WINGS: REFERENCE-FREE GAUSSIAN SPLATTING INPAINTING WITH 3D-NATIVE GENERATIVE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inpainting 3D Gaussian Splatting scenes, a key challenge in 3D editing, requires generating plausible content within a masked region of 3D space. Prior approaches rely on 2D diffusion models to produce one or several inpainted reference views, making them susceptible to challenges associated with multi-view inconsistency and lengthy optimization times. Departing from these approaches, we introduce a reference-free Gaussian splatting inpainting method operating natively in 3D. Our method leverages the embedding space of a large, pre-trained 3D prior, combined with a structure completion network to feed a generative prior which reconstructs the missing region's geometry and appearance. Performing content generation entirely in 3D, it avoids the need to reconcile inconsistencies of multiple inpainted reference images, and is faster than related 2D-based methods. We demonstrate the effectiveness of our method qualitatively and quantitatively, through extensive experiments and a user study. To the best of our knowledge, this work is the first Gaussian splatting inpainting method to operate in the learned representation space of a 3D-native generative prior without relying on inpainted reference views.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become the de facto standard for scene reconstruction and novel view synthesis due to its high fidelity, explicit representation and efficient rendering. As a consequence, many recent works focus on editing 3DGS assets and scenes, often with the goal of changing or removing objects from the scene. Object removal, specifically, leads to an interesting problem: parts of the scene that were previously hidden (e.g., the tabletop on which a vase had rested before removal) become disoccluded and are now visible, but were neither observed nor reconstructed during the initial 3DGS training, leading to empty holes and artifacts. Such areas now need to be inpainted: an algorithm needs to synthesize a plausible 3D continuation of surrounding surfaces and appearance that remains coherent under novel views, a problem that becomes especially difficult when the original 2D observations are no longer available.

Earlier existing 3DGS inpainting methods use 2D image diffusion models to provide one or several 2D-inpainted reference views of the scene, which are then optimized back into the 3D representation through differentiable rendering. They therefore solve a 3D-completion problem indirectly, by first hallucinating 2D projections and subsequently optimizing a 3D representation to explain them. Since the generative prior operates in 2D, the inpainted content often is not multiview-consistent, and it requires special care to consolidate these disagreements into 3D, often leading to blurry artifacts in and around the inpainted regions. Additionally, generating multiple inpainted reference views and re-optimizing the scene introduces significant computational overhead, leading to long inpainting times (Wu et al., 2025b) and motivating the reduction of this complexity by operating directly in 3D.

Recent large-scale 3D generative models provide a compelling alternative: rather than generating 2D content that is subsequently lifted into 3D, their learned priors enable reasoning about the missing content directly in 3D. Motivated by this observation, we introduce a reference-free approach to 3DGS inpainting that operates natively in three-dimensional space. Specifically, we start by embedding the 3DGS scene into the latent space of the 3D prior. Expressing the observed 3D context in a latent space, we predict the missing geometric structure of the target region, providing spatial support

We **IN**paint **G**aussian **S**cenes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

for the following latent masked region feature generation. Having predicted the target region's structure, we subsequently generate the associated appearance features by conditionally sampling from the generative prior, via a combination of a progressive context-conditioning algorithm and a novel self-attention masking mechanism. Finally, we transform the generated latent features to Gaussians via a generative decoder adapted to the observed 3DGS scene in order to ensure perceptual consistency between existing and generated content, before merging generated and existing context Gaussians.

Importantly, our approach does not require inpainted reference views from a 2D model: the missing content is generated directly in 3D space, which avoids blur and artifacts from multi-view inconsistencies of several hallucinated 2D completions. Further, generating content in 3D allows us to skip expensive post-inpainting 3DGS optimization, which notably reduces inference time, a key requirement in interactive applications. The generated Gaussians are, by design, multiview-consistent, and conditioning the generation on the observed context and adapting the decoder to scene-specific characteristics allows the generated Gaussians to seamlessly integrate with the existing scene.

The contributions of this work are the following: **(i)** we introduce **WINGS**: a novel, reference-free framework for inpainting 3DGS with a 3D-native generative prior that does not rely on any 2D-inpainted reference views. **(ii)** We propose a novel structure-completion network and an attention-masking strategy for structured latent inpainting that enables generation of context-consistent target-region features at inference time without the need to retrain the generative prior. **(iii)** We introduce a scene-conditioned decoder adaptation procedure that bridges the representation gap between the learned prior and the observed scene's arbitrary appearance distribution.

2 RELATED WORK

3D Gaussian Splatting Inpainting. The first methods for 3D radiance field inpainting, SPIn-NeRF (Mirzaei et al., 2023) and Weder et al. (2023), address object removal in NeRFs by obtaining multiview segmentation masks and independently inpainting the masked RGB input views with a learned 2D inpainter. To mitigate inconsistencies across the inpainted views, SPIn-NeRF optimizes the NeRF via a perceptual loss rather than pixel-wise supervision, and further inpaints depth-maps from the original NeRF to use as geometric priors during optimization. Adapting this idea to Gaussian splatting, a first category of 3DGS inpainting methods leverages multiple views inpainted through a 2D diffusion model followed by optimization from points initialized around the target region. GaussianEditor (Chen et al., 2024b) transforms an inpainted render to a mesh, which is then re-converted to Gaussians and re-optimized with supervision from the inpainted view. Gaussian Grouping (Ye et al., 2024) inpaints in 2D with LaMa (Suvorov et al., 2022), then lifts inpainted pixels to 3D and proceeds to 3DGS optimization supervised by the inpainted views.

More recently, 3DGS inpainting approaches have largely relied on joint appearance and depth inpainting. GScream (Wang et al., 2024) uses guidance from estimated monocular depth to initialize Gaussians from a single 2D inpainted view, while GPGS (Lee & Cho, 2026) leverages a point cloud masked autoencoder to perform geometry completion. AuraFusion360 (Wu et al., 2025a) uses depth diffusion to initialize Gaussians, and Inpaint360GS (Wang et al., 2026) performs depth-guided inpainting, combined with cross-frame conditioning in order to improve multi-view consistency. 3DGIC (Huang et al., 2025) uses a similar depth-guided approach to identify background pixels. Similarly, InFusion (Liu et al., 2024) combines inpainted reference views with a depth completion model, and Zhou et al. (2025) proposes an uncertainty-guided approach to Gaussian inpainting.

Recent approaches increasingly investigate inpainting in 3D space directly, with or without inpainted reference views. InstaInpaint (You et al., 2025) leverages a large reconstruction model (Zhang et al., 2025) to inpaint a 3DGS scene conditioned on a unique inpainted reference view. 3D-GIMP (Tian et al., 2026) similarly inpaints a single keyframe in 2D and then runs a 3D-aware patch-match algorithm, while GS-RoadPatching (Chen et al., 2025) searches for correspondences in 3D with applications to completion of road scans.

Despite increasingly incorporating 3D-aware geometric priors, many of the aforementioned approaches still derive the missing region's appearance from one or more 2D inpainted reference views and subsequently reconcile or refine this information in the 3D representation via depth-cues, cross-view-attention or other auxiliaries. Our method, in contrast, operates directly in 3D space.

Generative priors. Recently, generative models have emerged as remarkable means of creating 3D assets with text or image conditioning (Xiang et al., 2025b;a; Zhao et al., 2026; Yan et al., 2026). While largely operating inside learned latent spaces, some generative 3D models interface with 3D Gaussians, through variational autoencoders (Xiang et al., 2025b) or direct generation (Yan et al., 2026).

VoxHammer (Li et al., 2026) performs inpainting and editing on 3D structured latents by sampling from a general 3D generative model (Xiang et al., 2025b), relying on a single inpainted image injected as conditioning in the prior. Albeit related to our method, VoxHammer requires an inpainted reference view, is focused on mesh- and structured-latent-editing, and restricts 3DGS applications to asset editing. Similarly, InpaintSLat (Chung et al., 2026) proposes a structured latent inpainting framework based on the TRELLIS model (Xiang et al., 2025b) to edit 3D assets. Importantly, both frameworks focus on mesh editing and object-level completion, rather than full-scene inpainting.

Overall, existing 3DGS inpainting methods overwhelmingly rely on 2D diffusion priors, while 3D-native editing methods largely operate within the learned distribution of their native representations, whose appearance domain often differs substantially from that of photorealistic, reconstructed scenes. Bridging the two, we show how a powerful pre-trained 3D generative model can be transferred to arbitrary Gaussian splatting scenes by conditionally generating a latent representation and adapting the generative decoder to the scene-specific, photorealistic Gaussian distribution.

Training-Free Conditioning of Generative Priors. A complementary line of work repurposes pre-trained generative models for conditional generation without task-specific re-training. RePaint (Lugmayr et al., 2022) performs image inpainting with an unconditional diffusion model by repeatedly injecting observations of the known region throughout the reverse process, and harmonizes generated with observed content via re-sampling. More generally, diffusion posterior sampling (DPS, Chung et al. (2023) incorporates measurement constraints into generation through likelihood-guided updates, while ReSample (Song et al., 2023) enforces hard data consistency when solving inverse problems with latent diffusion models. Orthogonally, inference-time attention control methods such as MasaCtrl (Cao et al., 2023) manipulate self-attention to control information flow during generation without modifying the underlying model. Sartor et al. (2026) further show that joint attention with attention dropout can be leveraged to control interactions between input modalities.

In contrast to the previous methods, our approach builds on these principles in a structured *three-dimensional* latent space, conditioning a frozen 3D generative prior on observed scene latents while restricting information flow between observed and inpainted regions.

3 METHOD

We consider a 3DGS scene G and a user-prescribed masked region $\mathcal{M}$. After removing the Gaussians lying in $\mathcal{M}$, usually corresponding to an unwanted object, our goal consists in generating Gaussian primitives G_m inside the masked region while remaining consistent with the observed remaining context G_c. Adopting the probabilistic view of Kheradmand et al. (2024), we thus aim at sampling masked region Gaussians from the conditional distribution $G_m \sim \mathcal{P}(G_m|G_c)$.

This conditional distribution is highly multi-modal at two levels. At a semantic level, the generation of plausible content admits infinitely many valid solutions, and at the primitive level, multiple combinations of Gaussians can result in perceptually identical results. Because of the former, relying on 2D inpainting priors (Ye et al., 2024; Liu et al., 2024) can lead to multi-view inconsistencies; while the latter presents a significant obstacle to generating content directly in the space of 3D Gaussians. Motivated by these observations, our method operates in the structured latent space of a pre-trained 3D-native generative prior, namely TRELLIS (Xiang et al., 2025b).

After recalling the general architecture of the 3D-native generative prior used here (Section 3.1), we describe our protocol for sampling $G_m \sim \mathcal{P}(G_m|G_c)$. First, we predict the sparse voxel structure of the missing region (Section 3.2), before generating latent features for these voxels through a combination of conditional sampling and a self-attention masking mechanism (Section 3.3). Finally, we adapt the latent decoder to the specific appearance of the scene (Section 3.4), and merge the decoded Gaussians with the original context. An overview of our method is displayed in Figure 1.

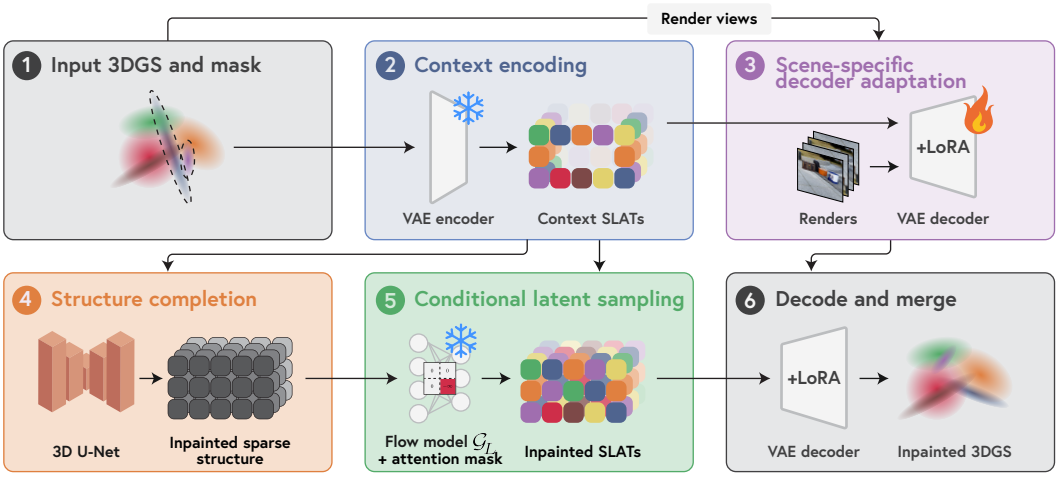


Figure 1: Overview of WINGS. Given the input 3DGS encoded as structured latent (SLAT) and a mask $\mathcal{M}$, our structure completion network predicts masked-region occupancy. The SLATs are inpainted directly in 3D, conditioned on the context latents and decoded to Gaussian splats.

3.1 PRETRAINED 3D-NATIVE PRIOR

The generative prior we use operates in a learned latent space, where 3D content is modeled by a collection of L d-dimensional latent tokens $\boldsymbol{z} \in \mathbb{R}^{d \times L}$ obtained by encoding the Gaussian splats. These latents reside on a voxel grid, and are henceforth denoted as *structured* latents (SLATs), enabling us to distinguish the latents $\boldsymbol{z}_c$ obtained from the observed context from the latents $\boldsymbol{z}_m$ that we generate within the masked region $\mathcal{M}$: we denote $\boldsymbol{z} = \boldsymbol{z}_m \cup \boldsymbol{z}_c$. Our goal is thus to sample from the conditional distribution $\mathcal{P}_{\mathcal{G}_L}(\boldsymbol{z}_m | \boldsymbol{z}_c)$, which is induced by a latent generator $\mathcal{G}_L$.

In practice, we use the two-stage TRELLIS architecture (Xiang et al., 2025b): a first model $\mathcal{G}_S$ predicts the latent occupancy structure O (a 64^3 binary voxel grid) and a second model $\mathcal{G}_L$ assigns latent features z_i to active voxels through rectified flow integration. We use TRELLIS.1 as it provides a pre-trained variational autoencoder interfacing Gaussians with SLATs. A notable challenge that we address in this paper is inpainting Gaussians from photorealistic, multi-object scenes, which conflicts with the non-photorealistic appearance of TRELLIS-generated assets stemming from the model's object-centric training procedure. We tackle this distribution mismatch through inpaint-specific structure completion (Section 3.2), conditional latent sampling (Section 3.3) and scene-conditioned decoder adaptation (Section 3.4).

3.2 ENCODING AND STRUCTURE COMPLETION

Context encoding. Although we seek to perform inpainting on potentially large scenes, the TRELLIS prior operates at a specific scale determined by the fixed resolution of its SLAT structure. We therefore extract a local context $G_{c^*} \subset G_c$ around the target region $\mathcal{M}$ by discretizing and finding the smallest bounding box such that the total ratio of voxels in G_{c^*} to voxels in the masked region is 6. We find this empirically determined ratio to be a good trade-off between enough context information and sufficiently dense resolution for the reconstructed Gaussians G_m. Our early experiments revealed the TRELLIS latent flow model $\mathcal{G}_L$ to be biased towards generating axis-aligned structures, so we automatically align the bounding box with the visual features of the context G_{c^*} using the structure tensor (Bigun et al., 1991) (see Appendix B.3 for details). Following the TRELLIS protocol, we finally encode this context into context latents $\boldsymbol{z}_c$ via the pretrained VAE encoder $\mathcal{E}$.

Structure completion. The 64^3 binary occupancy grid O is known for context but initially empty for the voxels corresponding to the masked region $\mathcal{M}$. Accurate completion of this structure is crucial, since the subsequent latent generator $\mathcal{G}_L$ can only produce content for voxels marked as active in O. Xiang et al. (2025b) use a structure flow model $\mathcal{G}_S$ for this task and investigate the application of the RePaint algorithm (Lugmayr et al., 2022) for structural completion. However, we find that $\mathcal{G}_S$'s learned objective of generating new, complex content competes with the inpainting task, which

aims at producing a visually plausible continuation of the surrounding context rather than complex new structures. While satisfactory for image-conditioned generative inpainting tasks (Xiang et al., 2025b; Li et al., 2026), $\mathcal{G}_S$ exhibits degraded generation quality in the structure completion regime in which we operate (see Figure 4 for an analysis).

We therefore opt to train a new structure completion model that directly predicts O based on the context structure O_c and the target region mask $\mathcal{M}$. We choose a 3D residual U-Net architecture (Çiçek et al., 2016; He et al., 2016) with bottleneck attention, which allows us to benefit from the translation-equivariance and multi-scale receptive field of convolutional networks, while preserving fine-grained spatial structure through skip-connections. We train our model on a curated corpus of 900 diverse voxelized scene-level 3D Gaussian splatting scenes with randomly sampled masked regions. No scene from any of our benchmark evaluation datasets was used in training the structure prediction network. In order to guarantee clean structure completion example pairs, we curate the dataset by rejecting samples where the target structure deviates substantially from a quadric extrapolation of the context structure or contains high volumetric variations. The dataset curation process is further detailed in Appendix C.1 and results in more than 71,000 training samples. We train our structure completion network for 300 epochs using a linear combination of weighted binary cross-entropy and Dice loss: $\mathcal{L}_{\text{structure}} = 0.3\,\mathcal{L}_{\text{W-BCE}} + 0.7\,\mathcal{L}_{\text{DICE}}$.

3.3 Conditional structured latent sampling

The predicted structure O_m specifies the spatial structure of target region latents z_m, but not the value of their latent features, which we must sample from the conditional distribution $\mathcal{P}_{\mathcal{G}_L}(\boldsymbol{z}_m|\boldsymbol{z}_c)$. However, TRELLIS' flow model $\mathcal{G}_L$ only allows us to sample from the *unconditional* prior $\mathcal{P}_{\mathcal{G}_L}(\boldsymbol{z})$ (notwithstanding image/text conditioning injected through cross-attention, unused here). In the following section, we show that this general prior can be used to approximate conditional sampling in the case of inpainting by combining two schemes: context conditioning and self-attention masking.

Progressive context conditioning. The conditional sampling scheme that we adopt is based on the RePaint strategy (Lugmayr et al., 2022), which repurposes an unconditional flow model by altering the sampling trajectory. Let $\boldsymbol{z}^t = \boldsymbol{z}_m^t \cup \boldsymbol{z}_c^t$ be the structured latents at flow time t. At each integration step, the flow model $\mathcal{G}_L$ predicts a rectified flow velocity $v(\boldsymbol{z}^t, t)$ and advances through the flow trajectory by generating $\boldsymbol{z}^{t+dt} = \boldsymbol{z}^t + v(\boldsymbol{z}^t, t)dt$. Given that the true context latents $\boldsymbol{z}_c^0$ are known, RePaint then replaces the context $\boldsymbol{z}_c^{t+dt}$ with $\hat{\boldsymbol{z}}_c^{t+dt}$, the known context latents noised at that time:

$$\boldsymbol{z}^{t+dt} = \hat{\boldsymbol{z}}_c^{t+dt} \cup \boldsymbol{z}_m^{t+dt} \quad \text{where} \quad \hat{\boldsymbol{z}}_c^t = (1-t)\boldsymbol{z}_c^0 + t\epsilon, \quad \epsilon \sim N(0,1). \tag{1}$$

In order to harmonize the joint generation of context and masked latents, this consistency replacement is supplemented by stochastic renoising jumps that provide additional opportunities for the masked region to be denoised towards context-consistent latents. While this conditional sampling strategy was originally suggested by TRELLIS (Xiang et al., 2025b), its efficacy is highly tied to the use of their original structure prediction model $\mathcal{G}_S$, as its generated sparse structure is generally in-distribution for $\mathcal{G}_L$. While switching to our U-Net predictor improves structure completion, it also creates a potential mismatch between the context and masked region distributions. Using a high number of flow integration timesteps for SLAT generation alleviates this issue, but leads to a significant slowdown. We instead introduce a complementary conditioning mechanism that alters the flow transformer's self-attention.

Self-attention masking. The TRELLIS latent flow model $\mathcal{G}_L$ uses transformer blocks that compute multi-head self-attention over structured latents. Applying the RePaint strategy here initially leaves the target region latents $\boldsymbol{z}_m$ primarily determined by the flow model's unconditional trajectory. Since they continuously self-attend throughout the transformer and due to potential sparse structure distributional mismatch, they risk to form a self-consistent region that is too robust for the context to impose its condition.We therefore propose to alter the information flow during integration in a way that explicitly promotes information sharing between context and target latents, while simultaneously hindering their self-attention. Inspired by the mutual self-attention mechanism from Cao et al. (2023), we use an additive attention mask M to prevent the target region's queries from attending to any of the target region's keys:

$$A_{\text{masked}} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right), \quad M_{ij} = \begin{cases} -\infty & \text{if } z_i \text{ and } z_j \text{ are in } \boldsymbol{z}_m \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

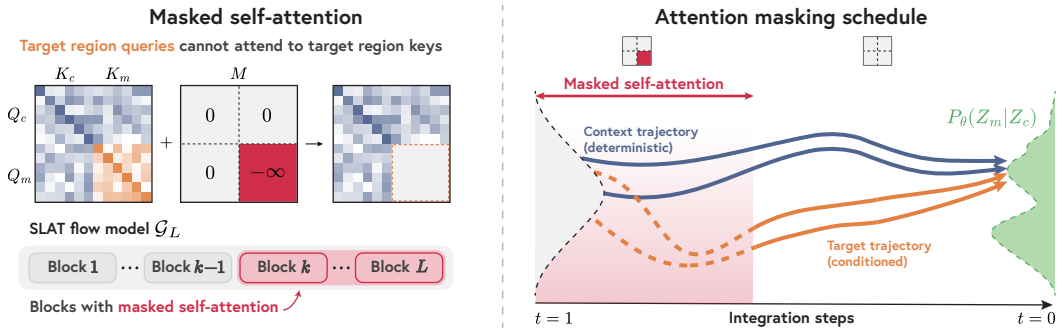


Figure 2: Attention masking. The attention mask acts on the query-key dot product $QK^\top$, effectively zeroing out target-to-target self-attention interactions. We note that, since the mask is not applied to the value tensor V, the output of self-attention is not null for target latents.

This is complementary to RePaint, as the latter enforces conditioning through backwards jumps along the sampling trajectory, stochastic re-noising and context latent replacement at each integration time step, while our attention masking mechanism directly alters information flow during the evaluation of the velocity $v(\boldsymbol{z}^t, t)$. In order to preserve self-consistency within the masked region, we apply this masked self-attention only for later transformer blocks (see Appendix B.1) and only for integration steps between $t = 1.0$ and $t = 0.8$ (see Figure 2). In summary, combining RePaint-like jumps with our masked attention strategy allows us to conditionally sample the unconditional flow model $\mathcal{G}_L$ and generate latents that are conditioned on, and therefore consistent with, the context region.

3.4 Scene-conditioned decoding

After having obtained structured target latents z_m via the aforementioned conditional latent sampling stage (Section 3.3), we decode $\boldsymbol{z} = \boldsymbol{z}_m \cup \boldsymbol{z}_c$ into Gaussians using the pre-trained VAE decoder $\mathcal{D}_\theta$ with parameters θ. However, since $\mathcal{D}_\theta$ was trained on the limited, object-centric dataset of TRELLIS, its induced pushforward measure $(\mathcal{D}_\theta)_\#(\mathcal{P}_{\mathcal{G}_L}(\boldsymbol{z}_m|\boldsymbol{z}_c))$ is not guaranteed to match our target $\mathcal{P}(G_m|G_c)$. As such, even semantically correct latents can be decoded into perceptually dissimilar Gaussians, resulting in visually unpleasant boundary discontinuities between context G_c and generated Gaussians $\mathcal{D}_\theta(\boldsymbol{z}_m)$.

We address this distribution mismatch by adapting the decoder to the observed scene's appearance. Alongside the latent generation process, we augment $\mathcal{D}_\theta$ with a rank-16 LoRA adapter (Hu et al., 2021) for $\mathcal{D}_\theta$. We unfreeze the final layer of $\mathcal{D}_\theta$, responsible for generating the Gaussian opacity, scale, and rotation values. The 2.70 M parameters ϕ of this lightweight adapter are optimized through minimization of a photometric loss on rendered views:

$$\phi^* = \arg\min_\phi \mathop{\mathbb{E}}_{v\sim\mathcal{V}}\Big[\mathcal{L}_{\text{data}}\big(\mathcal{R}(\mathcal{D}_{\theta,\phi}(\boldsymbol{z}_c), v); \mathcal{R}(G_{c^*}, v)\big)\Big] \tag{3}$$

where $\mathcal{R}$ is a differentiable rasterizer and $\mathcal{V}$ is a collection of 36 anchor viewpoints. Importantly, these renders only provide supervision for adapting the view-independent decoder to the scene; they are never 2D-inpainted or serve as inputs or conditioning for the completion process. The same adapted decoder and latent representation are shared across all viewpoints. We also do not require specific reference views as we sample synthetic viewpoints around the cropped context G_{c^*}.

4 Results

We evaluate our method across three 360° radiance-field inpainting benchmarks using both quantitative image-quality metrics and a user study, as well as qualitative analysis.

Baselines and evaluation protocol. We compare against Gaussian Grouping (Ye et al., 2024), InFusion (Liu et al., 2024), and Inpaint360GS (Wang et al., 2026), the latter of which reports strong performance on its proposed Inpaint360 benchmark. For all methods, we use the same input 3DGS scene and pre-defined 3D mask. This differs from evaluation protocols in which scene reconstruction and mask segmentation are performed independently for each method (Wu et al., 2025a; Wang et al.,

Table 2: Averaged metrics across datasets. Bold and underline indicate best- and second-best.

Method	LPIPS ↓	FID ↓	Local-FID ↓	KID ($\times 10^3$) ↓	Human Pref. (%) ↑	Runtime (s) ↓
Gauss. Grouping	0.255	66.98	66.53	18.02	34.8	780.0
InFusion	0.250	88.71	87.74	40.31	18.6	212.9
Inpaint360GS	<u>0.236</u>	<u>51.17</u>	<u>52.43</u>	<u>6.16</u>	<u>63.6</u>	<u>202.1</u>
Ours	**0.235**	**49.96**	**49.21**	**6.08**	**82.9**	**176.8**

2026), as we intend to isolate differences due to the inpainting method itself. As such, our results also apply to 3DGS scenes that do not result from an optimization-based reconstruction. All experiments run on a single NVIDIA H100 GPU; see Section B.1 for implementation details.

Datasets. We evaluate on three unbounded 360° radiance-field inpainting datasets: (1) COR-NeRF (Lee et al., 2024), containing 11 diverse indoor and outdoor scenes with objects at different scales; (2) Inpaint360 (Wang et al., 2026), containing 11 single- and multi-object scenes; and (3) 360-USID (Wu et al., 2025a), containing seven scenes. In total, our evaluation comprises 29 scenes.

4.1 Quantitative and qualitative results

Automated metrics. We evaluate WINGS on a comprehensive collection of quantitative metrics. We employ distributional perceptual metrics as our main quantitative evaluators: for each method, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018). As FID resizes images to 299×299 to accommodate the Inceptionv3 backbone's input size, it is poorly suited for evaluating fine detail in the inpainted region (Xie et al., 2023; Pavasovic et al., 2026). To address this, we additionally report Local-FID, obtained by cropping a 299×299 region centered on the target region, thus allowing FID to operate at full resolution. We also report LPIPS. As detailed in Appendix A, we set aside PSNR and SSIM since such reference-based metrics do not account for the large number of valid inpainting solutions in the target region.

User study. As detailed in Appendix A.1, automated metrics poorly correlate with human preference in our inpainting setting: we thus further assess the visual quality of our approach via a blind user study. Across the 29 scenes from the COR-NeRF, Inpaint360 and 360-USID datasets, we randomly sample a viewpoint from a held-out set, and render an input view (with the object present) and the inpainted results of two methods, displayed in random left-right order. Users are asked which result looks better, with the options of skipping (e.g., if the region is occluded from this view) or signaling a tie, in case both results look equally good or bad. During the study, 29 participants rated a total of 1401 pairs, an average of 48.3 per user. Table 1 shows that our method is overwhelmingly preferred by human evaluators, achieving a 75% preference rate against the strongest baseline and attaining an expected rank of 1.48.

Table 1: User study. Each cell contains the tie-adjusted win-rate (higher is better) in %, row-vs-column. See App. A.2 for details.

Method	Ours	Inpaint360GS	Gaussian Grouping	InFusion	Expected rank ↓
Ours	–	75.11	84.19	93.01	**1.48 ± 0.07**
Inpaint360GS	24.89	–	77.04	85.04	2.13 ± 0.07
Gaussian Grouping	15.81	22.96	–	67.17	2.94 ± 0.08
InFusion	6.99	14.96	32.83	–	3.45 ± 0.08

Qualitative results. Qualitative inspection of our results confirms the quantitative analysis: our method consistently achieves highly plausible and sharp inpainting results that respect well the visual structure of the scene and integrate seamlessly with their surroundings. Other methods often produce blurry and fuzzy regions or show significant color shifts or artifacts. We show a subset of our results in Figure 3; for a full gallery and a comparison against baselines on all scenes, please see Appendix E and the electronic supplemental.

4.2 Ablation study

Attention masking. This ablation uses the same encoded context SLATs $\boldsymbol{z}_c$ and predicted structure O but removes the attention masking mechanism (see Figure 2). The structured latent conditional sampling stage hence decays back to the original conditional sampling of $\mathcal{G}_L$ used by RePaint, as implemented by Xiang et al. (2025b). The number of integration steps, backwards jump length and number of jumps is kept identical for both.

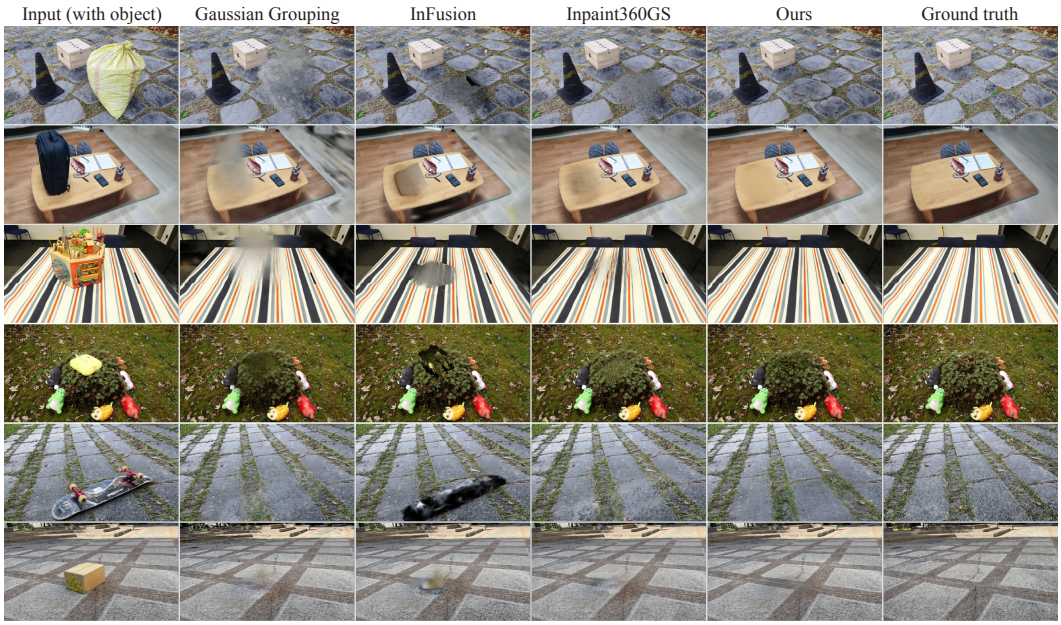


Figure 3: Qualitative comparison on the COR-NeRF, Inpaint360 & 360-USID datasets (top to bottom, two rows each, best viewed zoomed in). See the supplemental video to assess multiview-consistency.

Structure completion. We ablate our structure completion U-Net and instead use RePaint sampling of the sparse structure model $\mathcal{G}_S$ in a way identical to Xiang et al. (2025b), with 50 integration time steps. After obtaining the completed structure O, we perform attention-masked SLAT generation with the usual hyperparameters.

Decoder adaptation. For this ablation, we predict structure using our U-Net and run our full latent pipeline (with progressive context conditioning and self-attention masking), and simply refrain from applying the scene-conditioned parameters ϕ^* to $\mathcal{D}_\theta$ during the final decoding of generated latents.

Table 3: Perceptual metrics for our method and its ablations.

Method	LPIPS ↓	FID ↓	Local-FID ↓	KID ($\times 10^3$) ↓
Ours	**0.235**	**49.961**	**49.214**	**6.080**
Ours (w/o attention masking)	<u>0.236</u>	<u>52.806</u>	<u>51.718</u>	<u>9.497</u>
Ours (w/o axis alignment)	0.241	55.336	54.661	10.030
Ours (w/o structure prediction model)	0.238	57.260	59.181	11.893
Ours (w/o decoder adaptation)	0.250	68.776	63.755	24.867

Table 3 reports perceptual metrics for all ablations. Decoder adaptation has the largest impact on perceptual quality, with its removal substantially degrading all four metrics, while the structure completion model provides the second-largest contribution, particularly for Local-FID. Attention masking yields a smaller albeit consistent improvement, but contributes significantly to seamless blending and overall visual quality, as the qualitative visualizations in Figure 5 show.

We further ablate our structure completion model against the baseline proposed by TRELLIS (Xiang et al., 2025b), applying the RePaint algorithm (Lugmayr et al., 2022) to the sparse structure flow model $\mathcal{G}_S$. We evaluate both approaches on a held-out dataset of 3DGS scenes unused during training of our structure completion U-Net, and report their relative performance in Figure 4. We find the sparse structure completion of the TRELLIS baseline to strongly degrade as the masked region size increases compared to context, while our structure completion model offers both accurate and consistent predictions even at high masked voxel ratios.

5 Discussion & Conclusion

Limitations. While our method successfully inpaints 3DGS assets in unbounded 360° scenarios and outperforms other current approaches in both perceptual metrics and human preference, it is not free of limitations, which we highlight here as avenues for future work. First, we observe that content

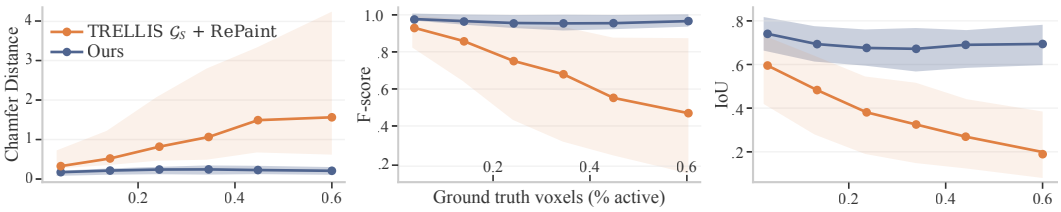


Figure 4: Sparse structure reconstruction metrics against masked-to-context voxel ratio. Our proposed structure completion network strongly outperforms TRELLIS $\mathcal{G}_S$ + RePaint for all context ratios.

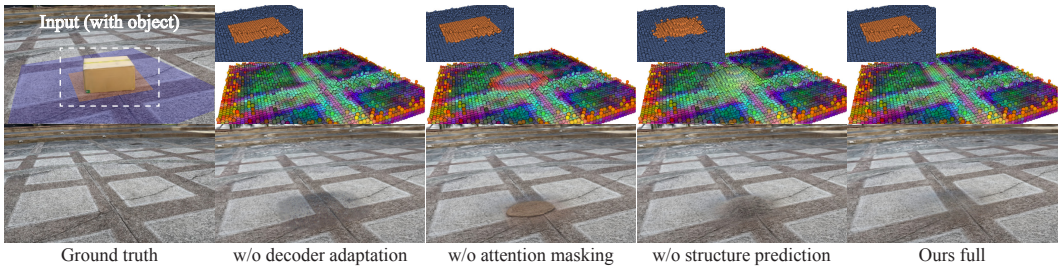


Figure 5: Ablation visualization. We show the context and mask regions as overlays on the input image. Each column shows the completed structure (inset, top row), the PCA of the inpainted features after the integration process (principal three components in RGB, middle row), and the final decoded result in the bottom row.

generation of complex and intricate patterns in latent space is biased towards axis-aligned patterns, and struggles with misaligned motifs. We devise an axis-alignment heuristic to overcome this, but future research could help mitigate this at a structural level, e.g., inspired by SO(3)-equivariant structure alignment mechanisms (Ding & Hofmann, 2025) or by leveraging recent approaches to equivariant 3D-native priors (Ji et al., 2026). Moreover, due to architectural limitations of the TRELLIS VAE decoder $\mathcal{D}_\theta$, our approach does not generate the higher-order spherical harmonic components of the inpainted splats. While we plan to investigate this in future work, our core framework is agnostic to the specific choice of backbone, and thus could accommodate future priors supporting higher-order spherical harmonics.

Additionally, despite operating with a lower runtime than existing 2D-based approaches, our method requires around three minutes for inpainting. As Table 5 shows, a large fraction of that time is allocated to the decoder adaptation stage. Rather than training a single adapter per scene, future work will focus on training a single multi-scene adapter, which will amortize the training cost over successive interactive edits of the same scene.

Finally, pixel-reference metrics such as PSNR and SSIM are ill-suited for evaluating the output of a stochastic process such as diffusion-based inpainting, since perceptually plausible variations (e.g., a different realization of the same wood grain) may incur a large pixel-wise error despite being visually convincing (see our discussion in Section 4 and Appendix A). Extending perceptually motivated, inpainting-specific quality metrics (Chen et al., 2024a) to 3DGS inpainting is beyond the scope of this work, but remains an important direction for future research.

Conclusion. We propose a novel, 3D-native framework for inpainting regions of real-world, unbounded Gaussian splatting scenes. Our approach leverages a powerful generative prior to conditionally create plausible content in latent space. We introduce three separate mechanisms improving the quality and faithfulness of inpainting: a learned structure completion network, a self-attention masking strategy to improve information exchange between context and generated latents, and a scene-specific decoder adaptation module. Our method completes in a few minutes, outperforms existing 3DGS inpainting methods, achieves the best aggregate metrics on a variety of scenes from three different datasets, and is consistently ranked highest by human preference. To the best of our knowledge, our method is the first 3DGS inpainting approach to perform content generation in the representation space of a 3D-native prior, opening important future perspectives for interactive 3D scene interaction and editing.

AI USE STATEMENT

Our use of generative AI tools during the conduction of this research work mainly falls in two categories. First, we used AI coding assistants during the writing of the code base that supports our experiments. This includes co-writing lines of code and suggesting some experimental parameters and default configuration values. Second, we used AI critics as a means to get early feedback about our scientific methodology: we asked about the relevance of our experimental protocol and its adequacy to our claims, about which part of our bibliography was most related to our work about the consistency, self-completeness and readability of our drafts. We have *not* used generative AI tools for directly redacting the present paper nor for its appendix, at the exception of the HTML/CSS theme of the electronic supplemental (table content was procedurally generated from our evaluation data). We also did *not* use AI for any of the figures presented in this paper, nor for its overall structure or the reference section. AI was *not* used to pick the research topic, formulate mathematical claims, alter datasets or interpret results. The qualitative analysis of our results and the user study were sourced on humans, without AI judgment. *We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.*

REPRODUCIBILITY STATEMENT

Our method builds on the well-known and publicly available TRELLIS architecture, and our evaluation uses established, publicly available datasets. We include extensive implementation details, including hyper-parameters and configuration values, in the appendix and plan to release our code and model weights upon acceptance to enable the community to reproduce our results and build upon our work. The electronic supplemental shows 10 randomly sampled viewpoints for each scene and baseline from which we conducted the user study. We will release the full set (15 GB) and protocol upon publication so that anyone can reproduce the study.

REFERENCES

J. Bigun, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991. ISSN 1939-3539. doi: 10.1109/34.85668.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, February 2018.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 0006-3444. doi: 10.2307/2334029.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513, October 2023. doi: 10.1109/ICCV51070.2023.02062.

F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, February 2005. ISSN 0167-8396. doi: 10.1016/j.cagd.2004.09.004.

Guo Chen, Jiarun Liu, Sicong Du, Chenming Wu, Deqi Li, Shi-Sheng Huang, Guofeng Zhang, and Sheng Yang. GS-RoadPatching: Inpainting Gaussians via 3D Searching and Placing for Driving Scenes. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, December 2025. doi: 10.1145/3757377.3763892.

Tianyi Chen, Jianfu Zhang, Yan Hong, Yiyi Zhang, and Liqing Zhang. Assessing Image Inpainting via Re-Inpainting Self-Consistency Evaluation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 291–300, Boise ID USA, October 2024a. ACM. ISBN 979-8-4007-0436-9. doi: 10.1145/3627673.3679693.

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and Controllable 3D Editing with

Gaussian Splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21476–21485, June 2024b. doi: 10.1109/CVPR52733.2024.02029.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Jaeyoung Chung, Suyoung Lee, and Kyoung Mu Lee. InpaintSLat: Inpainting Structured 3D Latents via Initial Noise Optimization, May 2026.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pp. 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8. doi: 10.1007/978-3-319-46723-8_49.

Roger R. Davidson and Robert J. Beaver. On Extending the Bradley-Terry Model to Incorporate Within-Pair Order Effects. *Biometrics*, 33(4):693–702, 1977. ISSN 0006-341X. doi: 10.2307/2529467.

Yuhui Ding and Thomas Hofmann. Scalable Non-Equivariant 3D Molecule Generation via Rotational Alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 13814–13824. PMLR, October 2025.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, June 1981. ISSN 0001-0782, 1557-7317. doi: 10.1145/358669.358692.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90.

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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6629–6640, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *ACM SIGGRAPH Computer Graphics*, 26(2):71–78, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134011.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, October 2021.

Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3D Gaussian Inpainting with Depth-Guided Cross-View Consistency. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26704–26713, June 2025. doi: 10.1109/CVPR52734.2025.02487.

Guangda Ji, Qimin Chen, Qinchan Li, Mingrui Zhao, Kai Wang, and Hao Zhang. SymTRELLIS: Symmetry-Enforced Voxel Latents for 3D Generation, September 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, August 2023. ISSN 0730-0301, 1557-7368. doi: 10.1145/3592433.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian splatting as Markov Chain Monte Carlo. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pp. 80965–80986, Red Hook, NY, USA, December 2024. Curran Associates Inc. ISBN 979-8-3313-1438-5.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. arXiv, 2015. doi: 10.48550/arXiv.1412.6980.

Yongjoon Lee and Donghyeon Cho. GPGS: Consistent 3D Object Removal via Geometry-Aware 3D Inpainting and Projected Image Refinement in 3D Gaussian Splatting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):5927–5935, March 2026. ISSN 2374-3468. doi: 10.1609/aaai.v40i8.37515.

Yongjoon Lee, Jaehak Ryu, Donggeun Yoon, and Donghyeon Cho. Consistent Object Removal from Masked Neural Radiance Fields by Estimating Never-Seen Regions in All-Views. In *Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part XXII*, pp. 416–431, Berlin, Heidelberg, December 2024. Springer-Verlag. ISBN 978-3-031-78311-1. doi: 10.1007/978-3-031-78312-8_28.

Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. VoxHammer: Training-Free Precise and Coherent 3D Editing in Native 3D Space. In *2026 International Conference on 3D Vision (3DV)*, pp. 1281–1292. IEEE Computer Society, March 2026. ISBN 979-8-3315-7312-6. doi: 10.1109/3DV69130.2026.00126.

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, April 2024.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, September 2018.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461, June 2022. doi: 10.1109/CVPR52688.2022.01117.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20669–20679, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.01980.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, July 2023. ISSN 2835-8856.

M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002. VIS 2002.*, pp. 163–170, Boston, MA, USA, 2002. IEEE. ISBN 978-0-7803-7498-0. doi: 10.1109/VISUAL.2002.1183771.

Krunoslav Lehman Pavasovic, Théophane Vallaeys, Stéphane Mallat, Giulio Biroli, Luke Zettlemoyer, Brian Karrer, and Jakob Verbeek. WaiT for the Signal: Simple Frequency-Aware Flow-Matching, July 2026.

Sam Sartor, Iliyan Georgiev, Michael Fischer, Valentin Deschaintre, and Pieter Peers. Overpainting: Localized Context-aware Diffusion Image Editing, September 2026.

Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency. In *The Twelfth International Conference on Learning Representations*, October 2023.

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 *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3172–3182, January 2022. doi: 10.1109/WACV51458.2022.00323.

Xuening Tian, Dieter Schmalstieg, and Shohei Mori. 3D-GIMP: When 3D Gaussian Inpainting Meets PatchMatch, July 2026.

Shaoxiang Wang, Shihong Zhang, Christen Millerdurai, Rüdiger Westermann, Didier Stricker, and Alain Pagani. Inpaint360GS: Efficient Object-Aware 3D Inpainting via Gaussian Splatting for 360° Scenes. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 117–127. IEEE, 2026.

Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3D Geometry and Feature Consistent Gaussian Splatting for Object Removal. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part III*, pp. 1–17, Berlin, Heidelberg, October 2024. Springer-Verlag. ISBN 978-3-031-72645-3. doi: 10.1007/978-3-031-72646-0_1.

Silvan Weder, Guillermo Garcia-Hernando, Áron Monszpart, Marc Pollefeys, Gabriel Brostow, Michael Firman, and Sara Vicente. Removing Objects From Neural Radiance Fields. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16528–16538, June 2023. doi: 10.1109/CVPR52729.2023.01586.

Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huan, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. AuraFusion360: Augmented Unseen Region Alignment for Reference-based 360° Unbounded Scene Inpainting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16366–16376, June 2025a. doi: 10.1109/CVPR52734.2025.01526.

Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. Amodal3R: Amodal 3D Reconstruction from Occluded 2D Images. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9181–9193, October 2025b. doi: 10.1109/ICCV51701.2025.00858.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *Tech report*, 2025a.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D Latents for Scalable and Versatile 3D Generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480, June 2025b. doi: 10.1109/CVPR52734.2025.02000.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. SmartBrush: Text and Shape Guided Object Inpainting with Diffusion Model. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22428–22437, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.02148.

Runjie Yan, Yan-Pei Cao, Peng Wang, Ding Liang, and Yuan-Chen Guo. Generative 3D Gaussians with Learned Density Control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, Los Angeles CA USA, July 2026. ACM. ISBN 979-8-4007-2554-8. doi: 10.1145/3799902.3811130.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXIX*, pp. 162–179, Berlin, Heidelberg, November 2024. Springer-Verlag. ISBN 978-3-031-73396-3. doi: 10.1007/978-3-031-73397-0_10.

Junqi You, Chieh Hubert Lin, Weijie Lyu, Zhengbo Zhang, and Ming-Hsuan Yang. InstaInpaint: Instant 3D-Scene Inpainting with Masked Large Reconstruction Model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, volume 15080, pp. 1–19. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-72669-9 978-3-031-72670-5. doi: 10.1007/978-3-031-72670-5_1.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo, Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu, Changrong Hu, Shaoxiong Yang, Song Zhang, Yang Liu, Tianyu Huang, Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He, Tian Liu, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation, May 2026.

Jun Zhou, Dinghao Li, Nannan Li, and Mingjie Wang. High-fidelity 3D Gaussian inpainting: Preserving multi-view consistency and photorealistic details. *Computers & Graphics*, 131:104362, October 2025. ISSN 00978493. doi: 10.1016/j.cag.2025.104362.

Appendix

We propose here an appendix to the present work. In addition, supplementary material is provided as a compressed zip file. We encourage the reader to view the two short videos demonstrating our framework, as well as the electronic gallery in `electronic_supplemental/index.html`.

A Additional results

A.1 Reference-based metrics

In addition to the results from Table 2, we report the reference-based distortion metrics PSNR and SSIM and the reference-based perceptual metric LPIPS in Table 4. Although very widely used in 3DGS inpainting benchmarks (Wu et al., 2025a; Lee & Cho, 2026; Chen et al., 2025), we argue that PSNR and SSIM are not satisfactory evaluators of inpainting quality. Indeed, these metrics report fidelity to a reference, which is in the present case a different 3DGS scene with the object

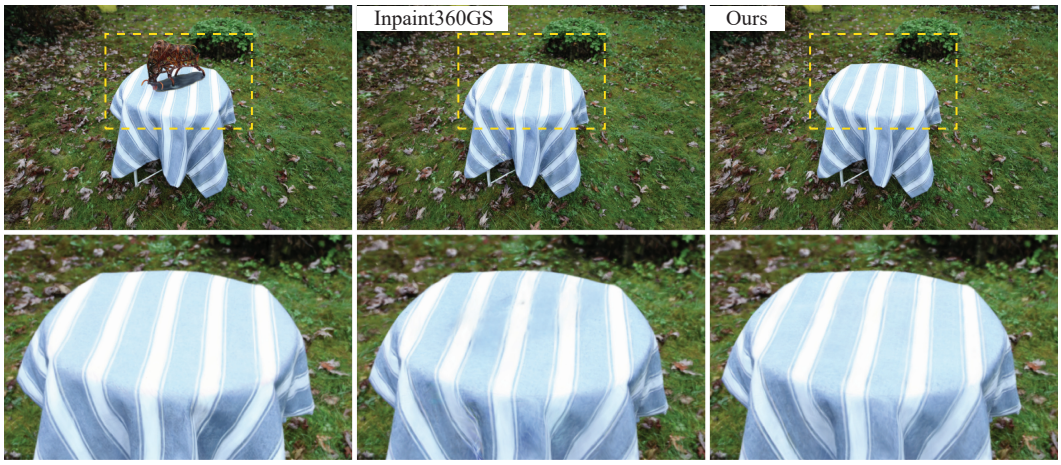


Figure 6: Inpainting results on the "Red bull" scene from the Inpaint360 dataset. Even though Inpaint360GS achieves better (lower) FID and KID scores, our method produces a more coherent continuation of the striped surface, and is preferred by 91.7% of human evaluators.

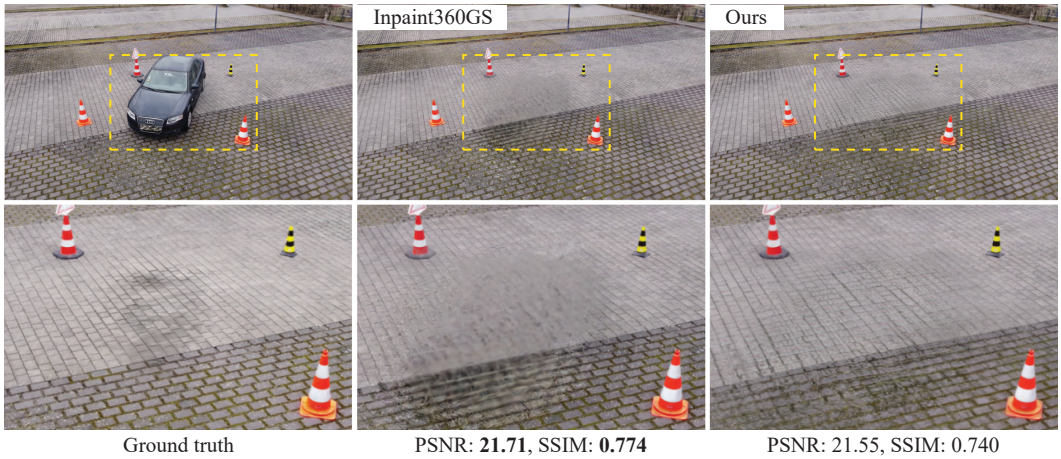


Figure 7: Inpainting results on the "Car" scene from the Inpaint360 dataset. Similar to Figure 6, Inpaint360GS performs better on the pixel-aligned metrics PSNR and SSIM, even though our method produces a more plausible and notably sharper inpainting result, making it the preferred choice for 100% of human evaluators.

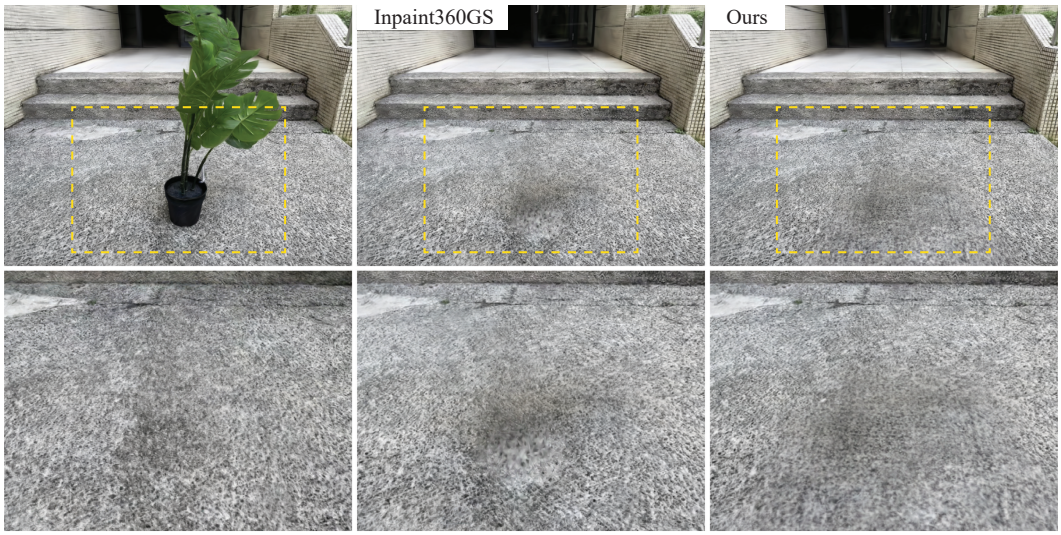


Figure 8: Inpainting results on the "Plant" scene from the 360-USID dataset. Here, Ours performs better on the distributional FID and KID metrics, despite being preferred by only 37.5% of human evaluators compared to Inpaint360GS.

removed. This is the source of many possible inconsistencies stemming from light or environment change. Furthermore, these metrics only evaluate the fidelity of inpainting with respect to one ground truth, not accounting for the large number of valid inpainting solutions admitted in the target region. Despite outperforming existing methods on PSNR and being strongly competitive on SSIM, we prefer to rely on distributional perceptual metrics such as the Fréchet Inception Distance (FID, Heusel et al. (2017)) or the Kernel Inception Distance (KID, Bińkowski et al. (2018)).

Table 4: Aggregate image-fidelity metrics, averaged across all datasets. Bold and underline indicate the best and second-best result in each column, respectively.

Method	**PSNR** ↑	**SSIM** ↑	**LPIPS** ↓
Gaussian Grouping	20.58	0.617	0.255
InFusion	20.68	0.628	0.250
Inpaint360GS	<u>21.94</u>	**0.644**	<u>0.236</u>
Ours	**21.95**	<u>0.640</u>	**0.235**

We observe that FID and its full-resolution variant Local-FID are the second-best correlated metrics with human preference, behind the reference-based LPIPS metric (Appendix A.1), while KID only weakly correlates with human judgment. Table 6 proposes a comparative alignment analysis of human preference and quantitative metrics, computed only for Ours-Inpaint360GS duels. The table cells illustrate the relative metric superiority of Ours against Inpaint360GS, colored in green when our method performs better and red when it performs worse. The color shade is proportional to the log-ratio between the two metrics, except for PSNR and the two KID metrics, where we use the difference, as PSNR is already a logarithmic metric and KID can take negative values. Remarkably, even the best-correlated metrics (LPIPS, FID and Local-FID) exhibit a timid Spearman ρ correlation coefficient with human preference of respectively 0.48, 0.40 and 0.40, indicating only partial agreement with human preference. In contrast, reference-based distortion metrics such as PSNR and SSIM are expectedly poor predictors of human preference, with Spearman ρ coefficients of 0.14 and 0.21 only. We illustrate these phenomena with qualitative visualizations: Figures 6 and 7 show cases where Inpaint360GS surpasses WINGS on metrics despite our method being preferred by humans, and Figure 8 shows a case where humans prefer Inpaint360GS but our result obtains better metrics.

In addition to justifying our choice to omit PSNR and SSIM from our quantitative evaluation protocol, these results highlight the lack of 3D-specific metrics able to assess inpainting quality in a way that aligns with human judgment.

A.2 USER STUDY

The user study was conducted through an interactive web application, displaying a render of the input scene, containing an object to be removed, and renders of two randomly chosen methods between Gaussian Grouping, InFusion, Inpaint360GS and WINGS. The opposing methods are sampled uniformly: in the final study, all method pairs received an average of 233.5 head-to-head comparisons. In particular, Ours and Inpaint360GS were involved in 231 head-to-head comparisons, distributed among 29 participants. In order to avoid bias due to position, the left-right ordering of the methods is chosen at random. Among the 1401 ratings, 111 exact (scene, view, method pair) tuples were shown to at least two different users. In these situations, the human participants agreed on identical ratings in 77.1% of observations, with a bootstrap 95% confidence interval of $[69.7\%, 84.6\%]$. The Krippendorff's α inter-rater reliability computed over these identical tuples is 0.680, with a per-duplicate bootstrap 95% confidence interval of $[0.579, 0.776]$. Under the Davidson model, 26 out of 29 participants obtain the same complete ordering of the four methods.

The human-preference score reported in Table 2 is computed by fitting a Bradley-Terry-Davidson model (Bradley & Terry, 1952; Davidson & Beaver, 1977) to the pairwise preferences registered during the user study. We report the expected tie-adjusted score against an average-skill opponent under this model:

$$\text{preference}_i = \mathbb{P}(i \succ \text{avg}) + \frac{1}{2}\mathbb{P}(i \sim \text{avg}).$$

Under the same model, the expected rank reported in Table 1 is computed as follows:

$$\text{Expected rank} = N_{\text{methods}} - \sum_{j \neq i} \left(\mathbb{P}(i \succ j) + \frac{1}{2}\mathbb{P}(i \sim j) \right).$$

Finally, the human preference score of Ours against Inpaint360GS reported in Table 6 is simply computed using a tie-adjusted preference with two players:

$$\text{preference}_i = \frac{W_{ij} + \frac{1}{2}T_{ij}}{N_{ij}},$$

A.3 RUNTIME ANALYSIS

We describe here in detail the runtime of our method and make its per-stage complexity explicit. Our framework comprises five main stages: **(a)** context encoding; **(b)** scene-specific decoder adaptation; **(c)** sparse structure completion; **(d)** conditional latent sampling and **(e)** final decoding and merging. Among these, stages (b) and (d) make up the overwhelming majority of the total inpainting time: the exact average runtimes of each stages are detailed in Table 5. Even though our method, performing inpainting in under 3 minutes, is suitable for interactive usage, we also benchmark a stripped-down lightweight version (Ours-XS), where we fine-tune the decoder for 100 steps only with a larger learning rate of 5×10^{-4} and a rank-8 adapter, and generate latent features for the inpainted region with fewer integration steps and a reduced RePaint schedule (50 steps, with 3 resamples and a jump size of 5). This reduced computational complexity allows us to perform inpainting in around 75 seconds with only a limited perceptual fidelity drop: on the three benchmark datasets, we achieve an FID of 57.14 and a LPIPS of 0.2366, comparable with the existing 2D-based approach of Gaussian Grouping (Ye et al., 2024) at a fraction of the runtime. This proves the ability of our lightweight framework to operate reliably at a reduced runtime, an important step towards real-time radiance field editing applications.

Table 5: Per-stage runtime of our method.

Stage	Ours	Ours (XS)
Context encoding	20.40	20.13
Decoder adaptation	103.02	23.58
Structure prediction	0.56	0.56
Conditional latent sampling	51.20	29.59
Decoding and merging	1.66	1.67
Total runtime (s) $\downarrow$	176.84	**75.39**
FID $\downarrow$	**49.96**	57.14

B IMPLEMENTATION DETAILS

B.1 IMPLEMENTATION DETAILS OF OUR METHOD

Generative prior. We use the publicly released TRELLIS model as the generative prior for all experiments in this work. In particular, we use the pre-trained `TRELLIS-image-large` weights for the SLAT VAE encoder and decoder and for the SLAT flow transformer $\mathcal{G}_L$ from commit `442aa1e` (Nov 5, 2025). For ablation studies implementing the sparse structure flow model, the model weights are also those from `TRELLIS-image-large`.

Context encoding. As described in Section 3.2, we encode the original scene to SLAT by selecting a region lying inside a bounding box. The generative prior used in this work, TRELLIS (Xiang et al., 2025b), operates in a latent space with a spatial resolution of 64. However, 3DGS scenes assume widely varying extents, and therefore large scenes cannot always be discretized with minimal information loss in a 64^3 space. The size of the context bounding box must be small enough to faithfully encode the appearance of the content, but large enough to provide sufficient information for sampling from the context-conditional distribution $\mathcal{P}_{\mathcal{G}_L}(\boldsymbol{z}_m|z_c)$. Although different scenes may naturally admit varying optimal context bounding box sizes, we fix the size of the bounding box in the experiments of this paper to the smallest bounding box where, after voxelization based on Gaussian primitive centers, the ratio of context to target region voxels is greater than 6. The orientation of this bounding box is detailed in Appendix B.3 below. The bounding box, lying in world space, is then normalized to the unit cube $[-0.5, 0.5]^3$, and Gaussian center positions are discretized to voxels lying in a 64^3 grid, which is the spatial resolution of the TRELLIS space. The normalization maps the $\mathbf{c}_{\text{world}} = (x, y, z)$ coordinates to:

$$\mathbf{c}_{\text{slat}} = \frac{\mathbf{c}_{\text{world}} - \mu}{s},$$

where μ is the position of the bounding box center and s is the size of the largest bounding box axis.

We then compute the DINOv2 (Oquab et al., 2023) features of 36 views sampled from a virtual orbit rotating around the normalized context, determined in the same way as Xiang et al. (2025b), with a render size of 518×518. As 3DGS scenes can contain large, low-opacity unwanted floater artifacts, we additionally drop voxels that are isolated in a $3 \times 3 \times 3$ neighborhood.

After generating the masked region Gaussians $G_m \sim \mathcal{P}_{\mathcal{G}_L}(G_m|G_c)$, we unnormalize their positions $\mathbf{c}_{\text{slat}}$ back to world space in order to merge generated and context Gaussians: $\mathbf{c}_{\text{world}} = \mathbf{c}_{\text{slat}} \cdot s + \mu$. Additionally, we apply a linear opacity ramp to the generated Gaussians lying up to 5 voxels outside of the target region's boundary, in order to ensure seamless transition between G_c and G_m.

Sparse structure prediction. Our structure prediction network implements a 3D residual U-Net architecture with attention in the bottleneck. We instantiate the model with an initial 64 convolutional channels, doubling over each of the three downsampling stages to reach 512 channels in the bottleneck. This results in a model with 49.7M parameters. The training protocol of our proposed sparse structure prediction model is described in Section C.1.

Progressive context conditioning. When evaluating the re-noised known context latents $\hat{\boldsymbol{z}}$ in Equation (1), we slightly remap the noise level, using $\left(10^{-5} + (1 - 10^{-5})t\right)\epsilon$ rather than $t\epsilon$. This exactly follows the protocol established by TRELLIS (Xiang et al., 2025b). We apply the RePaint scheme using 100 integration steps and 5 backwards jumps of length 10.

Self-attention masking. We conduct a hyperparameter grid search to determine the optimal flow transformer layers and integration time steps on which to perform self-attention masking. After a first coarse search, we observe that masking the late layers of the model at high-noise integration time steps is the best performing strategy. We then compare masking layers in range $19-24$, $18-23$ and $19-23$ while varying from $t = 1.0$ and $t = t_{\text{stop}}$ with $t_{\text{stop}} \in \{0.55, 0.6, 0.65, 0.7, 0.75, 0.8\}$. Hence, we choose to apply self-attention masking to layers $18-23$ of the 24-layer $\mathcal{G}_L$ flow model between $t = 1.0$ and $t = 0.8$. Applying self-attention masking at high-noise time steps allows the flow model to reconcile the trajectories of context and masked latents while the coarse appearance of the structured latents is forming, while unrestricted self-attention at later time steps allows target latents to attend to themselves in order to harmonize fine-grained detail.

Scene-conditioned decoder adaptation. Our scene-conditioned decoder adaptation module consists in a rank-16 LoRA adapter (Hu et al., 2021) attached to every block of the VAE decoder. Adapters are attached to each block's attention and MLP layers. Additionally, we unfreeze the final linear layer of the decoder, which is responsible for predicting the attributes of the 32 Gaussian primitives predicted for each voxel.

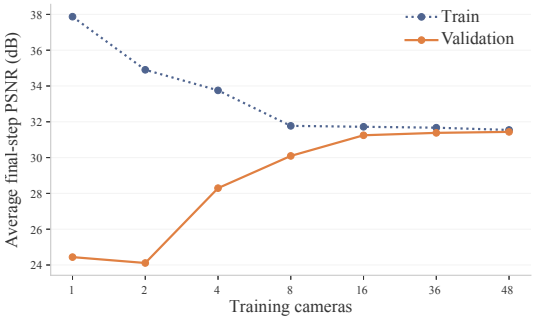


Figure 9: Train and validation viewpoint PSNR during scene-conditioned decoder adaptation vs. number of training cameras.

We train the adapter using the AdamW (Loshchilov & Hutter, 2018) optimizer with a learning rate of 10^{-4} for 600 steps. The supervisory renders are obtained by sampling 36 views from a virtual trajectory covering a 360° orbit around the context region, inherited from the virtual trajectory protocol of (Xiang et al., 2025b), which we modify by applying a sinusoidal modulation to the orbit radius, in order to capture range-dependent variations in the appearance of Gaussians. We additionally hold out 6 viewpoints from this trajectory to evaluate unseen viewpoint generalization.

The photometric loss $\mathcal{L}_{\text{data}}$ used to optimize the parameters ϕ is:

$$\mathcal{L}_{\text{data}} = \mathcal{L}_1 + \lambda_{\text{D-SSIM}}\mathcal{L}_{\text{D-SSIM}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} \tag{4}$$

We use $\lambda_{\text{D-SSIM}} = 0.2$ and $\lambda_{\text{LPIPS}} = 0.2$ for all experiments in this paper.

Figure 9 illustrates the impact of changing the number of training cameras for the scene-conditioned decoder adaptation stage. While using 1 to 8 cameras does not result in satisfying reconstruction, 16 or more cameras allow the adapter to faithfully match the original scene's appearance, with diminishing returns as the number of training cameras increase.

B.2 Baseline implementation details

We benchmark existing 3D Gaussian splatting methods in a controlled environment. All methods, including ours, receive an identical 3DGS scene as input as well as a predefined 3D segmentation mask. The per-Gaussian segmentation mask is created in 3D editing software and encompasses all Gaussians corresponding to the object to remove. The use of an existing 3D mask allows us to bypass the expensive and error-prone Gaussian segmentation stages of existing pipelines, enabling direct comparison of the inpainting mechanism of each.

Gaussian Grouping. We follow the official implementation of Gaussian Grouping (Ye et al., 2024). Instead of optimizing the 3D Gaussian splatting scene from scratch and performing the Gaussian segmentation pipeline, we use the same 3D mask $\mathcal{M}$ used by our method (and other implemented baselines), allowing for fair comparison of the inpainting frameworks themselves.

Inpaint360GS. Our experimental protocol, comprising an existing 3D mask, bypasses the object association and ID distillation stages of Inpaint360GS, instead using a known correct 3D segmentation mask identical to other approaches for fair comparison. We otherwise use the authors' released inpainting pipeline as provided, including virtual view generation, conditional color and depth inpainting, color-depth fusion and 3DGS fine-tuning. As described in the authors' implementation details, we run the inpainting pipeline with 2000 fine-tuning steps and loss parameters $\lambda_{D-SSIM} = 0.2$ and $\lambda_{LPIPS} = 0.005$. Whilst the authors prescribe running the method at $1/4$ resolution, we found that to have adverse effects on datasets where reference images have lower resolution. Instead, we evaluated the render size of the LaMa inpainted views and of the subsequent optimization renders at $r = \{256, 512, 1024, 2048\}$ (clamped at the maximum image resolution), and found a render size of 512 to offer the best results (On the COR-NeRF dataset, $r = 512$ obtains an FID of 36.10 while $r = 256, 1024$ and 2048 attain respectively $44.1, 39.6$ and 39.3). Additionally, we were unable to observe the reported 3 minutes inpainting time, instead measuring a slightly higher inpainting time of 202 seconds. Since our input modality does not include per-Gaussian instance identities for non-target objects, we do not use Inpaint360GS's auxiliary object removal mechanism in the inpainting pipeline, rather evaluating the inpainting and refinement component of the method.

InFusion. We implement InFusion (Liu et al., 2024) following the official code provided by the authors. We notice that InFusion struggles with 360° unbounded scene inpainting, offering poor depth unprojection and overall inpainting quality on our benchmarked datasets. As the official protocol does not restrict the position of the generated Gaussians, this resulted in large artifacts scattered in the direction of the unprojected depth with respect to the chosen view. Therefore, we decided to discard the generated Gaussians lying outside of the 3D mask $\mathcal{M}$, expanded by 5%, in order to provide a more faithful comparison of target region inpainting quality.

B.3 Automatic orientation of the SLAT grid

As mentioned in Section 3.2, we notice that TRELLIS' latent generation model $\mathcal{G}_L$ is more likely to generate lines and boundaries that are aligned with the axes of the voxel grid O that associate a spatial position to the latent features. In this section, we first analyze that the axis-aligned bias comes from the latent flow model $\mathcal{G}_L$ rather than other stages of the pipeline. We then describe how we address this issue in practice by fitting the orientation of the voxel grid to visual features of the context G_{c^*}.

Axis aligned bias. Figure 10 highlights and diagnoses the axis alignment bias of TRELLIS. First, we notice that the inpainting quality depends on the orientation of the SLAT grid, which we further illustrate in Figure 11. We quantify this degradation using LPIPS and show that this loss of quality does not depend on our attention masking **(a)**. We first rule out the possibility that the VAE (including structure prediction) causes the issue by empirically verifying that its auto-encoding is stable under rotation: we encode and decode the same scene under varying rotations around the vertical axis and measure no significant reconstruction error variation. However, we see in **(b)** that the latent flow model $\mathcal{G}_L$ is less reliable at predicting the correct flow velocity when the SLAT grid is misaligned. Future versions of TRELLIS or a prior more robust to orientation changes could mitigate this issue and further improve our results.

Automatic orientation. In practice, we assume that the vertical axis is already correct – a wrong vertical axis causes many other issues, e.g., in terms of user navigation, and the vertical axis is a global reference, so it is usually correct and the assumption holds. The orientation angle around this vertical axis to align with the TRELLIS prior, however, is specific to the local neighborhood, namely the cropped context G_{c^*}. We estimate it by analyzing a 512×512 top-down view of this context. We first harmonize the rendered image by subtracting its low frequencies (by Gaussian-blurring the image, $\sigma = 41$) and estimate the per-pixel dominant direction via the largest eigenvector of the local structure tensor (i.e., second-moment matrix) using a Sobel filter to estimate gradients. To improve the robustness of our estimation, we split the image into 8×8 tiles for which we compute the mean direction, then consider the median tile when sorting them by directional angle. To prevent the mask borders from impacting the orientation estimation, we zero the gradient in all pixels that lie within six pixels of the masked region. We finally rotate voxel grid O by this found angle around the vertical axis (or, equivalently, rotate the entire scene by the opposite angle).

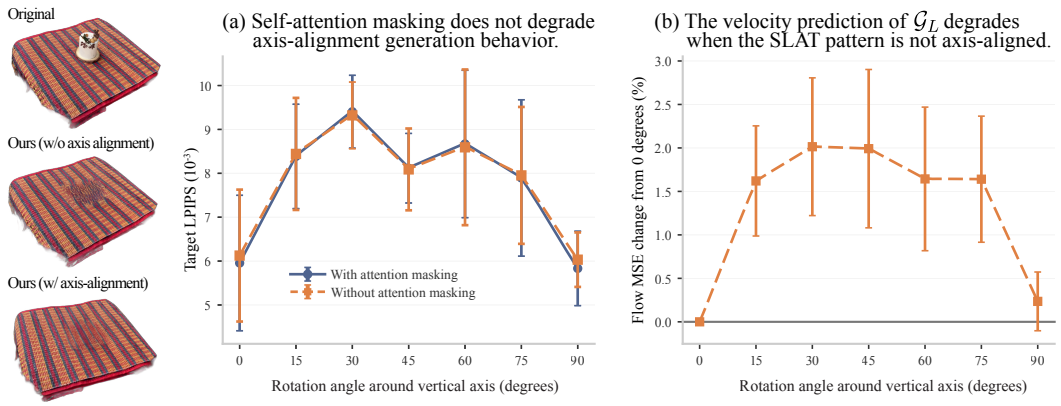


Figure 10: Illustration of the generative prior's bias towards generating axis-aligned patterns. (a) While self-attention masking neither degrades nor improves perceptual loss during inpainting for these examples, inpainting perceptually worsens between 0 and 90° rotations. (b) The flow velocity mean squared error also increases as the patterned SLAT grids are rotated, compared to an axis-aligned velocity at 0°.

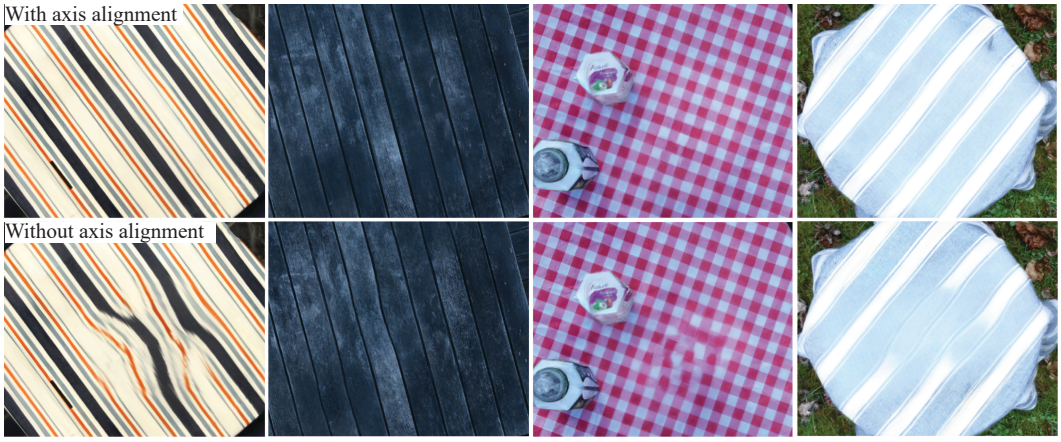


Figure 11: Qualitative visualization of the effect of our axis-alignment heuristic on inpainting results. Without it, the generation of context-consistent intricate patterns, being biased by the flow velocity towards alignment, is degraded.

C Dataset & Model Training

C.1 Sparse Structure Dataset Curation

We curate an internal dataset of 900 3D Gaussian splatting scenes depicting full 360° environments rather than asset-scale captures. None of the evaluation scenes from COR-NeRF, Inpaint360, or 360-USID are included in this dataset. We select 100 candidate cropped context regions per scene. We sample a context region bounding box with a per-axis extent sampled randomly between 2% and 80% of the original scene's scale. Inside this context region, we further sample a target region bounding box, whose per-axis extent is randomly sampled between 10% and 80% of the context region. Additionally, we reject context/target region candidate pairs containing less than 20 000 and 5 000 Gaussian primitives respectively. After sampling subscenes and accepting them based on the two criteria defined below, we apply data augmentation by horizontally flipping or rotating the structure by 90° increments, all with probability 0.25.

Objectness. The first criterion used to sample example pairs from the structure dataset depends on the presence or absence of objects in the target region, represented by local structure anomalies inside the masked region. We estimate the $3 \times 3 \times 3$ local covariance of every masked region voxel

$\mathbf{p}$ and approximate the surface variation in the following way: (Pauly et al., 2002)

$$\sigma(\mathbf{p}) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

with $\lambda_1 \geq \lambda_2 \geq \lambda_3$ the eigenvalues of the local covariance matrix. Small values of σ indicate locally planar structure. We reject a subscene when more than 30% of the target voxels show a variation $\sigma > 0.2$.

Structure predictability. In addition to the local structure covariance metric, we filter training dataset admissibility by computing target region predictability with respect to a quadric extrapolation of context. We partition the context structure C into connected components and estimate the PCA local tangent frame of the component adjacent to the masked region (Hoppe et al., 1992). Then, we fit a second-order polynomial height field (Cazals & Pouget, 2005) using context voxels:

$$f(u, v) = a_0 + a_1 u + a_2 v + a_3 u^2 + a_4 uv + a_5 v^2$$

Finally, a target region is deemed admissible if the fraction of target voxels lying close to the extrapolated height field is higher than 0.5. We note that this deliberately permissive admissibility test simply serves as a way to filter target regions unexplainable by context out of the dataset, rather than selecting only target regions predictable by quadric extrapolation of context. The curated dataset contains a large number of non-flat target regions, such as corners or curved surfaces: Figure 12 displays representative examples from the held-out test set, accepted by our data curation process and complex to predict with quadric extrapolation. While classic techniques such as plane fitting with RANSAC (Fischler & Bolles, 1981) could also be used to approximate individual examples, our learned structure prediction network is able to model occupancy completion from arbitrary context, without requiring any assumptions about its underlying structure.

To verify this, we conduct an ablation study by comparing the respective metrics of our structure completion network and the quadric extrapolation baseline, on both held-out test 3D Gaussian splatting scenes and on the inpainting benchmark datasets. On the structure reconstruction task, our network outperforms quadric extrapolation over Chamfer distance, F-score and IoU metrics, indicating stronger sparse structure prediction capabilities (Figure 13).

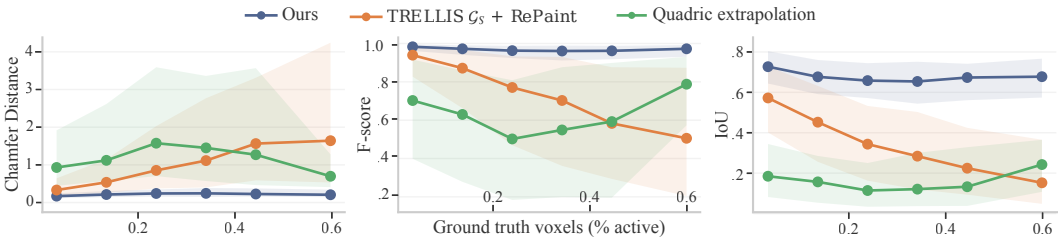


Figure 13: We ablate our proposed structure completion network with the TRELLIS backbone (see main text) and additionally show a comparison against a quadric-extrapolation baseline of the active ground truth voxels, which performs worse in all cases.

C.2 Model training

We divide the hereby constructed dataset into training, validation and test splits, using 650 scenes for training, 100 scenes for validation and 150 test scenes. Importantly, the train/test split is done at scene level and not at context/target region level, thus preventing data leakage. We train our model on 8 NVIDIA A100 GPUs for 300 epochs. We train using the Adam optimizer (Kingma & Ba, 2015), a batch size $B = 16$, with a cosine-annealed learning rate between 2×10^{-4} and 2×10^{-6}. As mentioned in Section 3.2, the loss function used for training is a linear combination of weighted cross-entropy and Dice loss: $\mathcal{L}_{\text{structure}} = 0.3\,\mathcal{L}_{\text{W-BCE}} + 0.7\,\mathcal{L}_{\text{DICE}}$. We use a weighted binary cross-entropy positive class weight of 100 during training. We conduct training with L2 weight regularization parameterized by $\lambda = 1 \times 10^{-4}$. After training for the full 300 epochs, the neural network weights achieving the lowest validation set loss are retained.

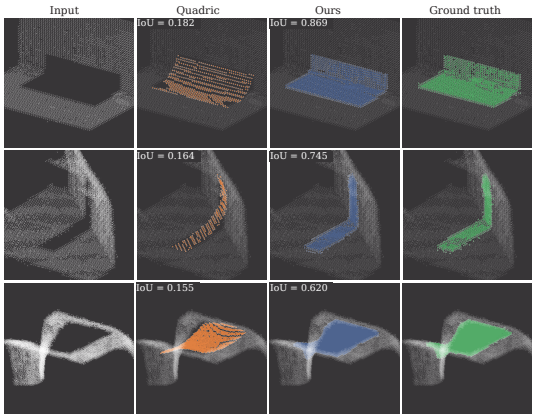


Figure 12: Quadric extrapolation against our learned structure prediction network for held-out test examples. Our model is able to successfully predict occupancy from complex context structure.

D Detailed per-scene results

Table 6 reports human preference rating for each scene used in the user study. We then report a per-scene breakdown of perceptual metrics: Tables 7, 8 and 9 detail the metrics of the four benchmarked methods, respectively on the COR-NeRF, Inpaint360 and 360-USID datasets.

Finally, Figures 14, 15 and 16 show qualitative comparison grids of the four benchmarked methods over all scenes in the three evaluated datasets. In addition, the reader is encouraged to visualize videos and additional comparison views in the electronic supplemental.

Table 6: Scene-level comparison of Ours and Inpaint360GS according to automated metrics and human preferences. Metric cells display within-metric normalized differences, where positive values take a green colour and favour Ours. The Human column reports the tie-adjusted preference for Ours. Across the 29 scenes, humans prefer Ours on 21, Inpaint360GS on 7 and rate the two as equal on one.

			Single-reference			Distributional			
Dataset	Scene	**Human (%)**	PSNR	SSIM	LPIPS	FID	Local-FID	KID	Local-KID
COR-NeRF	aircap	75.0							
	backpack	100.0							
	campingbox	88.9							
	dolls	66.7							
	gunnysack	100.0							
	icebox	55.6							
	mustard	30.0							
	pot	25.0							
	sign	58.3							
	trashcan	30.0							
	tree	28.6							
Inpaint360	bag	80.0							
	car	100.0							
	cone_red	75.0							
	cone_yellow	45.5							
	cube	100.0							
	doppelherz	94.4							
	fruits	21.4							
	garden_toys	92.3							
	redbull	91.7							
	toys	80.0							
	truck	50.0							
360-USID	carton	87.5							
	cone	96.2							
	cookie	77.8							
	newcone	83.3							
	plant	37.5							
	skateboard	88.9							
	sunflower	90.0							
	Spearman ρ		+0.14	+0.21	+0.48	+0.40	+0.40	+0.13	+0.28
	Sign agreement		54%	54%	64%	68%	61%	64%	54%

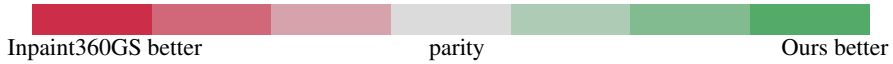


Table 7: Per-scene metrics on COR-NeRF. KID is reported $\times 10^3$. Human Pref. is each method's expected score against an average-skill opponent under the Davidson model, reported with a participant-bootstrap 95% confidence interval. Bold denotes the best method for each scene and metric.

		Reference-based			Distributional			Human
Scene	**Method**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**FID ↓**	**Local-FID ↓**	**KID ↓**	**Human Pref. ↑**
Aircap	Gaussian Grouping	20.35	0.599	0.259	44.04	41.31	4.26	48.3 ± 10.7
	InFusion	19.99	0.598	0.260	42.70	40.44	5.17	23.0 ± 10.7
	Inpaint360GS	**21.41**	0.612	0.251	37.65	34.47	2.28	63.4 ± 11.2
	Ours	21.13	**0.617**	**0.247**	**36.55**	**32.36**	**2.12**	$\mathbf{67.8 \pm 13.9}$
Backpack	Gaussian Grouping	21.34	0.856	0.211	60.66	57.02	13.40	16.2 ± 5.4
	InFusion	21.59	0.877	0.184	66.87	66.63	21.88	17.8 ± 8.7
	Inpaint360GS	23.04	0.899	0.159	40.91	42.19	**4.57**	65.0 ± 8.8
	Ours	**23.51**	**0.905**	**0.152**	**39.34**	**34.95**	5.49	$\mathbf{92.4 \pm 1.5}$
Campingbox	Gaussian Grouping	21.72	0.583	0.260	39.59	32.22	6.47	49.7 ± 8.6
	InFusion	20.21	0.575	0.287	124.72	121.77	79.75	10.7 ± 3.7
	Inpaint360GS	22.95	0.608	0.247	**29.42**	26.23	**1.36**	64.6 ± 6.9
	Ours	**23.14**	**0.610**	**0.244**	29.62	**26.16**	2.72	$\mathbf{82.9 \pm 7.3}$
Dolls	Gaussian Grouping	20.79	0.844	0.204	70.06	63.51	15.34	25.5 ± 8.2
	InFusion	21.14	0.865	0.173	60.73	52.64	9.95	42.9 ± 6.1
	Inpaint360GS	23.32	0.889	0.159	57.95	62.16	7.60	53.8 ± 6.3
	Ours	**25.27**	**0.904**	**0.139**	**43.31**	**37.64**	**3.61**	$\mathbf{77.0 \pm 8.0}$
Gunnysack	Gaussian Grouping	19.71	0.628	0.236	45.65	43.26	7.41	24.0 ± 9.4
	InFusion	19.21	0.648	0.218	34.38	33.67	3.12	30.8 ± 10.1
	Inpaint360GS	**20.35**	**0.654**	0.216	34.41	32.52	3.21	43.3 ± 8.1
	Ours	20.17	0.650	**0.214**	**32.01**	**31.44**	**1.46**	$\mathbf{89.7 \pm 2.4}$
Icebox	Gaussian Grouping	21.15	0.529	0.297	31.46	26.59	3.53	45.9 ± 14.6
	InFusion	22.21	0.547	0.293	34.50	34.44	5.15	19.3 ± 6.4
	Inpaint360GS	**23.15**	**0.558**	**0.281**	26.07	**22.55**	1.97	65.7 ± 14.0
	Ours	23.01	0.556	**0.281**	**25.93**	22.88	**1.59**	$\mathbf{72.4 \pm 10.7}$
Mustard	Gaussian Grouping	18.61	0.786	0.238	68.07	46.90	17.59	36.1 ± 9.4
	InFusion	21.65	0.839	0.196	42.89	33.72	4.58	40.0 ± 9.8
	Inpaint360GS	**23.37**	**0.863**	**0.175**	**36.12**	**30.15**	**0.99**	$\mathbf{67.2 \pm 8.7}$
	Ours	21.31	0.833	0.196	44.15	33.77	6.77	56.4 ± 8.7
Pot	Gaussian Grouping	19.65	0.575	0.255	65.31	65.84	15.75	44.5 ± 12.9
	InFusion	19.49	0.578	0.254	58.79	56.79	12.73	34.1 ± 14.1
	Inpaint360GS	20.29	**0.593**	**0.242**	**47.63**	**46.25**	**7.20**	$\mathbf{66.7 \pm 14.1}$
	Ours	**20.34**	0.588	0.246	60.88	64.06	15.39	54.6 ± 12.8
Sign	Gaussian Grouping	21.22	0.691	0.188	33.60	30.35	2.85	30.1 ± 13.5
	InFusion	21.37	0.704	0.177	27.26	24.32	1.18	43.7 ± 12.9
	Inpaint360GS	**22.18**	**0.712**	0.173	26.22	23.48	0.79	60.9 ± 12.6
	Ours	21.93	0.711	**0.172**	**26.16**	**23.16**	**0.41**	$\mathbf{65.8 \pm 11.7}$
Trashcan	Gaussian Grouping	21.96	0.585	0.217	28.83	21.79	0.95	52.6 ± 12.2
	InFusion	22.65	0.598	0.213	28.28	20.37	1.40	26.4 ± 11.8
	Inpaint360GS	23.28	**0.604**	0.207	**23.95**	18.40	$\mathbf{-0.39}$	$\mathbf{62.1 \pm 13.6}$
	Ours	**23.33**	**0.604**	**0.205**	**23.95**	**18.13**	-0.22	60.7 ± 13.7
Tree	Gaussian Grouping	21.38	0.582	0.282	41.35	34.79	2.53	$\mathbf{71.0 \pm 10.3}$
	InFusion	22.39	0.614	0.264	37.31	33.12	2.20	22.0 ± 9.5
	Inpaint360GS	**23.27**	**0.627**	**0.256**	36.71	32.63	1.41	63.4 ± 11.0
	Ours	**23.27**	0.625	**0.256**	**34.26**	**29.46**	**0.53**	45.8 ± 7.8
Average	Gaussian Grouping	20.72	0.660	0.241	48.06	42.14	8.19	39.6 ± 4.6
	InFusion	21.08	0.677	0.229	50.77	47.08	13.37	27.5 ± 3.3
	Inpaint360GS	**22.42**	**0.693**	0.215	36.10	33.73	**2.82**	61.2 ± 3.4
	Ours	22.40	0.691	**0.214**	**36.02**	**32.18**	3.63	$\mathbf{71.8 \pm 4.1}$
Median	Gaussian Grouping	21.15	0.599	0.238	44.04	41.31	6.47	—
	InFusion	21.37	0.614	0.218	42.70	34.44	5.15	—
	Inpaint360GS	**23.04**	**0.627**	0.216	36.12	32.52	**1.97**	—
	Ours	23.01	0.625	**0.214**	**34.26**	**31.44**	2.12	—

Table 8: Per-scene metrics on Inpaint360. KID is reported $\times 10^3$. Human Pref. is each method's expected score against an average-skill opponent under the Davidson model, reported with a participant-bootstrap 95% confidence interval. Bold denotes the best method for each scene and metric.

		Reference-based			Distributional			Human
Scene	**Method**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**FID ↓**	**Local-FID ↓**	**KID ↓**	**Human Pref. ↑**
Bag	Gaussian Grouping	24.84	0.835	0.148	35.46	35.38	−0.55	25.9 ± 6.9
	InFusion	25.16	0.842	0.157	60.34	70.92	15.04	10.7 ± 4.3
	Inpaint360GS	**27.47**	**0.856**	**0.133**	**29.08**	**25.85**	**−3.55**	75.0 ± 7.8
	Ours	27.20	0.848	0.136	30.28	27.20	−2.44	$\mathbf{88.8 \pm 5.1}$
Car	Gaussian Grouping	18.91	0.759	0.184	127.19	122.27	66.05	29.9 ± 5.7
	InFusion	17.58	0.721	0.217	171.26	143.10	108.98	25.7 ± 6.3
	Inpaint360GS	**21.74**	**0.774**	0.176	81.52	83.63	**16.21**	53.6 ± 6.7
	Ours	21.55	0.740	**0.172**	**79.75**	**67.69**	27.74	$\mathbf{85.7 \pm 5.5}$
Cone Red	Gaussian Grouping	20.46	0.726	0.189	33.16	30.22	6.30	28.1 ± 6.9
	InFusion	20.29	0.726	0.197	58.43	53.38	13.31	11.0 ± 5.2
	Inpaint360GS	20.60	**0.736**	**0.182**	22.36	**18.42**	0.32	76.2 ± 8.2
	Ours	**20.61**	0.733	0.184	**21.23**	20.81	**−1.44**	$\mathbf{86.7 \pm 7.2}$
Cone Yellow	Gaussian Grouping	21.23	0.727	0.192	46.08	44.67	7.03	32.4 ± 6.7
	InFusion	21.44	0.774	0.171	128.75	130.81	67.78	10.1 ± 2.8
	Inpaint360GS	21.78	**0.782**	**0.155**	**27.79**	24.30	**−9.04**	$\mathbf{82.6 \pm 5.1}$
	Ours	**21.90**	0.777	0.157	28.41	**19.42**	−8.42	80.3 ± 5.7
Cube	Gaussian Grouping	19.52	0.802	0.195	90.24	118.28	31.50	35.4 ± 8.9
	InFusion	18.63	0.831	0.175	88.03	110.25	23.85	16.3 ± 6.4
	Inpaint360GS	22.22	**0.866**	0.143	45.71	43.89	−1.15	49.0 ± 7.6
	Ours	**22.55**	0.864	**0.122**	**35.46**	**23.47**	**−4.87**	$\mathbf{90.2 \pm 3.4}$
Doppelherz	Gaussian Grouping	22.65	0.790	0.146	53.23	49.82	−6.91	17.5 ± 6.6
	InFusion	24.31	0.827	0.126	49.59	45.11	−4.29	24.6 ± 9.0
	Inpaint360GS	25.42	0.843	**0.118**	39.21	29.92	−10.29	61.4 ± 10.0
	Ours	**26.06**	**0.846**	0.119	**37.12**	**28.53**	**−11.47**	$\mathbf{89.7 \pm 4.2}$
Fruits	Gaussian Grouping	24.28	0.866	0.171	35.62	28.05	−5.97	26.1 ± 9.0
	InFusion	23.89	0.869	0.166	31.11	25.44	**−8.47**	23.4 ± 8.9
	Inpaint360GS	**25.42**	**0.879**	**0.158**	**29.05**	**23.14**	−8.10	$\mathbf{85.1 \pm 6.2}$
	Ours	25.16	0.875	0.162	31.42	25.32	−6.88	61.3 ± 15.3
Garden Toys	Gaussian Grouping	23.12	0.752	0.186	53.02	45.44	−5.52	43.6 ± 9.7
	InFusion	22.52	0.756	0.185	55.04	44.48	−2.02	12.4 ± 5.0
	Inpaint360GS	**23.58**	**0.763**	**0.178**	**43.49**	**40.59**	**−10.09**	51.2 ± 13.1
	Ours	23.14	0.747	0.189	55.23	46.80	−3.20	$\mathbf{89.5 \pm 4.3}$
Redbull	Gaussian Grouping	23.71	0.751	0.185	46.48	46.20	14.73	29.1 ± 6.0
	InFusion	22.90	0.748	0.186	89.76	100.81	50.38	9.4 ± 4.2
	Inpaint360GS	24.07	**0.756**	0.176	**31.87**	34.55	**2.47**	70.4 ± 9.3
	Ours	**24.15**	**0.756**	**0.174**	32.05	**29.20**	3.46	$\mathbf{90.8 \pm 3.2}$
Toys	Gaussian Grouping	**18.58**	**0.237**	0.527	100.35	76.14	58.16	64.6 ± 8.9
	InFusion	18.22	0.208	0.535	103.88	90.94	53.91	13.3 ± 3.5
	Inpaint360GS	17.92	0.216	0.526	**93.21**	**65.85**	**52.96**	53.1 ± 10.6
	Ours	18.14	0.225	**0.525**	94.61	66.16	53.98	$\mathbf{76.6 \pm 7.6}$
Truck	Gaussian Grouping	20.87	0.837	0.180	59.84	70.15	19.72	30.2 ± 6.8
	InFusion	22.35	0.861	0.162	88.96	105.32	54.61	29.9 ± 7.5
	Inpaint360GS	24.11	**0.876**	**0.144**	**38.15**	50.94	**4.37**	64.0 ± 9.8
	Ours	**24.60**	0.872	0.149	40.20	**46.71**	7.24	$\mathbf{75.3 \pm 7.8}$
Average	Gaussian Grouping	21.65	0.735	0.209	61.88	60.60	16.78	29.7 ± 5.4
	InFusion	21.57	0.742	0.207	84.10	83.69	33.92	12.4 ± 3.5
	Inpaint360GS	23.12	**0.759**	**0.190**	**43.77**	40.10	**3.10**	69.3 ± 4.8
	Ours	**23.19**	0.753	**0.190**	44.16	**36.48**	4.88	$\mathbf{88.1 \pm 3.7}$
Median	Gaussian Grouping	21.23	0.759	0.185	53.02	46.20	7.03	—
	InFusion	22.35	0.774	0.175	88.03	90.94	23.85	—
	Inpaint360GS	**23.58**	**0.782**	**0.158**	38.15	34.55	−1.15	—
	Ours	23.14	0.777	0.162	**35.46**	**28.53**	**−2.44**	—

Table 9: Per-scene metrics on 360-USID. KID is reported $\times 10^3$. Human Pref. is each method's expected score against an average-skill opponent under the Davidson model, reported with a participant-bootstrap 95% confidence interval. Bold denotes the best method for each scene and metric.

		Reference-based			Distributional			Human
Scene	**Method**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**FID ↓**	**Local-FID ↓**	**KID ↓**	**Human Pref. ↑**
Carton	Gaussian Grouping	20.46	0.459	0.324	44.84	68.15	15.06	36.1 ± 10.5
	InFusion	21.01	0.498	0.309	87.81	128.00	59.90	12.1 ± 5.2
	Inpaint360GS	21.38	**0.499**	0.302	37.63	63.23	11.52	72.0 ± 10.3
	Ours	**21.44**	0.495	**0.298**	**34.51**	**53.13**	**8.04**	$\mathbf{83.7 \pm 8.2}$
Cone	Gaussian Grouping	16.49	0.183	**0.319**	55.10	67.60	2.71	$\mathbf{77.4 \pm 10.0}$
	InFusion	16.45	0.184	0.326	68.07	84.39	7.95	27.9 ± 10.0
	Inpaint360GS	16.55	0.186	0.323	61.02	83.75	4.22	28.6 ± 9.8
	Ours	**16.56**	**0.187**	**0.319**	**51.26**	**64.72**	**−0.08**	65.1 ± 9.4
Cookie	Gaussian Grouping	18.11	0.478	0.366	195.05	218.27	79.21	19.4 ± 8.2
	InFusion	17.79	0.480	0.365	195.85	188.97	61.92	56.1 ± 11.2
	Inpaint360GS	**19.44**	**0.508**	**0.349**	161.61	**179.79**	43.58	55.0 ± 11.0
	Ours	19.32	0.505	0.351	**158.88**	185.81	**33.72**	$\mathbf{73.1 \pm 10.3}$
Newcone	Gaussian Grouping	21.62	0.447	0.311	88.21	75.68	27.26	31.4 ± 9.7
	InFusion	21.48	0.447	0.309	96.69	86.83	44.25	29.1 ± 9.6
	Inpaint360GS	21.64	**0.459**	**0.301**	**77.33**	**69.06**	**14.77**	60.7 ± 10.7
	Ours	**21.67**	0.458	0.307	81.49	74.78	21.56	$\mathbf{77.3 \pm 10.3}$
Plant	Gaussian Grouping	19.02	0.508	0.319	83.89	77.67	22.30	29.5 ± 10.9
	InFusion	18.65	0.490	0.341	105.98	131.25	44.07	16.4 ± 7.3
	Inpaint360GS	19.12	**0.514**	**0.312**	49.79	56.35	1.55	77.3 ± 11.0
	Ours	**19.25**	0.500	0.315	**44.94**	**54.00**	**−3.22**	$\mathbf{78.3 \pm 8.3}$
Skateboard	Gaussian Grouping	19.87	0.378	**0.300**	38.11	40.12	3.07	62.5 ± 10.6
	InFusion	17.52	0.378	0.316	256.83	194.96	271.66	9.8 ± 3.7
	Inpaint360GS	19.90	0.383	0.303	38.45	40.79	3.07	49.1 ± 12.2
	Ours	**20.02**	**0.387**	0.303	**35.66**	**39.61**	**1.69**	$\mathbf{85.6 \pm 7.9}$
Sunflower	Gaussian Grouping	19.98	0.745	0.262	131.87	130.43	54.05	24.9 ± 10.9
	InFusion	22.74	0.784	0.238	107.69	112.66	25.68	39.5 ± 11.0
	Inpaint360GS	**23.85**	**0.805**	0.222	89.64	91.26	9.11	53.0 ± 12.4
	Ours	23.63	0.802	**0.221**	**81.22**	**80.68**	**6.45**	$\mathbf{80.2 \pm 10.5}$
Average	Gaussian Grouping	19.36	0.457	0.314	91.01	96.85	29.10	37.0 ± 4.5
	InFusion	19.38	0.466	0.315	131.27	132.44	73.63	27.0 ± 5.2
	Inpaint360GS	**20.27**	**0.479**	**0.302**	73.64	83.46	12.55	53.4 ± 4.8
	Ours	**20.27**	0.476	**0.302**	**69.71**	**78.96**	**9.74**	$\mathbf{79.8 \pm 4.5}$
Median	Gaussian Grouping	19.87	0.459	0.319	83.89	75.68	22.30	—
	InFusion	18.65	0.480	0.316	105.98	128.00	44.25	—
	Inpaint360GS	19.90	**0.499**	**0.303**	61.02	69.06	9.11	—
	Ours	**20.02**	0.495	0.307	**51.26**	**64.72**	**6.45**	—

E DETAILED PER-SCENE VISUALIZATIONS



Figure 14: Qualitative comparison of our method and the implemented baselines on the COR-NeRF dataset. Our WINGS framework provides sharp, consistent inpainting across the majority of the scenes in this dataset.



Figure 15: Qualitative comparison of our method and the implemented baselines on the Inpaint360 dataset.

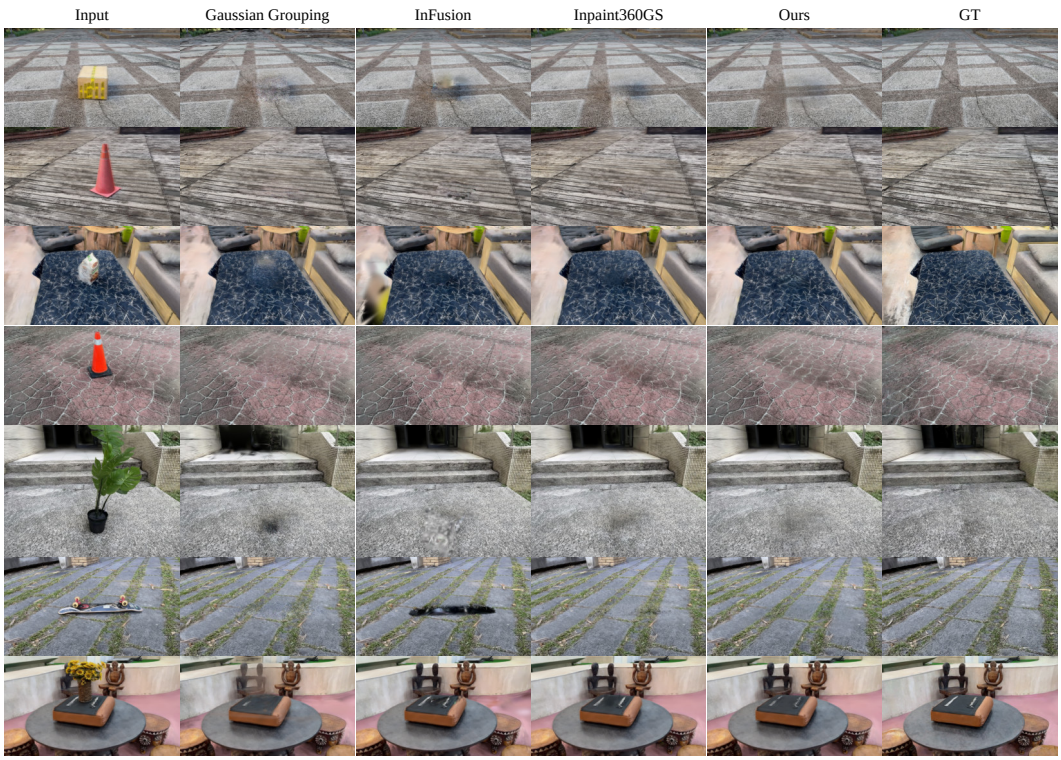


Figure 16: Qualitative comparison of our method and the implemented baselines on the 360-USID dataset.