fig. 14, we show rendered views and rendered depth maps across diverse scenes, demonstrating that our method achieves multi-view consistent texture and geometry. We further evaluate our method on the in-the-wild GS25 dataset [13] and Co3D [32] dataset under the unposed setting with eight input views. As shown in Figs. 12 and 15, FreeInpaint remains robust across diverse in-the-wild scenes.

A.8 Impact Statement

Our work focuses on the field of 3D generative models with important ethical implications, including the risk of generating misleading or manipulated visual content. We recognize these challenges and affirm our commitment to responsible research and development. While this technology holds promise for empowering creative professionals and democratizing 3D content creation, we are dedicated to proactively addressing ethical concerns in its deployment.

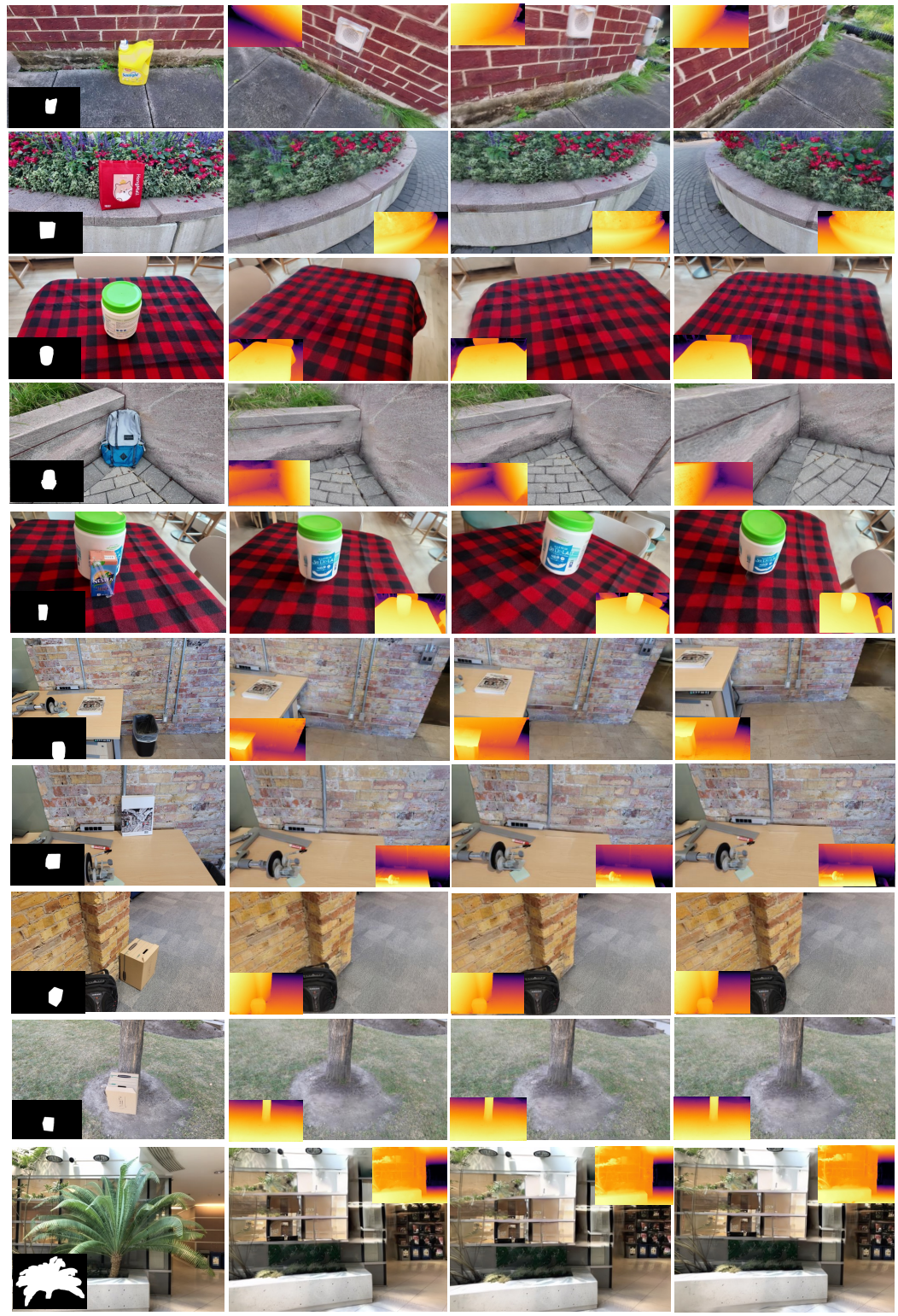


Figure 14: Additional Qualitative Results on the SPIn-NeRF [27], LLFF [24], and IMFine [34] datasets.

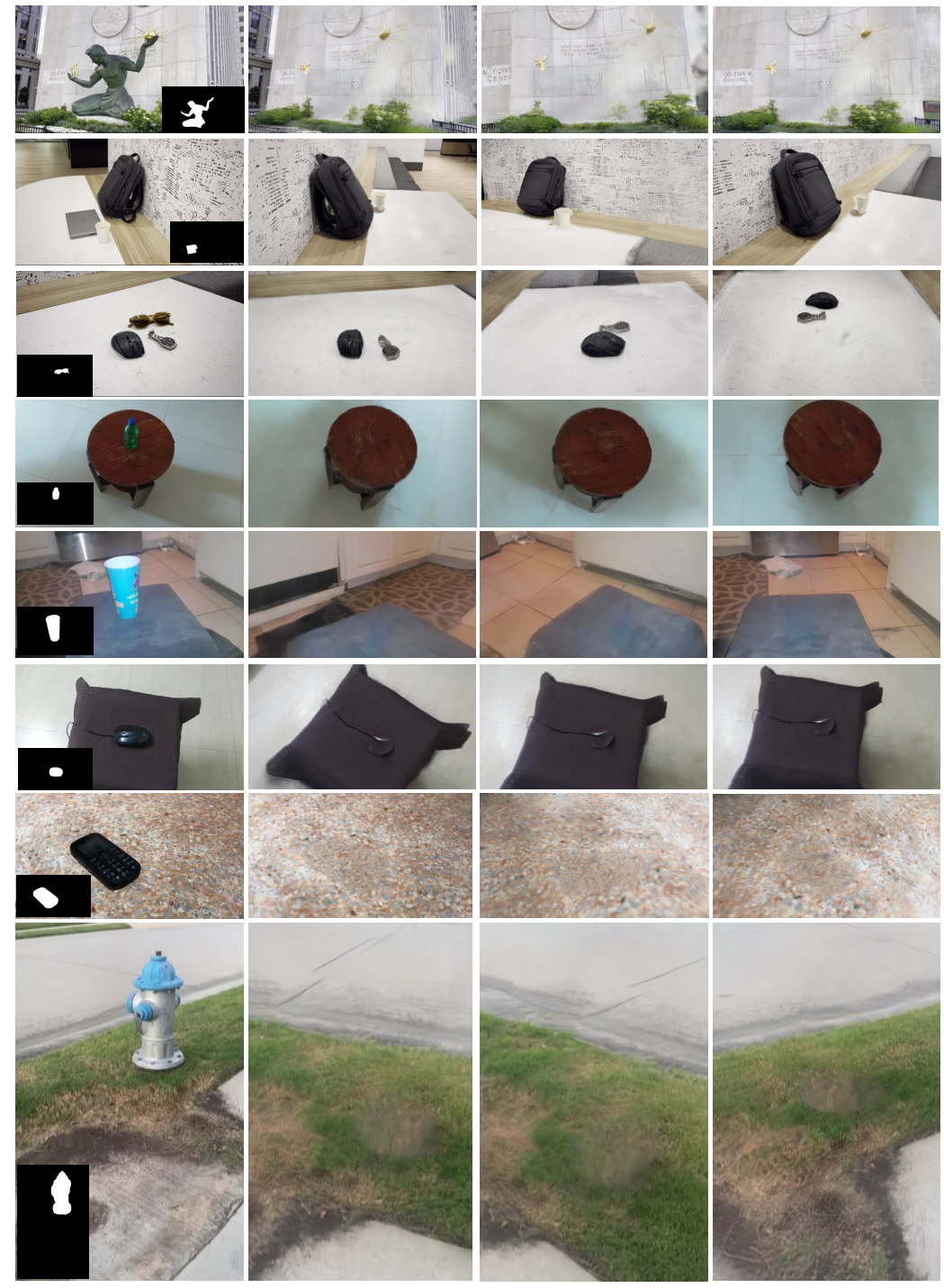


Figure 15: Additional Qualitative Results on GS25 [13] and Co3D [32] datasets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We outline the paper's objective, offer a brief overview of our methodology, and summarize our results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present an analysis of our constraints along with the future work in the Appendix (Section A.6).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our method does not focus on theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the implementation details in Section 4.2 and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We'll release our full code to the public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We offer the dataset usage and implementation details, encompassing hyperparameters, models, and other relevant information, in Section 4 and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our approach is applied across various scenes, with average scores reported in the experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details of computer resources in Section 4, showing the implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include an Impact Statement in our Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release scraped datasets or general-purpose generative models with high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We employ MIT-licensed datasets and open-source models that have been extensively utilized in previous research publications.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper doesn't release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments, user studies, or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.