

OccludeNeRF: Geometric-aware 3D Scene Inpainting with Collaborative Score Distillation in NeRF

Jingyu Shi

Futurewei Technologies
Purdue University
shi537@purdue.edu

Xiang Gao

Futurewei Technologies
Stony Brook University
gao2@cs.stonybrook.edu

Achleshwar Luthra

Futurewei Technologies
Texas A&M University
achleshwarluthra@gmail.com

Xiyun Song

Futurewei Technologies
xsong@futurewei.com

Jiazhil Li

Futurewei Technologies
University of Southern California
jiazhil@usc.edu

Zongfang Lin

Futurewei Technologies
zlin1@futurewei.com

David Gu

Stony Brook University
gu@cs.stonybrook.edu

Heather Yu

Futurewei Technologies
hyu@futurewei.com

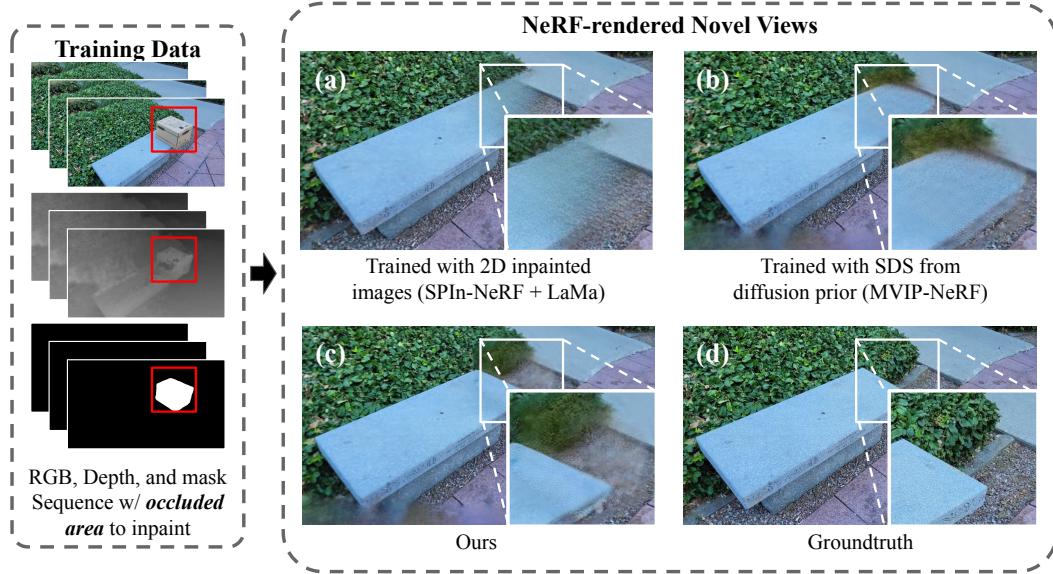


Figure 1. Given sequences of masked RGB and depth images, prior works (a) and (b) can train a 3D NeRF of the scene with the masked region inpainted. While showing promising visual fidelity in rendered novel views, the occluded region is not faithfully reconstructed due to limited prior information. E.g. the rendered bench extends further than that in the groundtruth (d). Our method (c) incorporates and propagates the limited information to multi-view updates of the NeRF and achieves a more faithful reconstruction of the groundtruth.

Abstract

With Neural Radiance Fields (NeRFs) arising as a powerful 3D representation, research has investigated its various downstream tasks, including inpainting NeRFs with 2D images. Despite successful efforts addressing the view

consistency and geometry quality, prior methods yet suffer from occlusion in NeRF inpainting tasks, where 2D prior is severely limited in forming a faithful reconstruction of the scene to inpaint.

To address this, we propose a novel approach that enables cross-view information sharing during knowledge dis-

tillation from a diffusion model, effectively propagating occluded information across limited views. Additionally, to align the distillation direction across multiple sampled views, we apply a grid-based denoising strategy and incorporate additional rendered views to enhance cross-view consistency. To assess our approach’s capability of handling occlusion cases, we construct a dataset consisting of challenging scenes with severe occlusion, in addition to existing datasets. Compared with baseline methods, our method demonstrates better performance in cross-view consistency and faithfulness in reconstruction, while preserving high rendering quality and fidelity.

1. Introduction

Neural Radiance Fields (NeRFs) [33] have emerged as a revolutionary approach to 3D scene representation, showcasing high performance in novel view synthesis. Recent research has explored diverse NeRF applications across domains, including Augmented and Virtual Reality, game development, and computer-aided design. Among all, a challenging task is to inpaint a 3D NeRF scene, i.e. to remove an undesired area and complete the area with visually coherent and geometrically plausible content that can be rendered consistently across multiple views.

Inpainting 3D NeRF scenes presents significant challenges. First, training a NeRF requires 2D images from multiple viewpoints with consistent inpainting to minimize artifacts and enhance visual realism. Prior approaches address this by focusing on the consistency of 2D inpainted images or incorporating consistent implicit knowledge distillation from diffusion-based 2D inpainting models, such as LaMa [35, 51, 56]. Alternatively, diffusion models have also gained attention as the 2D inpainter to generate 2d inpainted images [24, 53, 55]. In addition to using explicit 2D inpainting, some prior works utilize Score Distillation Sampling (SDS) [38] to distill generative prior from diffusion models for multiple views [6, 39, 53]. Despite achieving breakthroughs in rendering quality and multiview consistency, these methods remain challenged by occlusions in 3D NeRF inpainting, where the areas to inpaint are often obscured by objects to be removed, resulting in inconsistent 2D distillation or explicit inpainting over the occluded area across the views. Such occlusions limit available prior information about the 3D scene, leading to incomplete or unfaithful reconstructions, as shown in Fig. 1.

This work addresses the challenge of faithfully reconstructing occluded regions by leveraging information from the limited numbers of occluded views to infer constrained scene priors. Our approach utilizes multi-view information to guide the inpainting direction, resulting in consistent and faithful reconstructions of the original scenes without compromising rendering quality.

To this end, we present Occlude-NeRF, a novel approach

to mitigate the occlusion challenge in 3D NeRF Inpainting while ensuring 3D consistency across multiple views. Our method uses RGB and Depth images with corresponding binary masks marking inpainted regions as inputs. We first train a NeRF on the masked images to reconstruct the background. Meanwhile, to inpaint the masked regions, we followed MVIP-NeRF [6], using an SDS training scheme to obtain inpainting guidance from an off-the-shelf diffusion model [40]. To incorporate information from partially unoccluded views, we apply Collaborative Score Distillation Sampling (CDS) [20], which smooths the gradient update with information from other views and propagates the guidance among the views. To maximize information sharing among the views, we design a reference-view paradigm during training. We render two sets of views, with one used for loss back-propagation and the other only leveraged as the reference with no gradient computation. To further ensure consistency among multiple views during one distillation step, we applied a grid-denoising pattern in our noise prediction step, inspired by similar findings of prior works [1, 55]. By comparison with baseline methods and ablations of our features, we demonstrate how our method handles the occlusion challenge in 3D NeRF inpainting tasks. We propose a novel approach to seamlessly and faithfully inpainting severely occluded 3D scenes. Our code and dataset will be publicly available on GitHub. Our contributions are listed as follows:

- A modified CDS approach with multi-view information sharing in 3D NeRF inpainting task based on diffusion models, to mitigate the occlusion in inpainting areas.
- A grid-denoising pattern during score distillation to visually prompt the diffusion model to denoise towards consistent inpainting of distinct viewpoints.
- A reference-view training paradigm to increase the cross-view information sharing during NeRF training.

To assess the efficacy and performance of our method, in addition to existing datasets [32, 35], we construct

- A novel and challenging dataset for 3D NeRF inpainting, featuring scenes where the regions to inpaint suffer from occlusion and prior information for inpainting is limited.

2. Related Work

2.1. NeRF Inpainting

NeRF, or Neural Radiance Fields [33], have emerged as powerful representations for synthesizing novel views of complex 3D scenes with high fidelity. One potential use case of NeRFs is editing or inpainting a scene [57, 59], which involves filling in or reconstructing missing or corrupted regions to align seamlessly with the context.

NeRF editing can be done by adjusting the color and shape codes [27] supervised by priors from other models [11, 34, 52]. Inpainting, however, requires more than

straightforward scene editing; it necessitates that inpainted regions visually and geometrically integrate with the original scene. To tackle these challenges, previous methods have emphasized generating consistent 2D inpainted images [5, 9, 36, 51, 58] to guide NeRF optimization or training. Prior methods such as NeRF-In [25], SPIIn-NeRF [35], Liu et. al. [28] and Remove-NeRF [56] approach this problem by inpainting 2D images with a 2D inpainter [5, 40] and constrain the NeRF training with both inpainted 2D images and the 3D consistency among them.

The advent of diffusion models has significantly advanced 2D inpainting capabilities. Later methods adopt diffusion models as their 2D inpainters while incorporating 3D constraints into the inpainting process, such as visual prompting (NeRFiller [55]), fine-tuning and adversarial training (MALD-NeRF [24]), and 3D self-attention among views (CAT3D [10]).

Despite these advancements, methods that rely solely on 2D inpainted images face challenges in achieving complete geometry and multi-view consistency due to variations in 2D inpainting. MVIP-NeRF [10] addresses this limitation by incorporating SDS loss as a multi-view rendering loss, guiding NeRF training with both visual and geometric cues. This approach has also been utilized in object-level editing [61] and scene-level inpainting [39].

However, existing methods often struggle when severe occlusion is present in the scene and 2D priors provide limited information about the inpainted regions, complicating faithful scene reconstruction. Methods like NeRF-W [31] and Ha-NeRF [7] can remove transient objects from NeRF that occlude the area of interest but are less effective for static objects. Zhu et. al. [63] proposed a method to remove the occluding static object from the scene. However, their method is constrained by the presumption that the occluding object is closer than the background in the scene, while our Occlude-NeRF presents a generalized solution with no such constraints.

Occlude-NeRF builds on the use of SDS loss in NeRF inpainting, focusing specifically on addressing the challenges posed by occlusion. Specifically, we aim to inpaint and reconstruct the 3D scenes faithfully to the original scenes by enabling information sharing and aligning the distillation direction in SDS.

2.2. SDS with Diffusion Models

Diffusion models [8, 16, 48, 49] refine samples by progressive denoising from noise using a learned denoising process to approximate the target data distribution. With their ability to model complex data distributions starting from simple ones, such as Gaussian distributions, diffusion models have become the de facto state-of-the-art for image generation.

Beyond sampling in the image space, diffusion models have also been applied to 3D parameter space sampling.

DreamFusion [38] first introduced the framework of Score Distillation Sampling (SDS), which optimizes parameterized models, such as differentiable image generators like NeRF or 3D Gaussian Splatting (3DGS) [19], by distilling rich 2D priors from a pre-trained diffusion model to guide the image generator. SDS has liberated research in 3D vision from the constraints of expensive 3D training data and inspired numerous variants aimed at tackling different 3D challenges [13, 20, 54, 62, 65], such as enhancing gradient clarity during sampling [13], ensuring multi-view consistency [65]. Among all, Collaborative Score Distillation (CDS) [20] facilitates consistent visual synthesis by constraining a smooth vector function within a Reproducing Kernel Hilbert Space (RKHS) while approximating a target distribution through Stein Variational Gradient Descent [26]. While CDS has primarily been applied to panorama images, video, and 3D scene generation, our work extends its capability for 3D scene inpainting tasks by deriving a parameter update paradigm that enhances multi-view information sharing.

3. Preliminary

3.1. Neural Radiance Fields (NeRF)

Neural Radiance Fields (NeRF) provide a continuous and differentiable method for representing 3D scenes using a neural network. NeRF parameterizes the scene as a volumetric field that maps a 3D position and a viewing direction to color and density values. Formally, NeRF can be represented as a function: $F_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$, where $\mathbf{x} \in \mathbb{R}^3$ denotes a point in 3D space, $\mathbf{d} \in \mathbb{R}^2$ represents the viewing direction, $\mathbf{c} \in \mathbb{R}^3$ is the RGB color at the queried point, $\sigma \in \mathbb{R}^+$ is the volume density, θ indicates the learnable parameters of the network.

NeRF employs volume rendering to synthesize images from this 3D representation. The color observed along a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where \mathbf{o} is the camera origin and \mathbf{d} is the direction, is computed as: $C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt$, where t_n and t_f denote the near and far bounds along the ray, $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right)$ represents the accumulated transmittance from t_n up to point t , describing the probability that the ray has not been occluded.

3.2. Score Distillation Sampling

SDS is an alternative sample generation method proposed by Poole et al. [38]. By distilling the knowledge from a 2D text-to-image model, usually a pre-trained diffusion model, SDS optimizes a differentiable image generator (e.g. NeRF or 3DGS) towards a set of 3D parameters that renders high-fidelity images.

Let $\mathbf{x} = g(\theta)$ be an image rendered by a differentiable generator g parameterized by θ . SDS minimizes the density

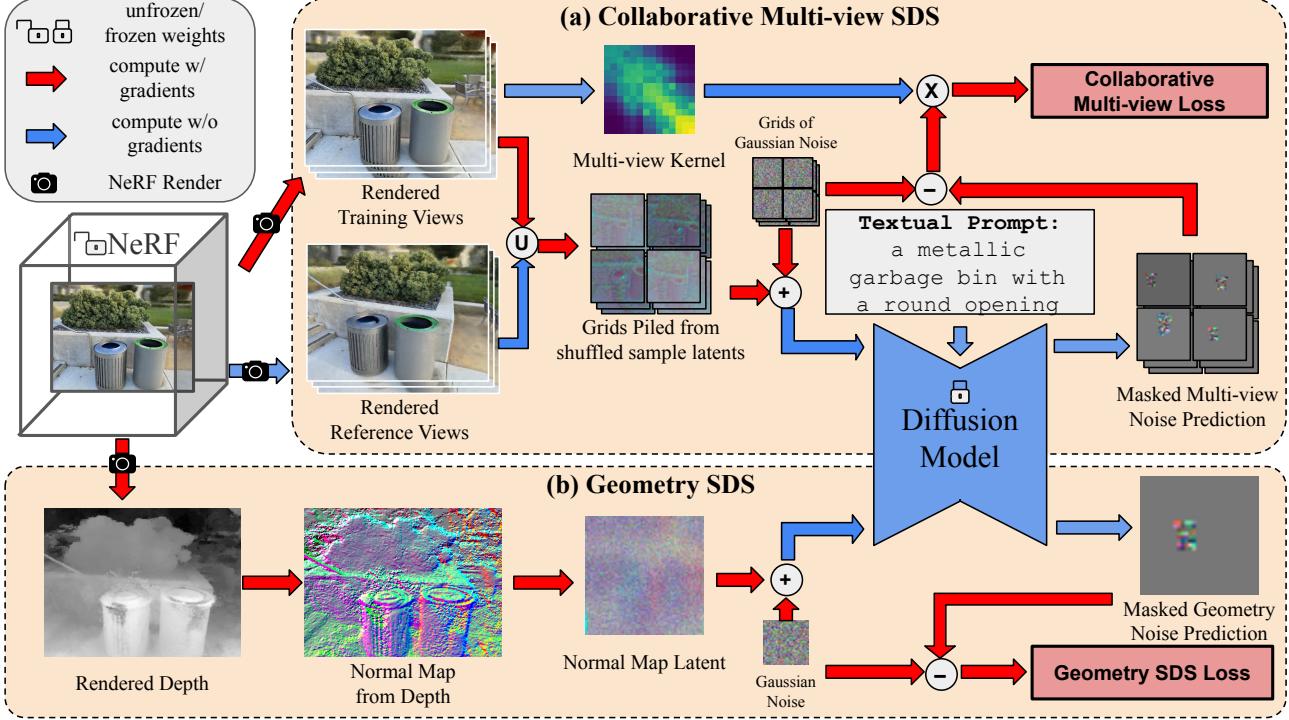


Figure 2. The overall workflow. At each iteration, our method takes masked RGB and depth images as input and back-propagates the pixel-wise loss in the unmasked RGB and depth to reconstruct the background (Eq. (4)). For the masked regions, we render a set of training views and a set of reference views, respectively. For the reference views, the gradients are disabled. We randomly sample from the union of the two sets and encode grids of latents, which are then added to a grid of Gaussian noise. We pass the grids to the diffusion model conditioned on a textual prompt describing the scene and obtain a masked prediction of the noise. We then compute a collaborative multi-view loss (Eq. (8)) with a multi-view kernel computed from the training set, assessing how much information to share among the training views. We apply a similar geometry SDS loss as in [6] (Eq. (5)). Note that in addition to the masked loss in the figure, we compute the pixel-wise losses in unmasked RGB and depth renderings as well. All losses are backpropagated to optimize the NeRF.

distillation loss [37], which is the KL divergence between the posterior of \mathbf{x} and the text-conditional density p_ϕ^ω :

$$L(\theta; \mathbf{x}) = \mathbb{E}_{t,\epsilon}[\alpha_t/\sigma_t D_{KL}(q(\mathbf{x}_t|\mathbf{x})||p_\phi^\omega(\mathbf{x}_t; y, t))] \quad (1)$$

where t is the denoising timestep and y is the embedded textual prompt. α_t is the scale and σ_t is the noise variance at t , together defining the noise scheduling. To update θ , SDS computes the gradient of the loss by:

$$\nabla_\theta L(\theta; \mathbf{x}) = \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_\phi^\omega(\mathbf{x}_t; y, t) - \epsilon)] \frac{\partial \mathbf{x}}{\partial \theta} \quad (2)$$

where $w(t)$ is a weighting function. Derived from SDS, CDS [20] aims to update a set of parameters $\{\theta_i\}_{i=1}^N$ that parameterize the image generator g for the images $\mathbf{x}_t^{(i)} = g(\theta_i)$. CDS solves the minimization of distillation loss in Equation 1 by using Stein Variational Gradient Descent (SVGD) [26] in order to update each θ_i synchronously within the set $\{\theta_i\}_{i=1}^N$:

$$\begin{aligned} \nabla_{\theta_i} L(\theta_i; \mathbf{x}) &= \frac{w(t)}{N} \sum_{j=1}^N (k(\mathbf{x}_t^{(j)}, \mathbf{x}_t^{(i)}) (\epsilon_\phi^\omega(\mathbf{x}_t^{(i)}; y, t) - \epsilon) \\ &\quad + \nabla_{\mathbf{x}_t^{(j)}} k(\mathbf{x}_t^{(j)}, \mathbf{x}_t^{(i)}) \frac{\partial \mathbf{x}^{(i)}}{\partial \theta_i}) \end{aligned} \quad (3)$$

where $k(\cdot, \cdot) : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^+$ is a positive definite kernel corresponding to a RKHS.

4. Methodology

In this section, we present our proposed method for incorporating multi-view information to address the challenge of occlusion in 3D NeRF inpainting.

Our pipeline is illustrated in Fig. 2. Given a set of RGB images of a scene, corresponding depth images, camera poses, and masks specifying regions to inpaint, our approach trains a NeRF representation of the scene. The objective is to ensure that the trained NeRF can render novel views with the masked regions consistently inpainted in 3D space. For the unmasked background, we train the NeRF with pixel-wise color and depth reconstruction loss:

$$L_{bg} = \lambda_1 \|\hat{\mathbf{x}}^{(i)} - \bar{\mathbf{x}}^{(i)}\|_2^2 + \lambda_2 \|\hat{\mathbf{x}}_d^{(i)} - \bar{\mathbf{x}}_d^{(i)}\|_2^2 \quad (4)$$

where $\bar{\mathbf{x}}^{(i)}$ and $\bar{\mathbf{x}}_d^{(i)}$ are the masked groundtruth RGB and depth map, and $\hat{\mathbf{x}}^{(i)}$ and $\hat{\mathbf{x}}_d^{(i)}$ are the rendered RGB and depth map, with λ_1, λ_2 being the corresponding weights.

For the masked area, we apply our proposed method to distill RGB prior from a diffusion model to address the occlusion problem in the color space (Eq. (8)). For geometry supervision of the NeRFs, we perform vanilla SDS for normal map prior, following MVIP-NeRF [6]:

$$\nabla_\theta L_{geo}(\theta; \mathbf{n}) = w(t)(\epsilon_\phi^\omega(\mathbf{z}_t; y, t) - \epsilon) \frac{\partial \mathbf{z}}{\partial \mathbf{n}} \frac{\partial \mathbf{n}}{\partial \theta} \quad (5)$$

where \mathbf{n} is the normal map computed from the rendered depth map. Note that both the geometry and the collaborative losses and noise predictions are computed only within the masked regions. We also explore guiding geometry with collaborative losses and report the less-satisfactory results in Supplementary Sec. 9.3. In the following subsections, we go through the design of our collaborative multi-view loss. Specifically, we employ a modified version of CDS to collectively update the NeRF parameters using information shared across a subset of views (Sec. 4.1). We further introduce reference views for collaborative knowledge distillation over more samples (Sec. 4.3). Finally, we implement a grid-based denoising strategy to enhance cross-view consistency in distillation (Sec. 4.2) and fine-tune the inpainting diffusion model for each scene to ensure visually consistent priors (Sec. 4.4).

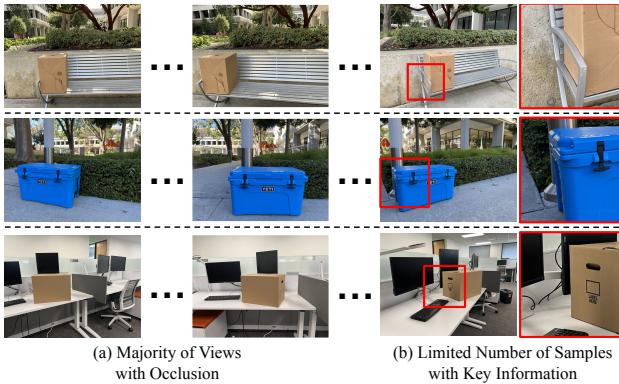


Figure 3. Illustration of the occlusion challenge in NeRF inpainting. In the training data, most views are occluded and only a few views have key information for reconstructing the occluded area. E.g., the armrest, the pole’s shape, and the monitor’s edge.

4.1. Multi-view CDS

We have witnessed how prior methods fail when the training data has limited information about the area to inpaint due to occlusion. The core of the challenge, as shown in Fig. 3, is to extract and propagate faithful information from the limited given views to the parameter updating throughout the NeRF training process.

To tackle this challenge, we aim to train the NeRF progressively with its parameters updated with consideration of multiple randomly selected training views, so that the key information can be extracted and propagated. To do this, we apply a modified version of CDS (Eq. (3)), where we adopt a radial basis function (RBF) as the kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{h} \|\mathbf{x} - \mathbf{x}'\|_2^2) \quad (6)$$

and denoise multiple views rendered from the NeRF to calculate the distillation loss, starting from:

$$\begin{aligned} \nabla_\theta L(\theta; \mathbf{x}) = & \frac{w(t)}{N} \sum_{i=1}^N \sum_{j=1}^N (\nabla_{\mathbf{z}_t^{(j)}} k(\mathbf{z}_t^{(j)}, \mathbf{z}_t^{(i)}) \\ & + k(\mathbf{z}_t^{(j)}, \mathbf{z}_t^{(i)})(\epsilon_\phi^\omega(\mathbf{z}_t^{(i)}; y, \mathbf{m}^{(i)}, t) - \epsilon)) \frac{\partial \mathbf{z}^{(i)}}{\partial \mathbf{x}^{(i)}} \frac{\partial \mathbf{x}^{(i)}}{\partial \theta} \end{aligned} \quad (7)$$

where $\mathbf{z}^{(i)}$ is the encoded latent of $\mathbf{x}^{(i)}$ by a VAE [21]. ϵ is the scheduled noise and ϵ_ϕ^ω is the predicted noise given the noised latent $\mathbf{z}_t^{(i)}$ at t . \mathbf{m}^i is the concatenation of the masks and unmasked image latent corresponding to $\mathbf{x}^{(i)}$. Meanwhile, $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are from the same set of rendered views during each update, where $\mathbf{x}^{(i)}$ is the view to back-propagate from and $\mathbf{x}^{(j)}$ ’s are the other views rendered at current iteration. We discuss how the first and the second terms spread out the influence of each view and prevent the updates from collapsing into a single mode of target distribution in Supplementary Sec. 7.

4.2. Grid-based Denoising

As many prior works have pointed out, distilling or training from individually inpainted 2D images results in inconsistent 2D appearance and artifacts in the 3D representations, usually due to the texture shift [24] in the high-frequency area or slight differences in the condition (the background to inpaint) [6, 55]. Previous SDS method [6] updates multiple views together with a sum over the distillation loss at each update, which overlooks the directional difference among the noise prediction of each view and still results in bluriness and artifacts, as shown in Fig. 4.

On the contrary, we incorporate the idea of visual prompting that has demonstrated good performance in 2D image space [1, 55] with our multi-view CDS. Specifically, at each update, instead of denoising one rendered image at a time, we randomly select multiple rendered training views and pile them into a grid of images. This grid of images is denoised as a single input, resulting in a corresponding grid of noise prediction, which is ungridded into corresponding noise predictions of each image.

Formally, we incorporate this method into Eq. (7):

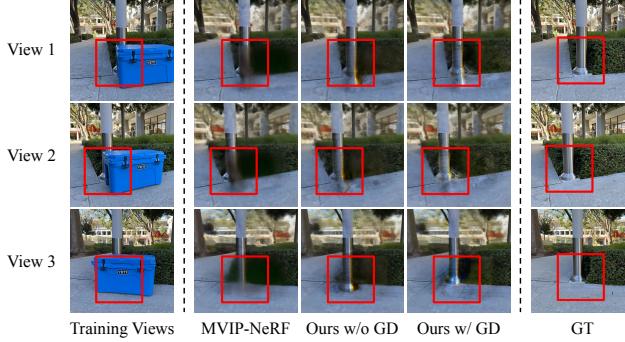


Figure 4. Illustration of the effect of normal denoising and grid-based denoising in distillation. In MVIP-NeRF, vanilla SDS denoising cannot form a consistent denoising direction for the distillation, resulting in blurriness and artifacts. With Grid-based Denoising in our method, we observe correct base locations of the pole, reduced artifacts, and clearer inpainting from multiple views.

$$\nabla_{\theta} L(\theta; \mathbf{x}) = \frac{w(t)}{N} \sum_{i=1}^N \sum_{j=1}^N (\nabla_{\mathbf{z}_t^{(j)}} k(\mathbf{z}_t^{(j)}, \mathbf{z}_t^{(i)}) + k(\mathbf{z}_t^{(j)}, \mathbf{z}_t^{(i)}) (\hat{\epsilon}_{\phi}^{\omega}(\mathbf{z}_t^{(i)}; \{\mathbf{z}_t^{(s)}\}_{s=1}^S, y, \mathbf{m}^{(\{s\}+i)} t) - \epsilon)) \frac{\partial \mathbf{z}_t^{(i)}}{\partial \theta} \quad (8)$$

where $\{\mathbf{z}_t^{(s)}\}_{s=1}^S$ is a subset from the current rendered views randomly selected to form a grid with $\mathbf{z}^{(i)}$ and $\mathbf{m}^{(\{s\}+i)}$ is the set of concatenations of corresponding masks and masked image latent. $\hat{\epsilon}_{\phi}^{\omega}$ is the proposed grid-based noise prediction, which is sequentially composed of, (1) a grid operation, (2) a regular noise prediction, and (3) an ungrid operation, to obtain noise predictions corresponding to the input views. For N rendered views, we shuffle them, perform grid-based denoising for M times, and take the average over the M noise predictions for each view. Let G and G^{-1} be the grid and ungrid operation, respectively:

$$G^{-1}(\hat{\epsilon}_{\phi}^{\omega}(G(\mathbf{z}_t^{(i)}, \{\mathbf{z}_t^{(s)}\}_{s=1}^S); y, G(\mathbf{m}^{(\{k\}+i)}), t)) = \{\bar{\epsilon}^{(i)}, \{\bar{\epsilon}^{(s)}\}_{s=1}^S\} \quad (9)$$

where $\{\bar{\epsilon}^{(i)}, \{\bar{\epsilon}^{(s)}\}_{s=1}^S\}$ are the predicted noises of $\mathbf{z}_t^{(i)}$ and $\{\mathbf{z}_t^{(s)}\}_{s=1}^S$ respectively. In short, grid-based denoising treats a grid of images as one, denoises them together, and predicts the noises correspondingly, resulting in consistent denoising directions within the set, as shown in Fig. 4.

To this end, we obtain the grid-based noise prediction w.r.t. the i^{th} rendered view:

$$\hat{\epsilon}_{\phi}^{\omega}(\mathbf{z}_t^{(i)}; \{\mathbf{z}_t^{(s)}\}, y, \mathbf{m}^{(\{s\}+i)} t) = \bar{\epsilon}^{(i)} \quad (10)$$

4.3. Reference Views Updating

So far, we have introduced two major techniques in our method that address knowledge sharing across multiple views to enhance occluded reconstruction and cross-view

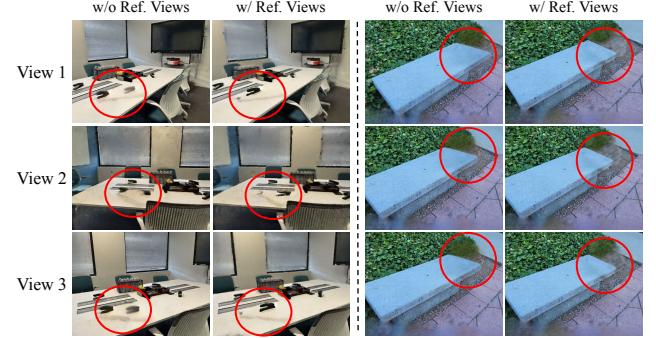


Figure 5. With Reference Views applied in the training process (bottom row), the NeRF learns to render the masked region with more information globally from the data, as compared to incorrect inpainting while rendering a smaller set of training views only (e.g. the hallucination of the stapler in the left scene or the extension of the bench’s corner in the right scene).

consistency. As shown in Eq. (8), both techniques distill knowledge from multiple rendered views to the NeRF. To maximize this knowledge distillation across multiple views, we propose a training paradigm with Reference Views. Specifically, during each iteration, we rendered a set of training views, V_{train} , to backpropagate the loss to train the NeRF and another set of reference views, V_{ref} , the gradient of which will not be calculated, to provide extra information in both CDS and Grid-denoising. In Eq. (8), the kernel will be calculated within V_{train} , i.e., $\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in V_{train}$. Meanwhile, grids will be formed by images from $V_{train} \cup V_{ref}$, i.e. $\{\mathbf{z}_t^{(s)}\}_{s=1}^S \subset (V_{train} \cup V_{ref})$.

We illustrate how training with Reference Views enhances the consistency and information sharing qualitatively in Fig. 5 and quantitatively in our ablation studies.

4.4. Per-scene Fine-tuning

As pointed out by prior works [24], applying latent diffusion models to real-world content such as NeRF is usually challenged from converging to a crisp and deterministic geometry and texture, due to the high diversity of synthetic content from diffusion models. In pursuit of a more realistic and blending-in convergence of the inpainting, we fine-tune the diffusion model for each scene individually. Specifically, we choose a fixed text token for each scene and apply LoRA [17] to fine-tune the U-Net of the diffusion model. Each training sample is generated by randomly masking a view from the training views with one rectangle or one circle ($p = 0.5$ each), with the masked region being the ground truth. We utilize prior-preservation loss [41] and MSE to supervise the learning. To avoid the model learning patterns from the area we aim to inpaint in NeRF, we mask the training views with the original masks for NeRF training and set the loss within to zero during fine-tuning.

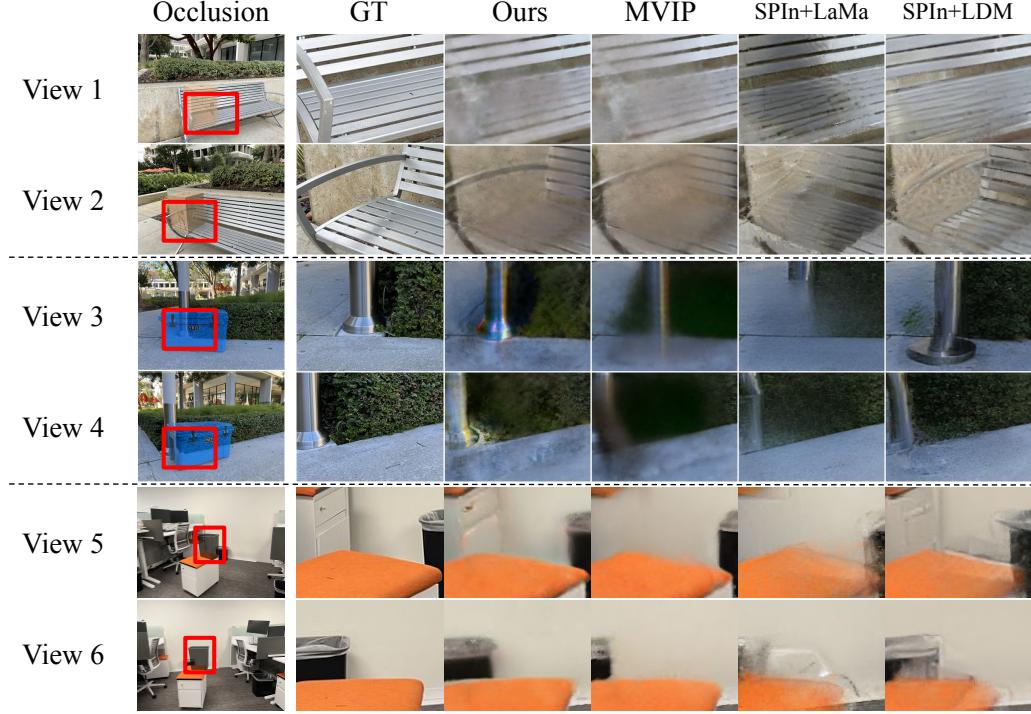


Figure 6. Visualization of qualitative results. For scenes that do not have much occlusion (View 1&2, the armrest is visible in most views), our method and the baseline methods can render similarly good novel views in terms of quality and fidelity. When challenged with severe occlusion (View 3-6), our method excels. Specifically, while baseline methods might render correctly in the views where occlusion is not severe, they cannot propagate the information to other views. For example, MVIP can correctly render the base of the pole in View 4, but it cannot render a consistent base in View 3. During distillation, such inconsistency is propagated and results in artifacts and blurriness in both views. Similar examples can be found in View 5&6 (incorrect positions of the trash bin). One limitation is that like prior methods, our method cannot remove the shadow.

	SPIIn-NeRF					Occlude-NeRF				
	PSNR \uparrow	LPIPS \downarrow	L2 \downarrow	SSIM \uparrow	Corrs. \uparrow	PSNR \uparrow	LPIPS \downarrow	L2 \downarrow	SSIM \uparrow	Corrs. \uparrow
SPIIn-NeRF+LaMa	15.992	0.284	0.287	0.416	73.080	14.154	0.321	0.369	0.619	37.483
SPIIn-NeRF+LDM	16.162	0.298	0.286	0.408	79.640	14.107	0.284	0.338	0.615	47.442
MVIP-NeRF	16.080	0.404	0.302	0.449	76.515	13.872	0.387	0.345	0.601	37.023
Occlude-NeRF	16.177	0.355	0.290	0.461	86.715	15.447	0.346	0.314	0.680	50.928

Table 1. Evaluation results for different methods on the SPIIn-NeRF and Occlude-NeRF datasets.

5. Experiments

5.1. Experiment Setup

5.1.1. Baseline Methods

We chose SPIIn-NeRF [35] with LaMa [40], SPIIn-NeRF with Latent Diffusion Model (LDM), and MVIP-NeRF [6] as our baseline methods for the 3D NeRF inpainting task. The first two baseline methods are representatives of NeRF inpainting with 2D inpainted images and MVIP-NeRF serves as our baseline for SDS-based NeRF inpainting.

5.1.2. Datasets

We chose the SPIIn-NeRF dataset and a subset of the LLFF [32] dataset annotated (with masks and pseudo-depth maps) by SPIIn-NeRF [35] for our experiments, to assess the algorithms’ performance in general cases where occlusion occurs regularly. Additionally, we collected a novel dataset for NeRF inpainting, named Occlude-NeRF. This dataset was deliberately crafted with severe occlusion in the scene to provide a rigorous test of the algorithms’ performance. The detailed procedure for creating this dataset is provided in Supplementary Sec. 8. In the SPIIn-NeRF dataset, there are 10 scenes each with 60 training views and 40 testing

	SPIn-NeRF					Occlude-NeRF				
	PSNR \uparrow	LPIPS \downarrow	L2 \downarrow	SSIM \uparrow	Corrs. \uparrow	PSNR \uparrow	LPIPS \downarrow	L2 \downarrow	SSIM \uparrow	Corrs. \uparrow
(i) Ours w/o CDS	14.802	0.389	0.329	0.383	85.517	14.841	0.415	0.315	0.615	45.413
(ii) Ours w/o G.D.	15.684	0.367	0.297	0.449	81.498	14.774	0.361	0.324	0.628	42.888
(iii) Ours w/o Ref.	16.037	0.372	0.318	0.453	82.996	14.867	0.356	0.327	0.620	45.698
(iv) Ours w/o F.T.	15.562	0.358	0.305	0.449	83.844	13.661	0.397	0.368	0.602	46.412
(v) Ours (full)	16.177	0.355	0.290	0.461	86.715	15.447	0.346	0.314	0.680	50.928

Table 2. Evaluation results for different ablations of our methods on the SPIn-NeRF and Occlude-NeRF datasets.

views with their corresponding masks of an object to remove. The LLFF dataset contains varying numbers of images in 5 scenes. The Occlude-NeRF dataset contains 6 scenes, each with 60 training views and 40 testing views.

5.1.3. Implementation Details

We implement all baseline methods with hyperparameters reported in their corresponding papers with a few exceptions. For SPIn-NeRF+LDM, we chose Stable Diffusion 2 (SD 2) Inpainting [40] as the 2D LDM inpainter. For MVIP-NeRF, we chose SD 2 Inpainting as the diffusion priors, for fair comparison. In the implementation of our methods, we chose SD 2 Inpainting as the diffusion priors and set the Grid size to 2×2 for computation efficiency. For the Reference-View training with the SPIn-NeRF and the Occlude-NeRF dataset, we randomly select 12 training views ($|V_{train}| = 12$) and 48 reference views ($|V_{ref} = 48|$). For the LLFF dataset, we set ($|V_{train}| = 4$) and ($|V_{ref} = 16|$) due to the smaller sample sizes. More detailed hyperparameters are listed in Supplementary Sec. 9.

5.2. Results

5.2.1. Metrics

In our experiments, we aim to assess the following attributes: **Quality and Fidelity**: The visual characteristics and realism of the images. **Faithfulness**: How accurately the rendered image corresponds to the original scene.

Cross-view Consistency: How the rendered views remain consistent across different viewpoints

We follow similar prior works [6, 24, 35] and evaluate the quality with PSNR and fidelity with LPIPS [60]. We apply L2 pixel-wise error and SSIM to assess the faithfulness of the synthesized views. We evaluate the cross-view consistency of the rendered views with a Correspondence Score (Corrs.), similar to [55], where we report the numbers of high-quality LoFTR [50] correspondences identified in 100 randomly sampled pairs of rendered images and their ground truths. To balance the randomness, we proceed for 20 iterations and take the averages. Note that, unlike some prior work, we do not evaluate with FID [14] or KID [3], because the NeRF datasets are relatively small with insufficient data points for stable calculation for such

metrics [2, 23, 29]. We only evaluate qualitatively on the LLFF dataset due to the absence of removal ground truth.

5.2.2. V.S. Baselines

We conduct quantitative evaluations to compare the efficacy of our method v.s. that of three baselines in 3D NeRF inpainting tasks. Specifically, we focus on the quality of the synthesized views and their faithfulness in reconstructing the original scenes. The results are reported in Tab. 1. Further visual comparisons are illustrated in Fig. 6.

From the quantitative results, we observe that our method achieves the highest SSIM and Corrs. on both datasets, demonstrating a stronger ability of structural preservation, and cross-view consistency. Both SPIn-NeRF methods yield lower LPIPS scores, which can be attributed to the minimization of LPIPS distance between the inpainting and rendering in their methods. We also see that on the SPIn-NeRF dataset, where there is not much occlusion, our method yields a close but slightly more L2 distance. While on the Occlude-NeRF dataset, where occlusion is severe in the scenes, our method outperforms all baseline methods in L2, SSIM, and Corrs., which indicates better consistency and faithfulness. Such results demonstrate that our method excels in tackling severely occluded 3D inpainting tasks while preserving satisfactory performance in image quality and fidelity but compromising a bit of perceptual coherency. More qualitative discussions can be found in Supplementary Sec. 10.

5.2.3. Ablation Studies

We conduct ablation studies on our method to investigate the effect of each of our modules. We test the ablations on the SPIn-NeRF dataset and the Occlude-NeRF dataset. Specifically, we start with the full method and ablate (i) CDS, (ii) Grid-based Denoising, (iii) Reference Views, and (iv) Per-scene Fine-tuning, respectively. The quantitative results are reported in Tab. 2.

Particularly, comparing ours with (i), we identify a significant drop in L2 and SSIM performance without CDS. Additionally, we see close Corrs. scores on the SPIn-NeRF dataset but a tremendous improvement of Corrs. scores on the Occlude-NeRF dataset. This affirms the efficacy of CDS in faithful reconstructions across the views, especially in

occlusion cases. Showing by the differences between (v) and (iii), the use of Reference views further enhances the efficacy of CDS in faithfulness with slightly improved image quality and fidelity. A significant drop in Corrs. (ii) denoising demonstrates that Grid-based denoising greatly contributes to the consistency among the views. As also suggested by the drop in PSNR and LPIPS scores of (iv) on both datasets, we conclude that per-scene fine-tuning enhances the overall quality of the view synthesis.

6. Conclusion

We introduce Occlude-NeRF, a novel approach to inpaint a 3D NeRF scene. Our method tackles the occlusion challenge unaddressed by prior work, where most views are occluded leaving a limited amount of information about the area to inpaint. Specifically, we propose a multi-view version of CDS, a Grid-based denoising pattern, and a Reference-view training paradigm to enable information sharing among the views. We also apply per-scene fine-tuning to enhance the rendering quality and fidelity. To assess Occlude-NeRF’s performance in occlusion cases, we construct a novel dataset with challenging occlusion. The major limitation of our methods lies in the reconstruction of high-frequency regions due to the averaging of multiple views during distillation, which we further showcase in Supplementary Sec. 10.1.

References

- [1] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. [2](#), [5](#)
- [2] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. [8](#)
- [3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. [8](#)
- [4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voeleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. [6](#)
- [5] Chenjie Cao and Yanwei Fu. Learning a sketch tensor space for image inpainting of man-made scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14509–14518, 2021. [3](#)
- [6] Honghua Chen, Chen Change Loy, and Xingang Pan. Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2024. [2](#), [4](#), [5](#), [7](#), [8](#)
- [7] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. [3](#)
- [8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. [3](#)
- [9] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, pages 1033–1038. IEEE, 1999. [3](#)
- [10] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. [3](#), [6](#)
- [11] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. [2](#)
- [12] Fengming He, Xiyun Hu, Jingyu Shi, Xun Qian, Tianyi Wang, and Karthik Ramani. Ubi edge: authoring edge-based opportunistic tangible user interfaces in augmented reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023. [9](#)
- [13] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2328–2337, 2023. [3](#)
- [14] 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. *Advances in neural information processing systems*, 30, 2017. [8](#)
- [15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. [4](#)
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [3](#)
- [17] 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. *arXiv preprint arXiv:2106.09685*, 2021. [6](#)
- [18] Rahul Jain, Jingyu Shi, Runlin Duan, Zhenghe Zhu, Xun Qian, and Karthik Ramani. Ubi-touch: Ubiquitous tangible object utilization through consistent hand-object interaction in augmented reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18, 2023. [9](#)
- [19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. [3](#)
- [20] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative score distillation for consistent visual synthesis. *arXiv preprint arXiv:2307.04787*, 2023. [2](#), [3](#), [4](#)
- [21] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [5](#)

- [22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 4
- [23] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017. 8
- [24] Chieh Hubert Lin, Changil Kim, Jia-Bin Huang, Qinbo Li, Chih-Yao Ma, Johannes Kopf, Ming-Hsuan Yang, and Hung-Yu Tseng. Taming latent diffusion model for neural radiance field inpainting. In *European Conference on Computer Vision*, pages 149–165. Springer, 2025. 2, 3, 5, 6, 8
- [25] Hao-Kang Liu, I Shen, Bing-Yu Chen, et al. Nerf-in: Free-form nerf inpainting with rgb-d priors. *arXiv preprint arXiv:2206.04901*, 2022. 3
- [26] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016. 3, 4
- [27] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5773–5783, 2021. 2
- [28] Yi Liu, Xinyi Li, and Wenjing Shuai. 3d scene de-occlusion in neural radiance fields: A framework for obstacle removal and realistic inpainting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 10144–10153, 2024. 3
- [29] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018. 8
- [30] Dizhi Ma, Xiyun Hu, Jingyu Shi, Mayank Patel, Rahul Jain, Ziyi Liu, Zhengze Zhu, and Karthik Ramani. avattar: Table tennis stroke training with embodied and detached visualization in augmented reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16, 2024. 9
- [31] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 3
- [32] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 2, 7
- [33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [34] Ashkan Mirzaei, Yash Kant, Jonathan Kelly, and Igor Gilitschenski. Laterf: Label and text driven object radiance fields. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 2
- [35] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinstein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 2, 3, 7, 8, 4
- [36] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 3
- [37] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018. 4
- [38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 3
- [39] Kira Prabhu, Jane Wu, Lynn Tsai, Peter Hedman, Dan B Goldman, Ben Poole, and Michael Broxton. Inpaint3d: 3d scene content generation using 2d inpainting diffusion. *arXiv preprint arXiv:2312.03869*, 2023. 2, 3
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 3, 7, 8, 6
- [41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 6
- [42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4
- [43] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 4
- [44] Jingyu Shi, Rahul Jain, Hyungjun Doh, Ryo Suzuki, and Karthik Ramani. An hci-centric survey and taxonomy of human-generative-ai interactions. *arXiv preprint arXiv:2310.07127*, 2023. 9
- [45] Jingyu Shi, Rahul Jain, Runlin Duan, and Karthik Ramani. Understanding generative ai in art: An interview study with artists on g-ai from an hci perspective. *arXiv preprint arXiv:2310.13149*, 2023. 9
- [46] Jingyu Shi, Rahul Jain, Seungguen Chi, Hyungjun Doh, Hyunggun Chi, Alexander J Quinn, and Karthik Ramani. Caring-ai: Towards authoring context-aware augmented reality instruction through generative artificial intelligence. *arXiv preprint arXiv:2501.16557*, 2025. 9

- [47] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 6
- [48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3
- [49] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3
- [50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 8
- [51] 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 *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 2, 3
- [52] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 2
- [53] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süssstrunk. Innerf360: Text-guided 3d-consistent object inpainting on 360-degree neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2024. 2
- [54] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [55] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20731–20741, 2024. 2, 3, 5, 8, 4
- [56] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023. 2, 3
- [57] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 2
- [58] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 3
- [59] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2
- [60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8
- [61] Xingchen Zhou, Ying He, F Richard Yu, Jianqiang Li, and You Li. Repaint-nerf: Nerf editing via semantic masks and diffusion models. *arXiv preprint arXiv:2306.05668*, 2023. 3
- [62] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12588–12597, 2023. 3
- [63] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20722–20731, 2023. 3
- [64] Junzhe Zhu, Peiyi Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023. 5
- [65] Zixin Zou, Weihao Cheng, Yan-Pei Cao, Shi-Sheng Huang, Ying Shan, and Song-Hai Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7900–7908, 2024. 3

OccludeNeRF: Geometric-aware 3D Scene Inpainting with Collaborative Score Distillation in NeRF

Supplementary Material

7. Analysis on Multi-view CDS

We have proposed the core distillation sampling strategy in our main manuscript, namely Equation (8). In this Supplementary section, we discuss how Equation (8) helps propagate the information from limited views to the collaborative update of the NeRF parameters.

7.1. Kernel-Weighted Noise Prediction

7.1.1. Noise Prediction

For each view i , the noise prediction $\hat{\epsilon}^{(i)}$ is computed as a kernel-weighted combination of noise predictions from all views:

$$\hat{\epsilon}^{(i)} = \frac{1}{N} \sum_{j=1}^N k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \epsilon^{(j)} \quad (11)$$

where:

- $\epsilon^{(j)}$ is the noise prediction for view j .
- $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ is the kernel function, which measures similarity between the latents $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$.

The kernel $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ ensures that:

- Views j are from the rendered set as view i . Those from views j with relevant information (e.g., visible occluded areas) contribute more strongly to the noise prediction $\hat{\epsilon}^{(i)}$ of view i .
- Occluded view i to incorporate details from views j where the occluded area is visible.

7.2. Loss Function and Gradient

7.2.1. Loss Function

The loss for each view i is defined as:

$$L^{(i)} = \hat{\epsilon}^{(i)} - \epsilon_{\text{gt}}^{(i)} \quad (12)$$

where $\epsilon_{\text{gt}}^{(i)}$ is the ground-truth noise for view i . Substituting $\hat{\epsilon}^{(i)}$, the loss becomes:

$$L^{(i)} = \frac{1}{N} \sum_{j=1}^N k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \epsilon^{(j)} - \epsilon_{\text{gt}}^{(i)} \quad (13)$$

7.2.2. Gradient of the Loss

The gradient of this loss with respect to the latent $\mathbf{z}^{(i)}$ is:

$$\nabla_{\mathbf{z}^{(i)}} L^{(i)} = \frac{1}{N} \sum_{j=1}^N \nabla_{\mathbf{z}^{(i)}} \left(k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \right) \epsilon^{(j)} \quad (14)$$

For a Gaussian RBF kernel with scale h :

$$k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \exp \left(-\frac{1}{h} \|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|_2^2 \right) \quad (15)$$

the gradient is:

$$\nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = -\frac{2}{h} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)}) k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \quad (16)$$

Substituting this into $\nabla_{\mathbf{z}^{(i)}} L^{(i)}$, we get:

$$\nabla_{\mathbf{z}^{(i)}} L^{(i)} = \frac{1}{N} \sum_{j=1}^N \left(-\frac{2}{h} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)}) k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \right) \epsilon^{(j)} \quad (17)$$

This gradient shows how information propagates between views i and j , with the kernel $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ modulating the strength of interaction.

7.3. NeRF Parameter Updates

7.3.1. Gradient Propagation

The latent updates are propagated to the NeRF parameters θ through backpropagation. The total gradient for θ is:

$$\nabla_{\theta} L = \sum_{i=1}^N \nabla_{\theta} L^{(i)} \quad (18)$$

Using the chain rule:

$$\nabla_{\theta} L^{(i)} = \nabla_{\mathbf{z}^{(i)}} L^{(i)} \cdot \nabla_{\theta} \mathbf{z}^{(i)} \quad (19)$$

Substituting $\nabla_{\mathbf{z}^{(i)}} L^{(i)}$ from above, we get:

$$\begin{aligned} \nabla_{\theta} L &= \sum_{i=1}^N \nabla_{\theta} \mathbf{z}^{(i)} \cdot \\ &\left(\frac{1}{N} \sum_{j=1}^N \left(-\frac{2}{h} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)}) k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \right) \epsilon^{(j)} \right) \end{aligned} \quad (20)$$

7.4. Role of the Two Terms in the Kernel Update

The kernel update involves two key terms:

$$\nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = -\frac{2}{h} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)}) k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \quad (21)$$

and:

$$k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}). \quad (22)$$

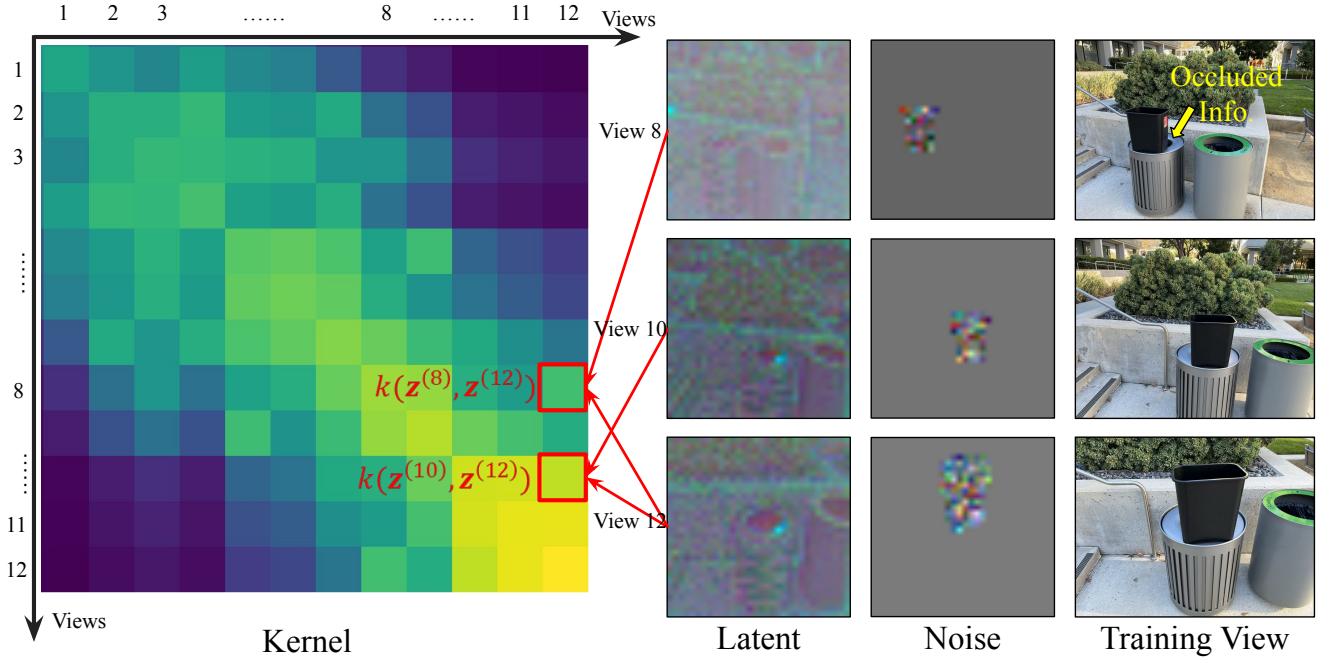


Figure 7. Illustration of the effect of our CDS kernel in one iteration. The heatmap of the kernel is on the left. The warmer the color, the higher the kernel value. We can see the corresponding views on the right-hand side. The closer (in the latent space) the views are, the higher the correspondence value in the kernel (View 10 & 12). Meanwhile, views with further distance but containing important information can also be related to the update (e.g. the kernel has a relatively high value between (View 8 & 10). In this way, the information about the occluded area (e.g. View 8, where the hole of the left trash bin is visible) can be propagated to the update of other views (e.g. View 12, where the hole is completely occluded).

First Term: Gradient of the Kernel ($\nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$)

This term serves several critical purposes in the kernel-based updates:

1. Repulsive Force to Maintain Diversity:

$$\nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = -\frac{2}{h} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)}) k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \quad (23)$$

The term $(\mathbf{z}^{(i)} - \mathbf{z}^{(j)})$ computes the directional vector pointing from $\mathbf{z}^{(j)}$ to $\mathbf{z}^{(i)}$. The negative sign ensures that the gradient drives $\mathbf{z}^{(i)}$ away from $\mathbf{z}^{(j)}$, creating a repulsive effect between similar latents. This repulsion prevents all latents from collapsing into a single representation, ensuring sufficient diversity among the latent representations for different views.

2. Propagation of Occlusion Information: When a view j contains visible information about an occluded area, its latent $\mathbf{z}^{(j)}$ contributes gradients to the update of $\mathbf{z}^{(i)}$ through this term:

$$\nabla_{\mathbf{z}^{(i)}} L^{(i)} = \frac{1}{N} \sum_{j=1}^N \nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \epsilon^{(j)}. \quad (24)$$

If $\mathbf{z}^{(j)}$ is close to $\mathbf{z}^{(i)}$, the kernel $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ will be large, amplifying the influence of $\mathbf{z}^{(j)}$ on $\mathbf{z}^{(i)}$. This ensures that visible details in $\mathbf{z}^{(j)}$ are propagated into the occluded representation $\mathbf{z}^{(i)}$.

3. Modulation by Kernel Weight: The term $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ modulates the strength of the gradient, ensuring that only nearby latents significantly influence $\mathbf{z}^{(i)}$. Mathematically, the magnitude of the gradient is proportional to the similarity between $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$, as measured by $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$.

Second Term: Kernel Weight ($k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$)

This term determines how much influence view j has on view i in the kernel-weighted noise prediction:

$$\hat{\epsilon}^{(i)} = \frac{1}{N} \sum_{j=1}^N k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \epsilon^{(j)}. \quad (25)$$

1. Weighted Contribution of Nearby Views: The kernel $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ assigns higher weights to views j with similar latents $\mathbf{z}^{(j)}$ to $\mathbf{z}^{(i)}$, ensuring that these views have a stronger influence on the noise prediction for view i . This is

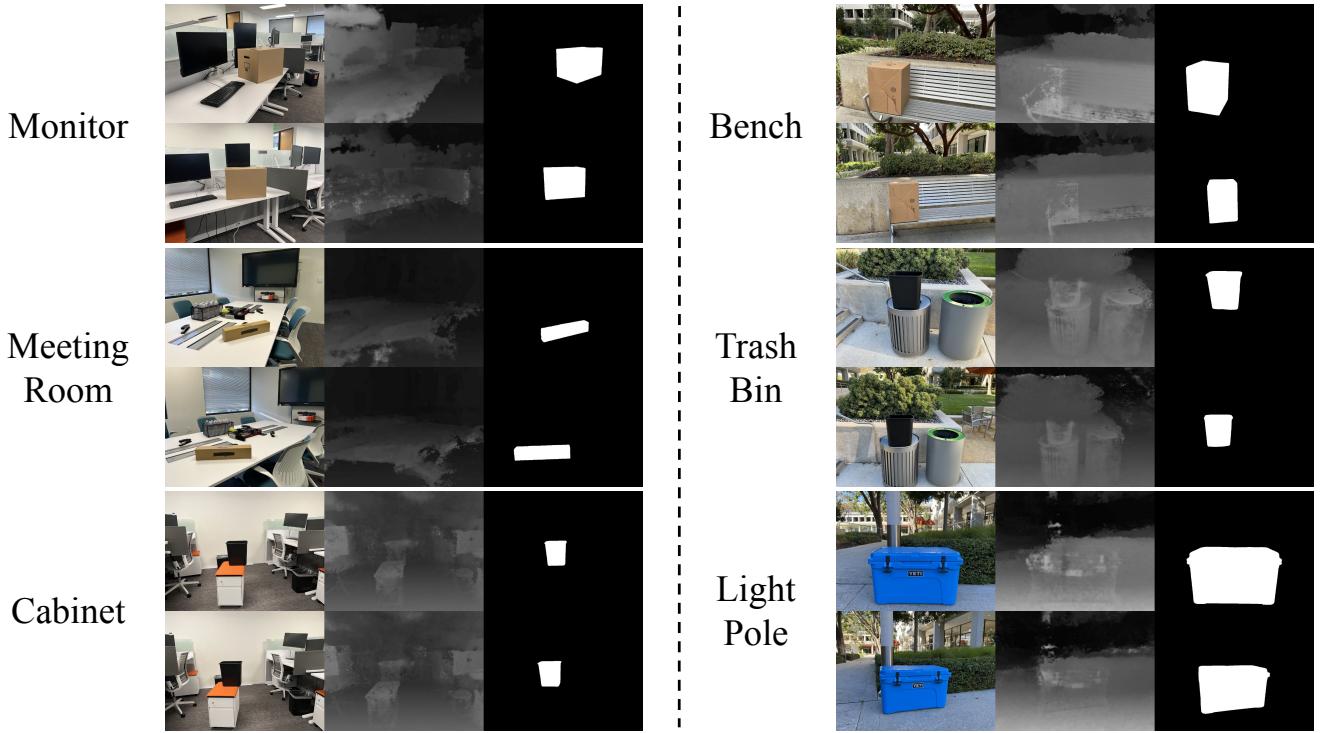


Figure 8. Visualization of our customized dataset. For each scene, we set up an obstacle blocking an area and mask the obstacle for inpainting tasks. We collect testing views and training views with RGB images and masks. We also generate the pseudo-depth maps (visualized as the disparity maps) corresponding to each view.

particularly critical when $\mathbf{z}^{(j)}$ contains visible information about an occluded area in $\mathbf{z}^{(i)}$, as the kernel amplifies the contribution of $\mathbf{z}^{(j)}$.

2. Occlusion-Aware Updates: For occluded areas, $\mathbf{z}^{(i)}$ from views where the occlusion is visible dominates the noise prediction for $\mathbf{z}^{(i)}$, effectively propagating information about the occluded area across views:

$$\hat{\epsilon}^{(i)} \approx \frac{\sum_j^N k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \epsilon^{(j)}}{\sum_j^N k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})}. \quad (26)$$

This weighted update ensures that the occluded representation $\mathbf{z}^{(i)}$ aligns with the visible views.

3. Locality of Influence: The kernel decays rapidly with distance in the latent space:

$$k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \exp\left(-\frac{1}{h}\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|_2^2\right). \quad (27)$$

As $\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|$ increases, $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \rightarrow 0$, ensuring that only nearby latents significantly influence the updates. In this way, we avoid distillation of 2D prior from views that are too far away, which may result in inconsistent 2D inpainting results, as also pointed out by prior work [55].

Combined Functionality of the Two Terms

The combined effect of the two terms is as follows:

- The **first term** ($\nabla_{\mathbf{z}^{(i)}} k$) ensures that information propagates between views and prevents collapse by introducing repulsive forces.
- The **second term** (k) amplifies contributions from relevant views, particularly those with visible occluded areas, ensuring effective information sharing.

Together, these terms propagate occlusion information across views, align latent representations, and maintain diversity in the latent space, enabling robust NeRF training.

Why Occlusion Information Propagates to θ

1. Kernel-Based Weighting: The kernel $k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ ensures that visible views j contribute more strongly to occluded views i , propagating occlusion details across the latent space, as shown in Figure 7

2. Collaborative Updates to Latents: The gradient $\nabla_{\mathbf{z}^{(i)}} k(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ drives latent $\mathbf{z}^{(i)}$ of occluded views to align with $\mathbf{z}^{(j)}$ of visible views.

3. Backpropagation to NeRF: The updated latents $\mathbf{z}^{(i)}$ are used to refine the NeRF parameters θ , enabling the model to represent occluded areas consistently across all views.

8. Dataset Building

In this section, we introduce our procedure and details for constructing the Occlude-NeRF dataset.

8.1. Scene Setup

We collected data from six scenes in total, with three indoors and three outdoors, as shown in Figure 8. We name the three indoor scenes: *Cabinet*, *Monitor*, and *Meeting Room*, and the three outdoor scenes: *Bench*, *Trash Bin*, and *Light Pole*. The specific scene description is listed below:

- *Cabinet*: an office workspace with a symmetrical arrangement of two cubicles on either side. Each cubicle includes a white desk, a chair with a gray backrest, and an orange seat cushion. A black trash bin is placed on top of a mobile pedestal with an orange cushion. The black trash bin is masked.
- *Monitor*: an office workspace with a white desk and a cardboard box placed in the center. On the desk are two Dell monitors (one visible and turned off), a black keyboard, and a desk lamp on the left. The cardboard box is masked.
- *Meeting Room*: an indoor office meeting room with a white conference table surrounded by teal office chairs. On the table are various items, including a long cardboard box, staplers, and a black organizer containing stationery such as pens, highlighters, and sticky notes. The cardboard box is masked.
- *Bench*: a cardboard box placed on a silver metal bench in an outdoor area. The bench is positioned next to a concrete planter filled with green shrubs and small rocks. The cardboard box is masked.
- *Trash Bin*: two outdoor trash bins in front of a concrete planter with green shrubs. The left bin is metallic with vertical slits, and a black rectangular container is placed on its circular opening. The right bin is smooth, gray, and labeled "Compost" with a green rim. The black container is masked.
- *Light Pole*: a bright blue cooler placed on a concrete sidewalk, next to a metal pole and a neatly trimmed hedge. The cooler is masked.

8.2. Data Collection

For each scene, we collect 60 training views and 40 testing views. For each training view, we obtain a mask by prompting a point at the object to mask, using the Segment Anything Model (SAM) [22]. Each mask is dilated with a 3×3 kernel for 3 iterations. To obtain relatively accurate camera pose estimations, we mark the objects' location in the scene with a marker, place the object to take one image, and remove the object for another while the camera remains static. In this way, we obtain a testing view with and without the object in the scene. We then put the object back according to the mark and move the camera for other

views. We conducted this procedure because we found prior work's [35] method for estimating camera poses resulted in unstable accuracy since they use COLMAP [42, 43] to perform structure from motion with images with and without objects. Therefore, in our case, we obtain extra images with objects for the testing views, so that we can estimate the poses for both training views and testing views together. We then obtain the pseudo depth maps for each view following SPIn-NeRF [35]. The collected dataset samples can be found in Figure 8.

9. Hyperparameter Details

In this section, we elucidate the hyperparameters we have used in our experiments, as well as some findings exploring the hyperparameters.

9.1. Implementation

We implement our Occlude-NeRF method on 2 NVIDIA H100 GPUs, trained for 10,000 iterations for each scene with the Adam optimizer with a learning rate of $1e - 4$ scheduled with a cosine annealing scheduler (max number of iterations: 50 and min learning rate: 0). For the distillation sampling, we follow prior work [6, 55] and choose timesteps uniformly increasing with the training from $t_{min} = 0.02$ to $t_{max} = 0.98$. For the classifier-free guidance [15], we choose a uniform value for all scenes to see the generalizability of our methods, in contrast to MVIP-NeRF [6]. Specifically, we set:

$$\hat{\epsilon} = \epsilon_{uncond} + \gamma \times (\epsilon_{text} - \epsilon_{uncond}) \quad (28)$$

where $\hat{\epsilon}$ is the final noise prediction, ϵ_{uncond} and ϵ_{text} are the noise prediction with no condition and conditioned by text, respectively, and γ is the guidance scale, which we set uniformly $\gamma = 7.5$. During training, the size of latent z is set to 256×256 for collaborative distillation and 512×512 for geometry distillation due to GPU RAM limits. For each iteration, we set the batch (of rays) size to 1024 and the number of samples along the ray to 32. Additionally, we set the number of samples for fine networks to 32. We test with three different numbers of the Grid-based Denoising $M = 1, 4, 8$ and choose $M = 4$ to balance performance and training time. The textual prompts to diffusion models are listed in Table 3:

9.2. Exploratory Study on Hyperparameters

In addition to the hyperparameter choices we have reported above, we explore the hyperparameter space and report interesting findings in this subsection.

9.2.1. Noise Scheduling

We test two noise scheduling methods. Namely, we first implemented a random sampling schedule, where a random



Figure 9. Comparison between (a) random timestep scheduling and (b) progressive timestep scheduling. We can easily observe the better convergence of shape and appearance of the latter.

noise timestep between t_{min} and t_{max} is chosen. We then implemented a progressive sampling schedule similar to [6, 64]:

$$t = t_{max} - (t_{max} - t_{min}) * iter/max_iter \quad (29)$$

where $iter$ is the current iteration number and max_iter is the total number of iterations.

Qualitatively, we found that the progressive sampling schedule fosters convergence toward clearer and sharper inpainting, as shown in Figure 9. This can be attributed to the larger changes in the earlier stages to form the 3D representations and smaller changes in the later stages, instead of randomly changing the update scale mid-training, which aligns with the similar findings in [55].

9.2.2. Randomization during Grid-based Denoising

We experimented with different numbers of times shuffling during Grid-based Denoising, namely $M = 1, 4, 8$. Our experiments qualitatively showed that 8 times of shuffling yields fewer artifacts in the inpainted area, followed by $M = 4$, and then $M = 1$, as shown in Figure 10. This can be attributed to the averaging effect of the shuffling step in our pipeline, where the influence of multiple views is merged into one update of distillation. However, increasing M by one means calling the U-Net for one iteration of denoising, which severely increases the training time. Therefore, we made a trade-off between training efficiency and performance by setting $M = 4$.

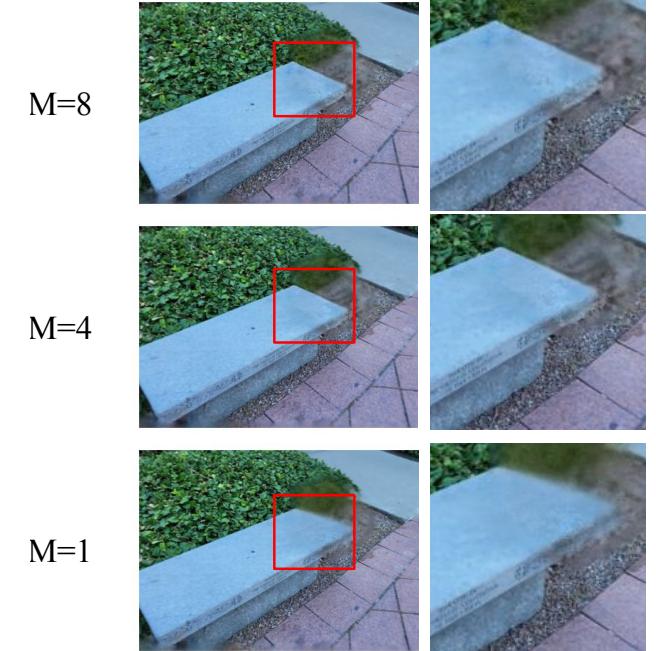


Figure 10. Comparison among different values of M . While $M = 8$ yields the best visual results regarding the sharpness of the inpainting. We made a trade-off between the computation cost in the training phases and the performance and eventually chose $M = 4$ for our experiment.

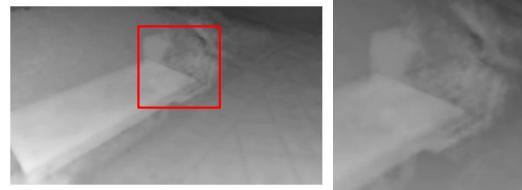
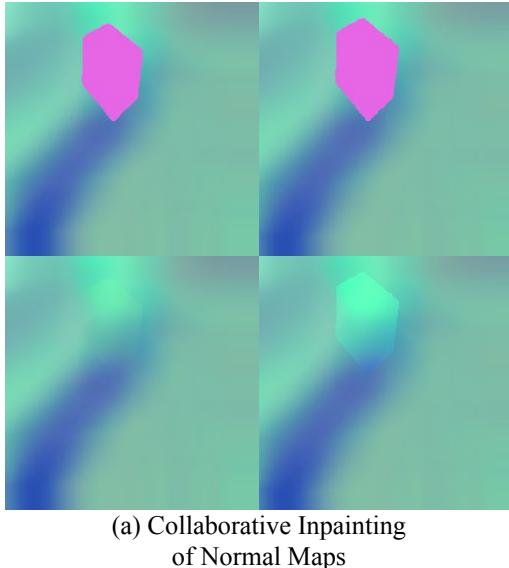
Dataset Name	Scenes	Prompts
SPIn-NeRF	1	”a stone park bench”
	2	”a wooden tree trunk on dirt”
	3	”a red fence”
	4	”stone stairs”
	7	”a grass ground”
	9	”a corner of a brick wall and a carpeted floor”
	10	”a wooden bench in front of a white fence”
	12	”grass ground”
	Book	”a brick wall with an iron pipe”
	Trash	”a brick wall”
Occlude-NeRF	Monitor	”a computer monitor on a white office desk”
	Meeting Room	”a black stapler on a white office table”
	Bench	”a silver metallic bench with slats and armrests”
	Trash Bin	”a metallic garbage bin with a round opening”
	Light Pole	”a metallic light pole next to a green ground plant”
	Cabinet	”a white filing cabinet with an orange cushion on top, next to a white wall”
LLFF	Fern	”plant and planter on dirt”
	Fortress	”a wooden tabletop”
	Horns	”glass windows and a white support pillar on carpet”
	Orchids	”a conference room with black office chairs and brown carpet”
	Room	”green leaves of a plant”

Table 3. Textual prompts used for each scene in our experiments.

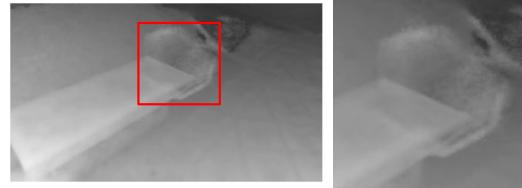
9.2.3. Qualitative Comparison with 3D-attention-based SDS

Inspired by prior works on 3D-attention in Diffusion Models [4, 10, 47], we attempted 3D-attention in SDS in our early trials. Specifically, we replaced the diffusion model we used with Stable Video Diffusion (SVD) [4] and MV-Dream (MVD) [47]. For SVD, we trained the NeRF with SD2 for 3000 iterations to get the initial reconstruction of the scene and then switched to SVD. During each iteration, we rendered 14 views with the first one used as the reference and passed then 14 views as a batch to SVD and back-propagated the distillation loss. For MVD, we disabled the mask condition and passed four rendered views as well as their camera poses as the conditions for the MVD model.

For both methods, the gradients are masked and only enabled in the masked area. Derived from text-to-image Stable Diffusion [40], SVD takes a sequence of images as input and allows an additional channel in the input to the U-Net by adding 3D attention layers across the time dimension to compute the self-attention within the batch of input images. Similarly, MVD utilizes 3D spatial attention to assess the view consistency of the 3D generation from 2D. The purpose of our trial is to investigate the possibility of directly using such models off-the-shelf in our 3D inpainting tasks to address cross-view consistency. The visual results are shown in Figure 12. We found that, although efficient in 3D generation tasks, these models are not satisfactory without being further modified and fine-tuned for 3d inpainting tasks, because they do not enable inpainting conditions



(b) Final Disparity Map
w/ CDS Geometry



(c) Final Disparity Map
w/ Vanilla SDS Geometry

Figure 11. (a) Collaborative Inpainting with Grid-based denoising does not yield consistent inpainting. The top row is the masked normal maps and the bottom row is the inpainted normal maps (normal maps of the bench scene in SPIn-NeRF). We observed inconsistent inpainting results. Comparing the final results with Vanilla SDS geometry guidance (c) and CDS geometry guidance (b), we observed increased artifacts in the latter.

where the masks and masked images are passed to the U-Net to specify the area and the context to inpaint. As a result, the 3D generation diffusion models will be prompted to generate 2D prior, which will change the entire image/view rather than just the masked region. The resulting distillation results do not constitute to a promising 3D inpainting even if we constrain the gradient flow to only the masked region.

9.3. Vanilla Geometry SDS v.s. CDS Geometry SDS

As a major contribution of our method, we apply collaborative SDS in the color space to tackle the occlusion problem in 3D inpainting. Yet, for the geometry SDS of a NeRF scene, we only apply vanilla geometry SDS, where we denoise a single normal map per iteration without collaboratively computing the cross-view loss. This is because, during the experiments, we found that CDS Geometry SDS does not yield consistent distillation in the geometry space. The inconsistency among the different views of normal maps easily leads to convergence into inpainting with artifacts in the geometry space as shown in Figure 11.

As a result, we do not apply collaborative SDS to the geometry space of our inpainting method. This can be attributed to the priors in diffusion models being trained mostly on large-scale datasets of natural RGB images paired with textual descriptions (e.g., "a mountain at sunset"), while normal maps are specialized data representing surface orientations using encoded RGB values (usually indicating x, y, z surface normals), which differ fundamen-

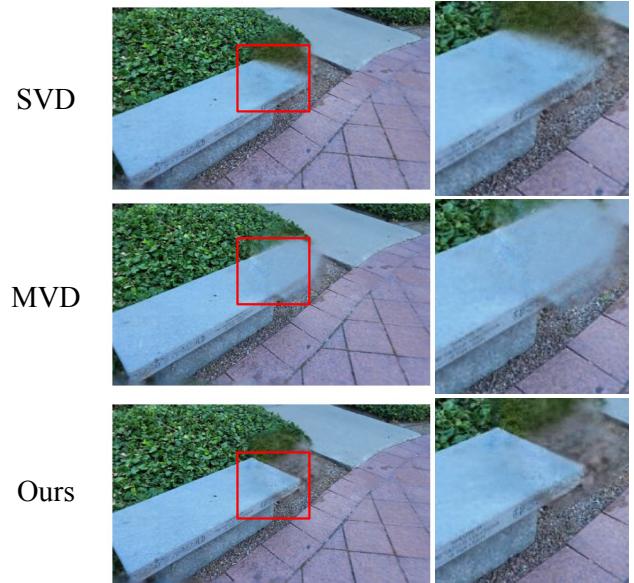


Figure 12. Qualitative comparison between 3D-attention-based diffusion models (SVD & MVD) and our choice of SD. We observe that SVD does not converge to a sharp reconstruction. Meanwhile, MVD does not yield a correct or faithful reconstruction of the scene.

tally from natural RGB images in structure and meaning.

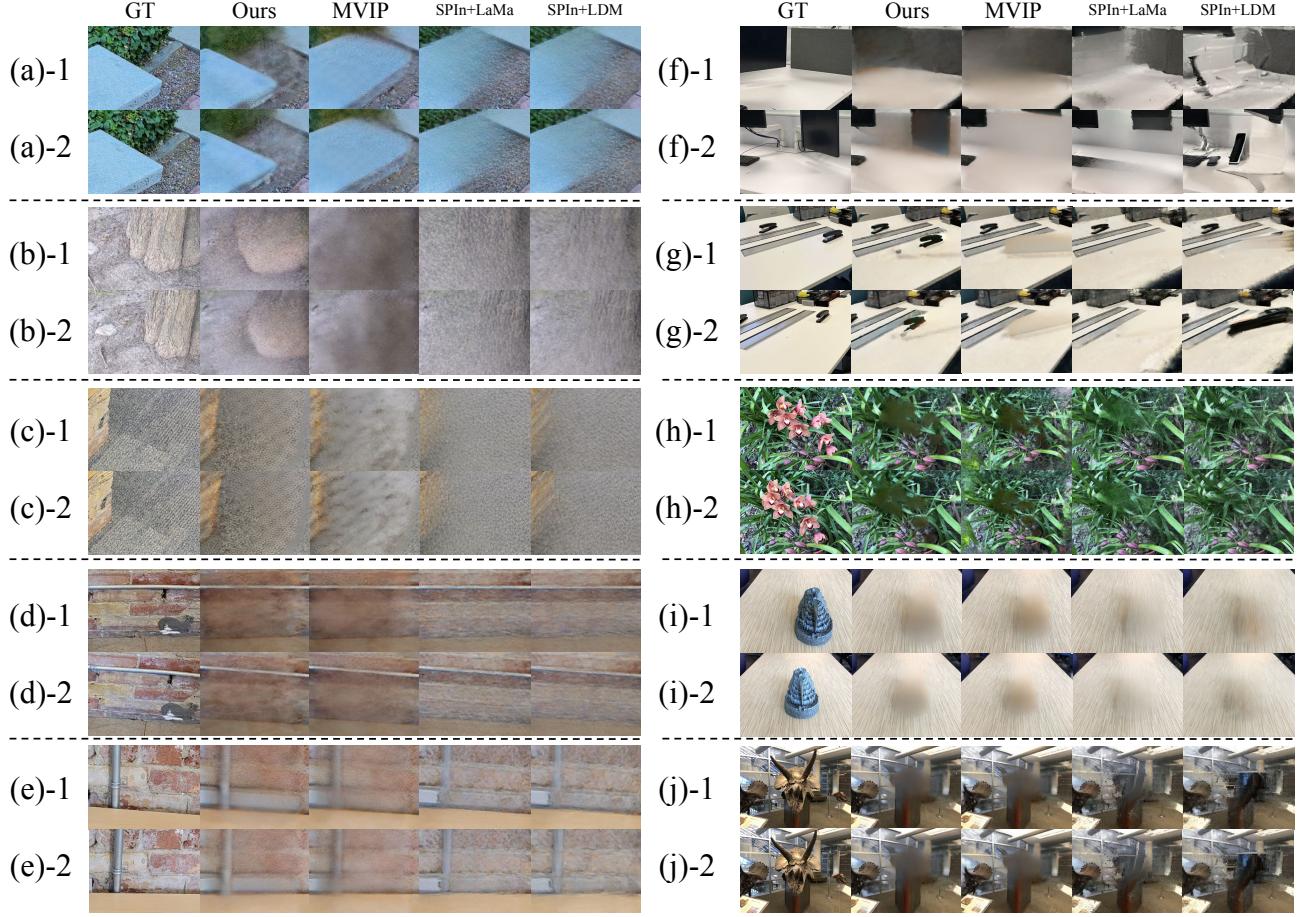


Figure 13. More qualitative results from our experiments. We visualize only the masked region for comparisons. Note that for the LLFF dataset ((h), (i), and (j)), there is no ground truth with the object removed. Thus we show the ground truths with objects instead. We observed more faithful reconstructions of the true scenes with our method in severely occluded cases ((a), (b), (c), (f), (g)). We also witness more cross-view consistency of our method compared to SPIn-NeRF-based methods ((f) and (j)). However, our method is limited in reconstructing high-frequency texture regions in the scene ((d) and (e)) the brick wall, (h) the orchid leaves, and (i) the table), which is a common challenge in SDS-based methods.

10. Qualitative Results

In this section, we present more qualitative results and discuss the shortcomings of our Occlude-NeRF method.

As shown in Figure 13, our methods can faithfully reconstruct the occluded areas with limited information compared to the baseline methods ((a) the true edge of the stone bench, (b) the root location of the trunk, (c) the shape of the wall corner, (f) the edge of the monitor, and (g) the location of the stapler). While being able to reconstruct the occluded area, our method maintains satisfactory visual reconstruction in the cases where occlusion is not severe ((d) and (e) the location of the pipe).

10.1. Limitation: High-frequency Region Reconstruction

As prior works [24, 55] pointed out, recovering high-frequency regions remains a common challenge for 3D generative methods like SDS. In this subsection, we showcase Occlude-NeRF’s limitation in generating high-frequency regions in Figure 13. Specifically, in scenes (d) and (e), we observe that the brick wall texture in ours and MVIP’s is blurred compared to that in SPIn-NeRF-based methods. Similarly, in (c) the gray carpet is rendered with artifacts with dot texture instead of the real texture of the carpet. Moreover, the patterns of the orchid leaves in (h) and the texture of the table in (i) are both blurred in ours and MVIP’s, while being sharper and clearer in those in SPIn-NeRF-based methods. This is attributed to the mechanism of SDS-based methods. Repetitive updates in SDS will av-

erage out the shape of high-frequency objects in the scene. To tackle this problem, we anticipate future work introducing a more advanced noise scheduling mechanism, where the shape of the reconstruction can be affirmed in the early stages and sharpened in the later stages to avoid blurriness.

11. Ethical Concerns

The ethical concerns of our method primarily revolve around its potential misuse and implications for privacy, authenticity, and societal impact [44, 45]. Similar algorithms have been applied to editing humanoid avatars [30, 46] and objects [12, 18] in virtual environments. The capability of this type of algorithm might be exploited to fabricate or manipulate digital evidence, misrepresent physical spaces, or breach privacy by reconstructing obscured or private areas without consent. Additionally, biases inherent in training datasets could lead to unfair or inaccurate reconstructions, potentially reinforcing stereotypes or producing misleading results, this is an especially common downfall of text-prompted generative AIs.