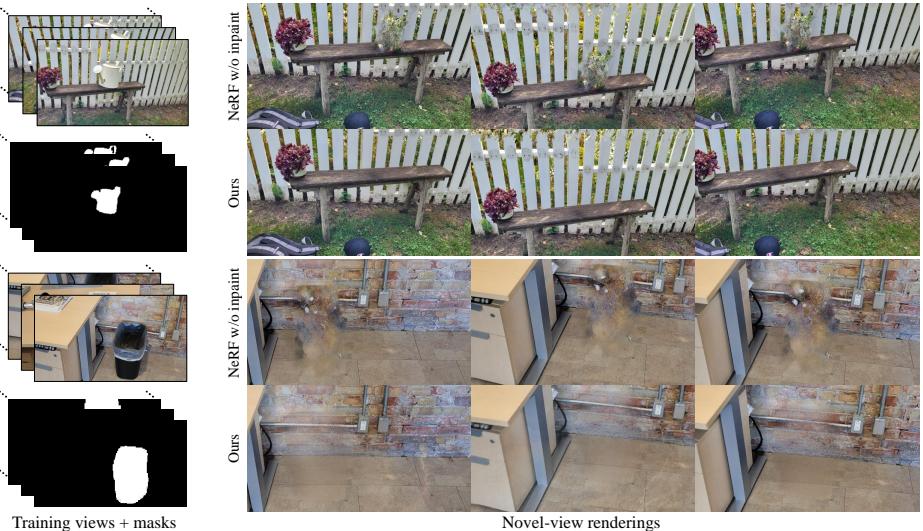


# Taming Latent Diffusion Model for Neural Radiance Field Inpainting

Chieh Hubert Lin<sup>1,2</sup>, Changil Kim<sup>1</sup>, Jia-Bin Huang<sup>1,3</sup>, Qinbo Li<sup>1</sup>, Chih Yao Ma<sup>1</sup>, Johannes Kopf<sup>1</sup>, Ming-Hsuan Yang<sup>2</sup>, and Hung-Yu Tseng<sup>1</sup>

<sup>1</sup>Meta, <sup>2</sup>University of California, Merced, <sup>3</sup>University of Maryland, College Park

<https://hubert0527.github.io/MALD-NeRF>



**Fig. 1: NeRF inpainting.** Given a set of posed images associated with inpainting masks, the proposed framework estimates a NeRF that renders high-quality novel views, where the inpainting region is realistic and contains high-frequency details.

**Abstract.** Neural Radiance Field (NeRF) is a representation for 3D reconstruction from multi-view images. Despite some recent work showing preliminary success in editing a reconstructed NeRF with diffusion prior, they remain struggling to synthesize reasonable geometry in completely uncovered regions. One major reason is the high diversity of synthetic contents from the diffusion model, which hinders the radiance field from converging to a crisp and deterministic geometry. Moreover, applying latent diffusion models on real data often yields a textural shift incoherent to the image condition due to auto-encoding errors. These two problems are further reinforced with the use of pixel-distance losses. To address these issues, we propose tempering the diffusion model’s stochasticity with per-scene customization and mitigating the textural shift with masked adversarial training. During the analyses, we also found the commonly used pixel and perceptual losses are harmful in the NeRF inpainting task. Through rigorous experiments, our framework yields state-of-the-art NeRF inpainting results on various real-world scenes.

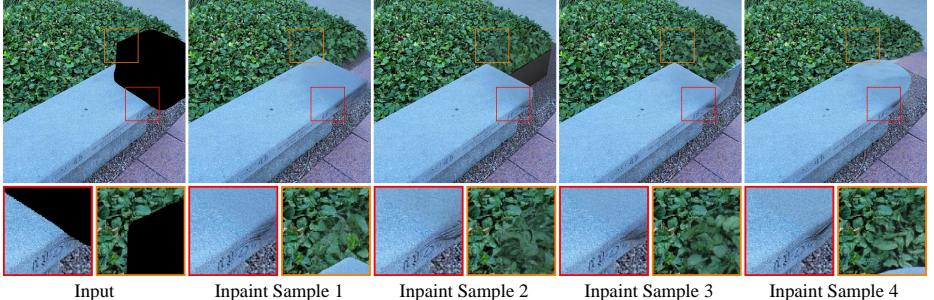
## 1 Introduction

The recent advancements in neural radiance fields (NeRF) [3, 24, 27] have achieved high-quality 3D reconstruction and novel-view synthesis of scenes captured with a collection of images. The success intrigues an increasing attention on manipulating NeRFs such as 3D scene stylization [8, 38] and NeRF editing [13]. In this work, we focus on the *NeRF inpainting* problem. As shown in Figure 1, given a set of images of a scene with the inpainting masks, our goal is to estimate a completed NeRF that renders high-quality images at novel viewpoints. The NeRF inpainting task enables a variety of 3D content creation applications such as removing objects from a scene [26, 39], completing non-observed part of the scene, and hallucinating contents in the designated regions.

To address the NeRF inpainting problem, existing algorithms first leverage a 2D generative prior to inpaint the input images, then optimize a NeRF using the inpainted images. Several efforts [25, 26, 35] use the LaMa [35] model as the 2D inpainting prior. Driven by the recent success of diffusion models [4, 9, 10, 30, 32], recent work [28, 39] use the latent diffusion model [30] to further enhance the fidelity. Nevertheless, unrealistic visual appearance and incorrect geometry are still observed in the inpainted NeRFs produced by these methods.

Leveraging 2D latent diffusion models for NeRF inpainting is challenging for two reasons. First, the input images inpainted by the 2D latent diffusion model are not 3D consistent. The issue leads to blurry and mist-alike results in the inpainting region if pixel-level objectives (i.e., L1, L2) are used during NeRF optimization. Several methods [26, 39] propose to use the perceptual loss function [46] to mitigate the issue. Although the strategy improves the quality, the results still lack high-frequency details. Second, as shown in Figure 2, the pixels inpainted by the latent diffusion model typically showcase a texture shift compared to the observed pixels in the input image. The issue is due to the auto-encoding error in the latent diffusion model. It introduces noticeable artifacts in the final inpainted NeRF (i.e., the clearly visible seam between the reconstructed and inpainted region).

In this paper, we propose to use a masked adversarial training to address the two above-mentioned issues. Our goal is to use the latent diffusion model to inpaint input images, and optimize a NeRF that 1) contains high-frequency details in the inpainted region, and 2) does not show texture difference between the inpainted and reconstructed regions. Specifically, we introduce a patch-based adversarial objective between the inpainted and NeRF-rendered images to NeRF optimization. Since the objective is not affixed to particular pixel similarity, we are capable of promoting high-frequency details without relying on absolutely consistent image pixels across the inpainted images. However, simply applying the patch-based adversarial loss does not eliminate texture shift around the inpainting boundary, as it exists in the “real” examples (i.e., inpainted images) of the adversarial training. To handle this, we design a masked adversarial training scheme to hide such boundaries from the discriminator, hence achieving improved quality. In addition to the masked adversarial training, we apply per-scene customization to finetune the latent diffusion model [18, 31], encouraging



**Fig. 2: Inconsistency and texture shift issue.** We present the 2D inpainting results from our latent diffusion model. Given the same input image and mask, the results are 1) not consistent and 2) produce a texture shift between the original and inpainted pixels. These issues introduce noticeable artifacts in the NeRF inpainting results.

the model to generate contents that are more coherent to the reconstructed scene. We find that the approach enhances the consistency across different input images, thus improving the quality of the final inpainted NeRF.

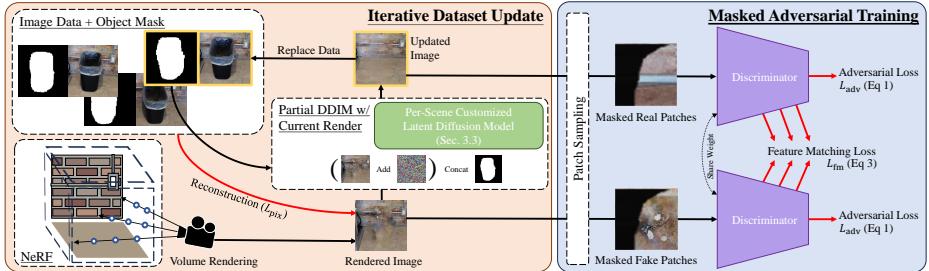
We conduct extensive quantitative and qualitative experiments on two NeRF inpainting benchmark datasets consisting of multiple challenging real-world scenes. Our proposed method, name MALD-NeRF, achieves state-of-the-art NeRF inpainting performance by marrying the merits of the Masked Adversarial learning and the Latent Diffusion model. MALD-NeRF synthesizes inpainting areas with high-frequency details and mitigates the texture shift issues. In addition, we conduct extensive ablation studies to dissect the effectiveness of each component and the effect of different loss function designs. We summarize the contributions as follows:

- We design a masked adversarial training scheme for NeRF inpainting with diffusion. We show that the design is more robust to 3D and textural inconsistency caused by the 2D inpainting latent diffusion model.
- We harness the generation diversity of the latent diffusion model with per-scene customization. In combination with the iterative dataset update scheme, our framework yields better convergence while training inpainted NeRF.
- We achieve state-of-the-art NeRF inpainting performance.

## 2 Related Work

### 2.1 3D Inpainting

Inpainting 3D data is a long-standing problem in computer vision, such as point clouds [47] and voxel [47] completion. The classical approach is to collect a large-scale dataset of the target data distribution, then train a closed-form model on the data distribution by sampling random inpainting masks [35]. However, to achieve high-quality results with a generalizable inpainting model, the approach requires a large-scale dataset that maintains high diversity and aligns with the testing distribution. Currently, there is no existing large-scale NeRF dataset



**Fig. 3: Method overview.** The proposed method uses a latent diffusion model to obtain the inpainted training images from the NeRF-rendered images using partial DDIM. The inpainted images are used to update the NeRF training dataset following the iterative dataset update protocol. (*reconstruction*) We use pixel-level regression loss between the NeRF-rendered and ground-truth pixels to reconstruct the regions observed in the input images. (*inpainting*) We design a masked patch-based adversarial training, which include an adversarial loss and discriminator feature matching loss, to supervise the the inpainting regions.

or feasible methods to directly train a closed-form inpainting model on NeRF representation.

## 2.2 NeRF Inpainting

Recent studies on NeRF inpainting focus on inpainting individual 2D images and mitigating the 3D inconsistency problem with different solutions. NeRF-In [33] shows that simply training the NeRF with pixel reconstruction loss with inpainted images leads to blurry results. SPIn-NeRF [26] is a more recent work that proposes that replacing pixel loss with more relaxed perceptual loss enables the NeRF to reveal more high-frequency details, and the method also leverages depth predictions to supervise the NeRF geometry. [42] is a concurrent work of SPIn-NeRF, which also utilizes LaMa [35] as the image inpainter. However, as shown in Figure 4, the overall LaMa inpainting quality remains sub-optimal compared to diffusion models, leading to less compelling visual quality. InpaintNeRF360 [39] also adopts a similar strategy that relies on perceptual and depth losses. However, in our study, we found using perceptual loss does not fundamentally resolve the problem, and often leads to sub-optimal quality. Reference-guided inpainting [25] introduces a more sophisticated mechanism with careful per-view inpainting, view selection, and inpainting by referencing previous results to address the cross-view



**Fig. 4: 2D image inpainting comparisons.** The inpainting quality of the LaMa [35] approach is less realistic compared to the LDM [30] model.

inconsistency. However, the approach is highly heuristic and sophisticated, the authors have not released either the implementation or rendering results. We have reached out to the authors to obtain visual results for comparison, but we did not receive responses. Inpaint3D [28] proposes to train a diffusion model on the RealEstate10k [48] dataset and shows high-quality scenes from a similar data distribution. However, due to the limited size of the training dataset, the method is unable to generalize to arbitrary scenes, such as the SPIn-NeRF benchmark. In contrast, we utilize a diffusion model pretrained on a large-scale internal dataset that serves as a strong prior model with outstanding generalization and shows that our method can be applied to unseen dataset distribution, such as the SPIn-NeRF benchmark.

### 2.3 Sparse-View NeRF Reconstruction Using Generative Prior

Reconstructing a NeRF from a sparse set of a few images [45] is a popular research topic due to its wide applications. Several recent work proposes to utilize generative priors, such as diffusion models [21, 43, 44] and GANs [29]. Despite utilizing generative priors, the problem focuses on finding reasonable geometric correspondence from ill-conditioned sparse image sets and enhancing the surface quality of low-coverage regions. The line of work does not consider the visual plausibility of uncovered regions, nor intentionally create disocclusion by removing objects. It is worth noting such a distinction is significant, as the sparse-view reconstruction problem assumes the true geometry is accessible (despite being ill-conditioned) from the training views with fully trusted pixels. Therefore, these algorithms are less concerned with the cross-view inconsistency problems caused by the generative models. In contrast, the NeRF inpainting problem requires the algorithms to form the inpainted geometry from scratch.

## 3 Methodology

### 3.1 Preliminaries

**Neural radiance fields (NeRFs).** Given a set of  $N$  images  $\{I_i\}_{i=1\dots N}$  with camera poses  $\{P_i\}_{i=1\dots N}$ , NeRFs aim to represent the 3D scene using a neural function  $f_\theta$ . The neural function  $f_\theta$ , which can be implicit MLPs [24] or voxelized 3D volumes [27], learns to map the 3D position along with viewing direction to the corresponding density and color. By applying volume rendering [11, 24], we can optimize a NeRF using a pixel-level regression loss between the rendered pixels  $\{x\}$  and ground-truth pixels  $\{\hat{x}\}$ .

**NeRF inpainting.** In addition to the images, we are given a set of binary masks  $\{M_i\}_{n=1\dots N}$  in this problem. The binary masks split the image pixels into two distinct sets: the unmasked pixels  $\{x_j^r\}$  that are used for reconstructing the observed part of the scene, and the masked pixels  $\{x^m\}$  indicating the unknown regions to be inpainted.

### 3.2 NeRF Inpainting with Latent Diffusion Models

Figure 3 presents an overview of the proposed framework. We use a latent diffusion model pre-trained on an internal image inpainting dataset to inpaint the 2D images, then replace the input images for NeRF training. The pixel-level loss function is used for reconstructing the known region in the input images, while the masked adversarial training is used for the inpainting region. We detail each component as follows.

**Reconstructing observed regions.** For reconstruction pixels, we supervise the NeRF model with an L2 objective  $L_{\text{pix}} = \|x_j^r - \hat{x}_j^r\|_2$ . The inter-level loss  $L_{\text{inter}}$  [1], distortion loss  $L_{\text{distort}}$  [1] and hash decay  $L_{\text{decay}}$  [3] are also used for regularization.

**Masked adversarial training for inpainting regions.** We do not use pixel distance losses in the inpainting region, as they are not robust to the highly diverse and 3D inconsistent inpainting results leading to blurry mist-like NeRF renderings. To address the issue, we use adversarial loss [12, 20] and the discriminator feature matching loss [41] to guide the supervision of NeRF in the inpainting regions. Specifically, we consider the patches of inpainted images as real examples, and the NeRF-rendered patches as the fake ones in the adversarial training.

However, as shown in Figure 2, the real pixels and the inpainted pixels have a textural shift, causing the discriminator to exploit such a property to recognize the real image patches. In this case, the discriminator promotes the textural discrepancy between the NeRF-rendered pixels in the reconstruction and inpainting regions. To alleviate the issue, we design a masked adversarial training scheme to hide the reconstruction/inpainting boundary on the image patches from the discriminator. For both NeRF-rendered and diffusion-inpainted images, we only keep the pixels within the inpainting mask region, and mask the pixels outside the masked region with black pixels. The design is conceptually similar to AmbientGAN [7] which trains the discriminator with corrupted inputs based on the underlying task of interest. In particular, our objective is to reduce the textural gap between the inpainting and non-inpainting renderings, instead of solving image restoration tasks. We later show that such a design indeed leads to superior inpainting geometry and performs better in quantitative evaluations in the ablation study.

Given the masking functions  $C^m$  for the inpainting region and  $C^r$  for the non-inpainting region, the adversarial loss for the discriminator  $D$  training is

$$L_{\text{adv}} = f(D(C^m(x^m))) + f(-D(C^r(\hat{x}^r))) \quad (1)$$

where  $f(x) = -\log(1 + \exp(-x))$ . We use the StyleGAN2 [20] discriminator architecture and train the discriminator with patches [19]. In addition, we use R1 regularizer [22] to stabilize the discriminator training with

$$L_{\text{GP}} = \|\nabla D(C^r(\hat{x}^r))\|_2^2. \quad (2)$$

Meanwhile, we extract the discriminator intermediate features after each discriminator residual blocks with  $F$ , then calculate the discriminator feature matching

loss [41] to supervise the inpainting area

$$L_{\text{fm}} = \|F(C^m(x^m)) - F(C^m(\hat{x}^m))\|_1. \quad (3)$$

**Monocular depth supervision for inpainting regions.** We leverage an off-the-shelf monocular depth prior to regularize the geometry of the learned NeRF. We use ZoeDepth [5] to estimate the depth  $\hat{d}_i$  of the inpainted images, then render the NeRF depth  $d_i$  by integrating the density along the radiances. Since the two depths are in different metrics, similar to [17], we solve a shift-scale factor between the two depth maps. In particular, we only use the reconstruction region to compute the shift-scale factor, and use the solved factor to rescale the estimated depth into the final depth supervision  $\hat{d}_i$ . Since the computation requires meaningful NeRF depths, we start applying the depth supervision after 2,000 iterations. We use a ranking depth loss  $L_D$  proposed in SparseNeRF [40].

**Total training objective.** Each training iteration of MALD-NeRF consists of three steps, each step optimizes the modules with different objectives. A reconstruction step optimizing NeRF with

$$L^r = \lambda_{\text{pix}} L_{\text{pix}} + \lambda_{\text{inter}} L_{\text{inter}} + \lambda_{\text{distort}} L_{\text{distort}} + \lambda_{\text{decay}} L_{\text{decay}}. \quad (4)$$

An inpainting step optimizing NeRF with

$$L^m = -\lambda_{\text{adv}} L_{\text{adv}} + \lambda_{\text{fm}} L_{\text{fm}} + \lambda_{\text{inter}} L_{\text{inter}} + \lambda_{\text{distort}} L_{\text{distort}} + \lambda_{\text{decay}} L_{\text{decay}}. \quad (5)$$

Finally, the discriminator training step using objective

$$L^D = L_{\text{adv}} + \lambda_{\text{GP}} L_{\text{GP}}. \quad (6)$$

The  $\lambda$ 's are weighting factors of these objectives.

**Iterative dataset update and noise scheduling in inpainting diffusion.** In practice, directly inpainting the input images leads to inconsistency issues across various viewpoints due to the high diversity and randomness of the diffusion model. We leverage two strategies to mitigate the issue. First, we use an iterative dataset update (IDU) approach similar to [13], where we gradually update the inpainting region every  $U$  iterations throughout the training. Second, the inpainting diffusion model only performs a partial DDIM [34] starting from time step  $t$  based on the current NeRF rendering. The time step  $t$  is determined based on the ratio of the training progress, i.e., the earlier the training is, the more noise is added to the inpainting region for denoising. Such a design aims to leverage the 3D consistency of NeRF rendering and gradually propagate such 3D consistent information to all images in the dataset. Specifically, we use the HiFA [49] scheduling that sets  $t = t_{\max} - (t_{\max} - t_{\min}) * \sqrt{k/K}$ , where  $k$  is the current NeRF training time step,  $K$  is the total iterations of the NeRF training, and  $(t_{\max}, t_{\min}) = (980, 20)$  is the DDPM [15] time steps used within the latent diffusion model.

### 3.3 Per-Scene Customized Latent Diffusion Model

In order to harness the expressiveness of the diffusion model and avoid synthesizing too many out-of-context objects that confuse the convergence of the inpainted NeRF. We finetune the inpainting diffusion model in each scene. For each scene, we set a customized text token for the scene, and LoRA [18] finetune both the text encoder and U-Net. We use a self-supervised inpainting loss similar to [35, 37]. For each image, we sample arbitrary rectangular inpainting masks and take the union of masks as the training mask. The LoRA-finetuning model is being supervised to inpaint the latent values within the training mask. Following the DDPM [15] training, we supervise the U-Net of the diffusion with L2 distance at a random time step. Meanwhile, since we are working on an object removal task, each image is paired with a 3D consistent mask that does not have ground-truth supervision. We set the loss values to zero within the object removal mask region.

## 4 Experiments

### 4.1 Experimental Setups

**Datasets.** We use two real-world datasets for all experiments:

- **SPIIn-NeRF** [26] is an object removal benchmark dataset consisting of 10 scenes. Each scene has 60 training views captured with an object (to be removed), and each view is associated with an inpainting mask indicating the desired object to be removed. In addition, each scene contains 40 testing views in which the object is physically removed during the capture.
- **LLFF** [23] consists of multiple real-world scenes with varying numbers of images (20-45). We use a six-scene subset provided by SPIIn-NeRF annotated with 3D grounded object removal masks.

Following prior work [26, 28], we resize all the images to have a long-edge size of 1008.

**Evaluation setting.** We follow the protocol in the SPIIn-NeRF [26] work. We optimize the NeRF using the training view images (with objects) associated with the inpainting masks for all compared methods. We only use the test views, where the object is physically removed from the scene, to compute the below metrics for evaluation:

- **LPIPS**: We use the LPIPS score [46] to measure the perceptual difference between the NeRF-rendered and ground-truth test view images.
- **M-LPIPS**: To better understand the inpainting performance, we *mask out* the region outside the object inpainting mask and measure the LPIPS score.
- **FID/KID**: As shown in Figure 5, the LPIPS score is not a proper metric for generative tasks. For instance, although the object generated in the inpainting area is valid content, it produces a high perceptual distance to the ground truth test view. To address the issue, we additionally report the FID [14] and KID [6] scores, which are commonly used metrics in generative

**Table 1: Quantitative comparisons.** We present the results on the SPIIn-NeRF [26] and LLFF [23] datasets. Note that the LLFF dataset does not have ground-truth views with object being physically removed, therefore, we only measures C-FID and C-KID on these scenes. The best performance is underscored.

Methods	SPIIn-NeRF				LLFF	
	LPIPS (↓)	M-LPIPS (↓)	FID (↓)	KID (↓)	C-FID (↓)	C-KID (↓)
SPIIn-NeRF	0.5356	0.4019	219.80	0.0616	231.91	0.0654
SPIIn-NeRF (LDM)	0.5568	0.4284	227.87	0.0558	235.67	0.0642
Inpaint3d	0.5437	0.4374	271.66	0.0964	—	—
InpaintNeRF360	0.4694	0.3672	222.12	0.0544	174.55	0.0397
Ours	<u>0.4345</u>	<u>0.3344</u>	<u>183.25</u>	<u>0.0397</u>	<u>171.89</u>	<u>0.0388</u>

model literature that quantify the distributional similarity between two sets of images and are sensitive to the visual artifacts. For each evaluated method, we compute the scores using NeRF-rendered and ground-truth images of all test views across all scenes in the dataset.

- **C-FID/C-KID:** For the LLFF dataset, since the dataset does not include test views with the object being physically removed, we alternatively measure the visual quality near the inpainting border. More specifically, we find the four furthest corners of the inpainting mask and crop image patches centered at these corners. Then, finally, compute the FID/KID scores between the real-image patches and the NeRF-rendering after the object is removed and inpainted.

**Evaluated methods.** We compare our method with the following approaches:

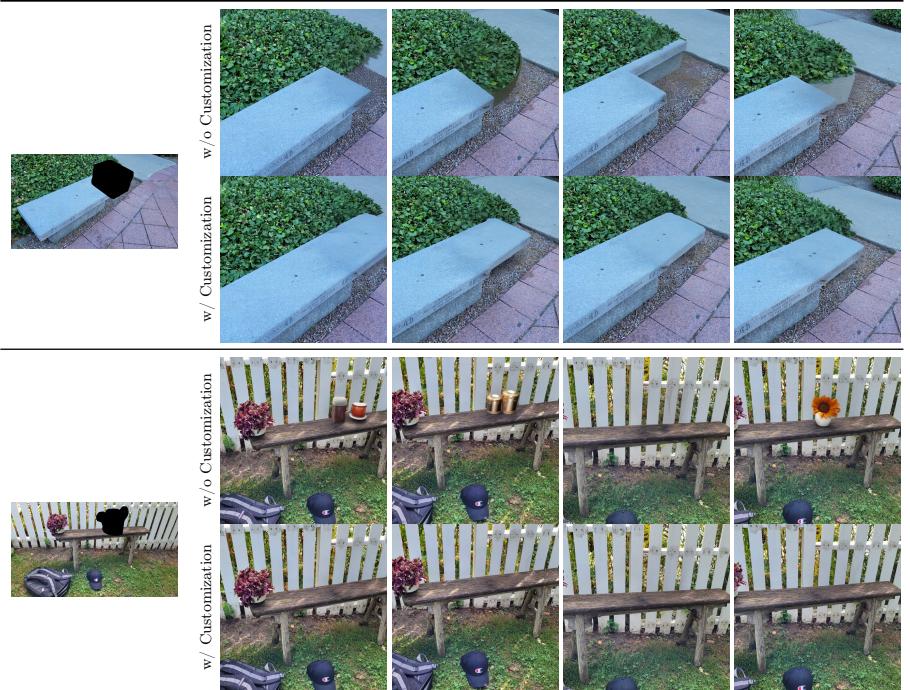
- **SPIIn-NeRF** [26]: We use the official implementation. Note that the authors do not provide the evaluation implementation. Therefore, the LPIPS scores reported in this paper differ from those presented in the SPIIn-NeRF paper. Nevertheless, we have contacted the authors to ensure that the SPIIn-NeRF results match the quality shown in the original paper.
- **SPIIn-NeRF (LDM)**: We replace the LaMa [35] model to our latent diffusion model in the SPIIn-NeRF approach while maintaining all default hyper-parameter settings.
- **InpaintNeRF360** [39]: We implement the algorithm as no source code is available. Specifically, we use our latent diffusion model for per-view inpainting and optimize the NeRF with the same network architecture devised in our approach with the objectives proposed in the paper.
- **Inpaint3d** [28]: We reach out to the authors for all the rendered images of test views for evaluation on the SPIIn-NeRF dataset.

## 4.2 Per-Scene Finetuning

In Figure 6, we qualitatively show the effectiveness of our per-scene finetuning on the latent diffusion model. Before the finetuning, our latent diffusion model inpaints arbitrary appearance, and even often time creates arbitrary objects in the inpainting region. Such a high variation is a major issue causing the NeRF

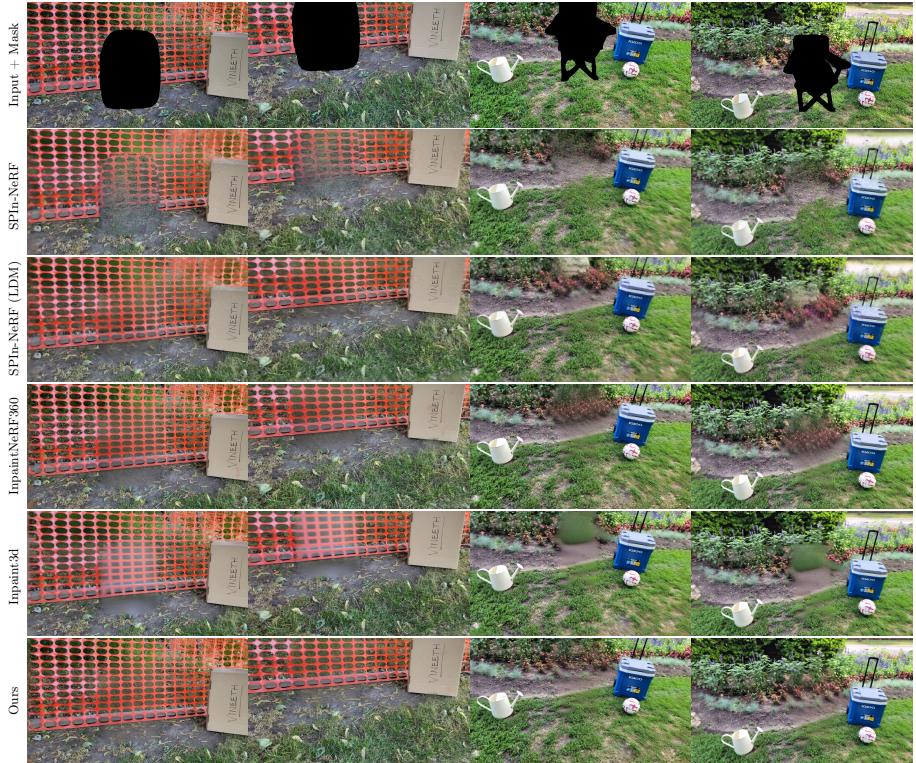


**Fig. 5: Drawbacks of LPIPS.** In some cases, the LPIPS score fails to indicate the visual quality. For example, generating a realistic baseball cap actually lowers the score as there is no object in the inpainting area in the ground truth image.



**Fig. 6: Per-scene customization.** Our per-scene customization effectively forges the latent diffusion model to synthesize consistent and in-context contents across views.

unable to converge, and often creates floaters and mist-like artifacts. Note that we used a text prompt “photo realistic, high quality, high resolution” and a negative prompt [16] “artifacts, low resolution, unknown, blur, low quality, human, animal, car.” In contrast, after the finetuning, the inpainted results maintain a high consistency across individual inpainting results. Note that we do not need extra prompts or negative prompts, since the finetuned model has learned scene-dependent tokens and finetuned the texture encoder with respect to the customized tokens.



**Fig. 7: Qualitative comparisons.** We present the results on the SPIn-NeRF [26] dataset. More results are shown in the supplementary material.

### 4.3 NeRF Inpainting

**Qualitative analysis.** In Figure 7, we show the two most challenging scenes in the SPIn-NeRF dataset. The scene with an orange net requires inpainting both the periodic textures of the net as well as the complex leaves on the ground. Only our MALD-NeRF and InpaintNeRF360 are able to correctly complete the repetitive pattern of the net. Moreover, our method can further complete some of the leaves on the net, while other methods all converge to blurry appearances. The second scene, removing the chair from a garden environment, involves completing complex plant textures at different granularity and patterns. Only our MALD-NeRF can preserve the high-frequency details and seamlessly inpaint all plants by following the heterogeneous textures in the corresponding area. All baseline methods converge to a blurry appearance due to the complexity of the texture and cross-view inconsistency from the inpainting prior. In particular, it is worth noting that Inpaint3D trains their method on RealEstate10k [48], which is one of the largest publicly available datasets for scene reconstruction. The method shows outstanding performance in environments similar to the RealEstate10k distribution, but shows a significant performance drop in real-world environments, such as the SPIn-NeRF dataset. This is potentially due to the limited

**Table 2: Quantitative ablation study.** We use the SPIIn-NeRF dataset [26] to validate the effectiveness of masked adversarial loss, feature matching loss, per-scene customization, and if the common-used pixel-level (i.e., I-Recon) and LPIPS loss functions are required. We highlight the performance which is worse than our complete approach **red**, and mark the better one in **green**.

Methods	LPIPS ( $\downarrow$ )	M-LPIPS ( $\downarrow$ )	FID ( $\downarrow$ )	KID ( $\downarrow$ )
Ours – Adv + I-Recon	<b>0.6623</b>	<b>0.5236</b>	<b>305.60</b>	<b>0.1177</b>
Ours – Adv + LPIPS	0.4231	0.3147	<b>192.86</b>	<b>0.0447</b>
Ours – Adv + I-Recon + LPIPS	<b>0.4359</b>	0.3172	<b>199.10</b>	<b>0.0458</b>
Ours + I-Recon	<b>0.5106</b>	<b>0.3730</b>	<b>256.82</b>	<b>0.0827</b>
Ours + LPIPS	0.4130	0.3130	<b>185.79</b>	<b>0.0419</b>
Ours – Per-Scene Finetune	<b>0.4894</b>	<b>0.3862</b>	<b>224.29</b>	<b>0.0596</b>
Ours – Feature Matching	<b>0.4382</b>	<b>0.4002</b>	<b>232.28</b>	<b>0.0716</b>
Ours – Adv Masking	<b>0.4367</b>	<b>0.3358</b>	<b>196.47</b>	<b>0.0472</b>
Ours	0.4345	0.3344	183.25	0.0397

dataset scale of RealEstate10k, and further reinforces the merits of utilizing a much more generalizable diffusion prior pretrained on the large-scale image dataset. We include all visual comparisons in the supplementary material.

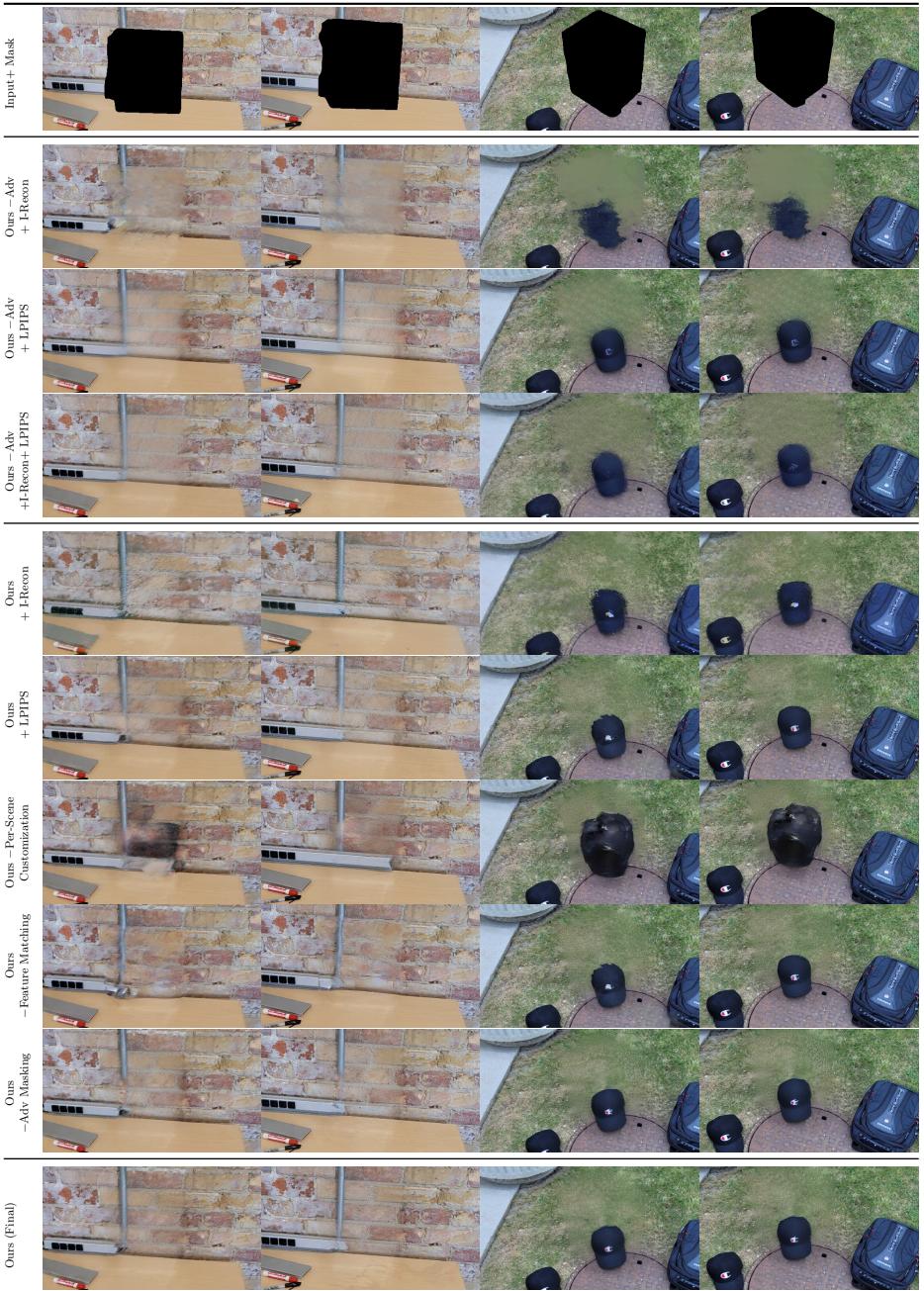
**Quantitative evaluation.** The quantitative analysis in Table 1 shows our method outperforms all methods in all metrics on both SPIIn-NeRF and LLFF benchmarks. Especially, the FID and KID metrics that focus on measuring the visual quality both indicate our method outperforms baselines by a large margin.

#### 4.4 Ablation Study

We conduct a detailed ablation on the SPIIn-NeRF dataset, which has a physically correct ground truth that measures the performance more reliably. Our ablation study is conducted in three major sections.

We first show that removing our adversarial loss leads to worse visual quality in FID and KID measures, and such a performance gap cannot be closed by any combination of pixel reconstruction or LPIPS perceptual losses. Note that, as mentioned in Figure 5, the gain in the LPIPS score maintains high stochasticity and does not reflect the actual perceptual quality. Furthermore, it is expected that a method optimized toward the LPIPS networks should yield a more favorable LPIPS score, but the FID and KID scores indicate such a performance gain does not convert to a better visual quality. The qualitative samples in Figure 8 also show that the inpainted results from these methods maintain a much blurry appearance compared to our final method, but the LPIPS measurement is unable to reflect such an apparent blurriness.

Second, we ablate that neither pixel nor LPIPS losses can provide better visual quality when combined with our adversarial-based training scheme. In particular, we found combining the pixel loss and our adversarial scheme causes instability that often leads to gradient explosion and fails the whole experiment. On the other hand, as we discussed in the introduction and Figure 2, the LDM



**Fig. 8: Qualitative ablation study.** In this study, we use the SPIn-NeRF dataset [26] to validate the effectiveness of masked adversarial loss, feature matching loss, per-scene customization, and if the commonly-used pixel-based (I-Recon) and LPIPS are needed.

model creates discontinuities between the inpainting region and the real pixels. The LPIPS objective, which has the receptive field crossing the discontinuous border, encourages to keeping a rather sharp border between the reconstruction and inpainting area, and even propagates the faulty gradients caused by the discontinuity into neighboring areas. Such behavior leads to poorer geometry near the border of the inpainting mask. In Figure 8, notice the “Ours + LPIPS” method not only has a clear discontinuity on the brick wall and the grass area, but also has a poorer geometry on the iron tube and the baseball cap. Note that the white artifacts on the table of the brick wall scene are irrelevant reflections of the foreground object introduced from other views, which is consistent and shared among all experimental results.

The third and the last part of the ablation shows the individual performance gain from each of our proposed components. In Table 2, removing any of the components leads to a consistent and significant performance loss in all measures. In Figure 8, we show that the behavior of each proposed component is consistent with our motivation. Removing the per-scene finetuning introduces random objects and creates obvious visual artifacts. Removing the feature matching loss simply unstabilizes the adversarial loss and creates obvious visual artifacts and wrong geometry. Training the adversarial loss without our masked adversarial scheme encourages the discriminator to keep the discontinuity between the real and inpainted region, and leads to worse geometry near the inpainting border. For instance, the continuity of the iron tube in the brick wall scene and the sharpness of the tree leaves in the garden scene are significantly impacted.

## 5 Conclusions and Discussions

In this work, we improve the NeRF inpainting performance by harnessing the latent diffusion model and solving optimization issues with a masked adversarial training scheme. We justify the significance of each proposed component via careful comparisons against state-of-the-art baselines and rigorous ablation studies.

**Potential negative impact.** NeRF inpainting with generative priors is closely related to generative inpainting, where certain frameworks aim to insert hallucinated content into a NeRF reconstruction. Such an application could lead to manipulating false information or creating factually wrong re-created renderings. Although we do not focus on such an application, which requires extra effort to harmonize and shadow-cast the inserted objects, our techniques could be utilized by these methods.

**Limitations.** Our framework involves an adversarial objective. Despite recent generative adversarial networks literature advancements, the framework’s performance remains highly stochastic. Also, it may not work well with low-shot NeRF reconstructions due to limited training data, or application scenarios with excessively large inpainting masks. Our framework significantly improves NeRF’s convergence by making the diffusion model generate consistent results across frames. However, our approach does not eliminate the microtextural variation

caused by the stochastic denoising process. Therefore, the inpainted texture remains visually more blurry than the real-world textures.

## References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV (2021) [6](#)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022) [18](#)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV (2023) [2](#), [6](#), [18](#)
4. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. <https://cdn.openai.com/papers/dall-e-3.pdf> (2023) [2](#)
5. Bhat, S.F., Birk, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) [7](#)
6. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: ICLR (2018) [8](#)
7. Bora, A., Price, E., Dimakis, A.G.: Ambientgan: Generative models from lossy measurements. In: ICLR (2018) [6](#)
8. Chiang, P.Z., Tsai, M.S., Tseng, H.Y., Lai, W.S., Chiu, W.C.: Stylizing 3d scene via implicit representation and hypernetwork. In: WACV (2022) [2](#)
9. Dai, X., Hou, J., Ma, C.Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenende, S., Wang, X., Dubey, A., et al.: Emu: Enhancing image generation models using photogenic needles in a haystack. In: NeurIPS (2022) [2](#)
10. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS (2021) [2](#)
11. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume rendering. ACM TOG (1988) [5](#)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS (2014) [6](#)
13. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: ICCV (2023) [2](#), [7](#), [19](#)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017) [8](#)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020) [7](#), [8](#)
16. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop (2021) [10](#)
17. Höllerin, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models. In: ICCV (2023) [7](#)
18. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) [2](#), [8](#)
19. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017) [6](#)

20. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020) [6](#), [18](#)
21. Liu, X., Kao, S.h., Chen, J., Tai, Y.W., Tang, C.K.: Deceptive-nerf: Enhancing nerf reconstruction using pseudo-observations from diffusion models. arXiv preprint arXiv:2305.15171 (2023) [5](#)
22. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: ICML (2018) [6](#)
23. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM TOG (2019) [8](#), [9](#)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) [2](#), [5](#)
25. Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinstein, A., Derpanis, K.G., Gilitschenski, I.: Reference-guided controllable inpainting of neural radiance fields. In: ICCV (2023) [2](#), [4](#)
26. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinstein, A.: Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023) [2](#), [4](#), [8](#), [9](#), [11](#), [12](#), [13](#)
27. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022) [2](#), [5](#), [18](#)
28. Prabhu, K., Wu, J., Tsai, L., Hedman, P., Goldman, D.B., Poole, B., Broxton, M.: Inpaint3d: 3d scene content generation using 2d inpainting diffusion. arXiv preprint arXiv:2312.03869 (2023) [2](#), [5](#), [8](#), [9](#)
29. Roessle, B., Müller, N., Porzi, L., Bulò, S.R., Kortschieder, P., Nießner, M.: Ganerf: Leveraging discriminators to optimize neural radiance fields. ACM TOG (2023) [5](#), [18](#)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) [2](#), [4](#)
31. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: CVPR (2023) [2](#)
32. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. URL <https://arxiv.org/abs/2205.11487> (2022) [2](#)
33. Shen, I.C., Liu, H.K., Chen, B.Y.: Nerf-in: Free-form nerf inpainting with rgb-d priors. Computer Graphics and Applications (CG&A) (2024) [4](#)
34. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021) [7](#)
35. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: WACV (2022) [2](#), [3](#), [4](#), [8](#), [9](#)
36. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM TOG (2023) [18](#)
37. Tang, L., Ruiz, N., Chu, Q., Li, Y., Holynski, A., Jacobs, D.E., Hariharan, B., Pritch, Y., Wadhwa, N., Aberman, K., et al.: Realfill: Reference-driven generation for authentic image completion. arXiv preprint arXiv:2309.16668 (2023) [8](#)

38. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics* (2023) [2](#)
39. Wang, D., Zhang, T., Abboud, A., Süsstrunk, S.: Inpaintnerf360: Text-guided 3d inpainting on unbounded neural radiance fields. *arXiv preprint arXiv:2305.15094* (2023) [2](#), [4](#), [9](#)
40. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: *ICCV* (2023) [7](#)
41. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: *CVPR* (2018) [6](#), [7](#)
42. Weder, S., Garcia-Hernando, G., Monszpart, A., Pollefeys, M., Brostow, G.J., Firman, M., Vicente, S.: Removing objects from neural radiance fields. In: *CVPR* (2023) [4](#)
43. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., et al.: Reconfusion: 3d reconstruction with diffusion priors. *arXiv preprint arXiv:2312.02981* (2023) [5](#)
44. Wynn, J., Turmukhambetov, D.: Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In: *CVPR* (2023) [5](#)
45. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: *CVPR* (2021) [5](#)
46. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *CVPR* (2018) [2](#), [8](#)
47. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: *ICCV* (2021) [3](#)
48. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG* (2018) [5](#), [11](#)
49. Zhu, J., Zhuang, P.: Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766* (2023) [7](#)

# Supplementary Material

## A More Visual Results

Please find more visual results and video renderings on our project page: <https://hubert0527.github.io/MALD-NeRF>.

## B Implementation Details

**NeRF.** We use a self-implemented NeRF framework similar to ZipNeRF [3] that uses hash-based [27] positional encoding along with multiple MLPs to predict the final density and RGB quantities. A scene contraction is applied to the NeRF [2] as all the scenes we experimented on are unbounded scenes. We use two proposal networks to perform importance sampling, followed by the main network. The network designs are similar to the Nerfacto implemented in the Nerfstudio [36].

**Hyperparameters.** For all experiments, we train the networks with 8 V100 GPUs for 30,000 iterations at a ray batch size of 16,384 using distributed data-parallel. The choice of batch size is constrained by the amount of GPU VRAM after loading the NeRF and other deep image priors, such as the latent diffusion model network for generative inpainting and the ZoeDepth model (NK version) for the depth loss. All these deep image priors are inferred without calculating gradients to reduce VRAM usage, and inference at a batch size of 1.

We use two separate optimizers for NeRF reconstruction and adversarial learning. The NeRF reconstruction uses an Adam optimizer with a learning rate decay from 0.01 to 0.0001, while adversarial learning uses an Adam optimizer with a learning rate 0.0001 throughout the training. Different from GANeRF [29], we found using RMSProp makes the training unstable. For the adversarial learning, we use a discriminator architecture similar to StyleGAN2 [20]. We train the discriminator with  $64 \times 64$  patches. We importance-sample  $256 \times 256$  image patches based on the number of inpainting pixels within the patch, then slice the image patch into the discriminator training patch and train the discriminator at a batch size of 16. For the importance sampling strategy, we first exclude patches with insufficient inpainting pixels (we empirically set the threshold to 50%). Assume that each patch index  $i$  contains  $d_i$  inpainting pixels, we assign a probability  $p_i = d_i / \sum_j d_j$  while sampling the patches for training.

As mentioned in Eq 4, Eq 5, and Eq 6 of the main paper, our networks are being trained with various loss terms. We balance the loss terms with  $\lambda_{\text{pix}} = 1$ ,  $\lambda_{\text{inter}} = 3$ ,  $\lambda_{\text{distort}} = 0.002$ ,  $\lambda_{\text{decay}} = 0.1$ ,  $\lambda_{\text{adv}} = 1$ ,  $\lambda_{\text{fm}} = 1$  and  $\lambda_{\text{GP}} = 15$ .

**Iterative Dataset Update.** We infer the latent diffusion model with a DDIM scheduler for 20 steps. During the iterative dataset update, we synchronize the random sampled image IDs across GPUs to ensure there is no overlap among GPUs, then update the 8 distinctly sampled images in the dataset with partial DDIM. For the partial DDIM, we first hard-blend the rendered pixels in the

inpainting mask region with the real pixels outside the inpainting region into a  $512 \times 512$  image, encode into the latent space with the auto-encoder of the latent diffusion model, then add the noise level at timestep  $t$  based on the current training progress and the HiFA scheduling. Therefore, as the training progresses, the final inpainted images will gradually converge to the current NeRF rendering results due to low noise levels. Since we update 8 images in each dataset update step, we set the frequency of iterative dataset update to one dataset update every 80 NeRF training steps, which is 8 times less frequent compared to InstructNeRF2NeRF [13]. The whole training approximately takes 16 hours on 8 V100 GPUs.