

RaFE: Generative Radiance Fields Restoration

Zhongkai Wu¹, Ziyu Wan², Jing Zhang¹, Jing Liao², and Dong Xu³

¹ College of Software, Beihang University, China

² City University of Hong Kong, China

³ The University of Hong Kong, China

ZhongkaiWu@buaa.edu.cn

zkaiwu.github.io/RaFE

Abstract. NeRF (Neural Radiance Fields) has demonstrated tremendous potential in novel view synthesis and 3D reconstruction, but its performance is sensitive to input image quality, which struggles to achieve high-fidelity rendering when provided with low-quality sparse input viewpoints. Previous methods for NeRF restoration are tailored for specific degradation type, ignoring the generality of restoration. To overcome this limitation, we propose a generic radiance fields restoration pipeline, named RaFE, which applies to various types of degradations, such as low resolution, blurriness, noise, compression artifacts, or their combinations. Our approach leverages the success of off-the-shelf 2D restoration methods to recover the multi-view images individually. Instead of reconstructing a blurred NeRF by averaging inconsistencies, we introduce a novel approach using Generative Adversarial Networks (GANs) for NeRF generation to better accommodate the geometric and appearance inconsistencies present in the multi-view images. Specifically, we adopt a two-level tri-plane architecture, where the coarse level remains fixed to represent the low-quality NeRF, and a fine-level residual tri-plane to be added to the coarse level is modeled as a distribution with GAN to capture potential variations in restoration. We validate RaFE on both synthetic and real cases for various restoration tasks, demonstrating superior performance in both quantitative and qualitative evaluations, surpassing other 3D restoration methods specific to single task. Please see our project website zkaiwu.github.io/RaFE.

Keywords: Neural Rendering · Generative Model · 3D Restoration · Neural Radiance Fields

1 Introduction

Recently, Neural Radiance Fields (NeRFs) [3, 4, 7, 12, 30, 31, 37, 40] have achieved great success in novel view synthesis and 3D reconstruction. However, most NeRF methods are designed based on well-captured images from multiple views with calibrated camera parameters. In real-world applications of NeRF, the data capture or transmission process often introduces various forms of image degradations, such as noise generated during photography in low-light conditions [28, 32]

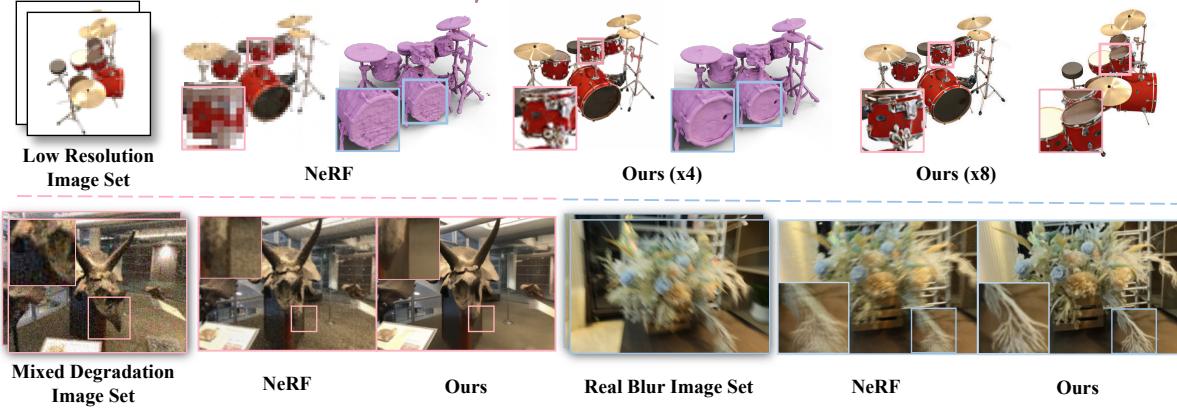


Fig. 1: High-quality restoration of radiance fields from various types of degradation. Given only degraded images, our method can restore high-quality NeRF. It is a generic approach that can be applied to various types of degradations, resulting in refinement of both geometry and appearance.

and blur caused by camera motion [26, 43], or JPEG compression and down-sampling during transmission [2, 42]. Simply restoring degraded images frame-by-frame can result in inconsistencies of geometry and appearance across different viewpoints. Directly reconstructing 3D models over these per-frame restoration results can easily induce inferior quality since current NeRF methods heavily rely on pixel-wise independent ray optimization with local computations, which are highly vulnerable to noise and other degradation.

Several NeRF variants have attempted to reconstruct 3D scenes with degraded multi-view images by introducing specific strategies or additional constraints for the optimization of radiance fields. For example, [19, 20, 26] deal with image blur artifacts by modeling the degradation kernel with NeRF, while [2, 42] super-sampling on rays or tri-planes to obtain high-resolution 3D from low-resolution observations. Additionally, [30] modifies NeRF to reconstruct the scene in linear HDR space by supervising directly on noisy raw images to address the noise generated in low-light conditions. [52] tries to improve the view synthesis quality by removing NeRF-specific rendering artifacts. Even with great success, all these approaches are only designed to handle specific types of degradation. To the best of our knowledge, currently there is no *generic pipeline* which supports the restoration of radiance fields under various types of degradation.

In this paper, we propose RaFE, a generic NeRF restoration framework that enables high-quality radiance fields reconstruction from captured images containing various types of degradation in a generative manner. Firstly, we leverage the success of off-the-shelf image restoration methods to restore the multi-view images with different forms of degradation, such as super-resolution, deblurring, denoising, removing compression artifacts or a combination thereof. It is important to note that since the images from different views are independently restored, there inevitably exist geometric and appearance inconsistencies between them. Naively optimizing a radiance field with these refined images would average out the inconsistencies and result in blurry outputs. To overcome this challenge, our insight here is, instead of describing a single 3D using inconsistent frames, we could consider these restored multi-view images as the renderings

from multiple distinct high-quality NeRF models with varied geometry and appearance. In this case, we abandon the commonly-used pixel-wise reconstruction objective and propose to leverage the generative adversarial networks (GANs) to model the distribution of these different high-quality NeRF models, which could effectively capture the inherent variability in ill-posed inverse problem, allowing for a better accommodation of the inconsistencies present in different views.

Specifically, our pipeline consists of two main stages. In the first stage, based on the type of degradation, we can employ the corresponding off-the-shelf image restoration methods [17, 25, 33, 34, 45, 48] to obtain a set of high-quality multi-view images. In practice, we prefer choosing restoration methods which have strong capabilities to recover high-quality and realistic texture details. In the second stage, we train a 3D generative model based on these restored multi-view images. Drawing inspirations from recent 3D generation works [6, 9, 36, 39], we construct a convolutional neural network (CNN) to generate tri-plane features, which are subsequently sampled and decoded into density and colors using MLP networks for NeRF rendering. Here, instead of generating single-level tri-plane features as previous works did, we decompose the tri-planes into two levels. The coarse-level tri-planes are constructed directly from low-quality images and remain fixed during training, representing the coarse structure of the modeled 3D distribution. Simultaneously, we train a generator to output the diverse fine-level tri-plane features, which act as residuals to be added to the coarse-level features for NeRF rendering. By focusing on learning the residual representations instead of the entire tri-planes for NeRF, we simplify the modeling and learning of restoration variations since we only need to learn the details while the coarse structure is provided by coarse-level tri-planes, which makes great improvement in rendering quality for more complex regions. To train the generator, we adopt an adversarial loss defined on NeRF rendered 2D images to encourage them to be indistinguishable from the high-quality restored images. We also incorporate a perceptual loss between the rendered images and the restored images to calculate structure constraints. Additionally, we propose patch sampling strategies to stabilize the generator training procedure. Once the generator has been trained, we can generate restored radiance fields with high quality renderings and a certain level of diversity by sampling different code in the latent space.

We conducted extensive experiments to validate the effectiveness of our method, both qualitatively and quantitatively. The experimental results showcase the superiority of our approach in various restoration tasks, such as super-resolution (upper row of Figure 1), camera motion blur (a real-world case at the right part of the lower row of Figure 1) and the restoration of mixed degradation consisting of noise, blur, and compression (left part of lower row of Figure 1). Our method not only generates images with richer and enhanced texture details but also achieves significant improvements in geometric refinement, as demonstrated by the mesh visualization in Figure 1. To summarize, our contributions are:

- We propose a generic radiance fields restoration pipeline that is applicable to various types of degradation.

- We introduce a generative method for NeRF restoration that enables better accommodation of geometric and appearance inconsistencies present in the multi-view images, thus allowing us to incorporate the success of image restoration into 3D restoration.
- We show the restoration method performs well on various degradation scenarios with both enhanced appearance and refined geometry.

2 Related Works

2.1 2D image restoration

Image restoration is a long standing problem in low-level vision domain and significant progress has been achieved in different specific tasks including image super-resolution, deblur, denoise and blind restoration. Previously reconstruction-based methods [11, 21, 23, 51, 53] show their success in these tasks. However, those reconstruction-based method are struggling to generate abundant high-quality details. Subsequently, generative restoration methods [10, 17, 25, 33, 34, 38, 41, 45, 48], particularly those based on diffusion model, have shown the great capability to render high-quality details. Deepfloyd [34] proposes a super-resolution model, which concatenates the low-resolution input with random noise at pixel level as a condition to guide the generation of high-resolution images. For blind restoration, DiffBIR [25] designs a degraded pipeline to simulate real-world degradation and utilizes the pre-trained diffusion model to generate photorealistic images. For camera motion blur, HiDiff [10] recovers exquisite images by using diffusion to generate feature with abundant detailed information.

2.2 Radiance Fields Restoration

NeRF restoration aims to reconstruct high-quality NeRF given only degraded images with various artifacts such as blur, noise, or low resolution. Up to now, several works [2, 8, 14, 19, 20, 26, 30, 32, 42, 43, 52] have explored this task under specific type of degradation. [20, 26, 43] deal with blurred input images by designing blur kernel or optimizing camera paths for NeRF rendering process. For NeRF super-resolution, [42] increased the ray sampling density, forcing multiple rays to render pixels equal to the same pixel, and applied a 2D refinement model to get final output images. [14] introduces a CEM [1] refinement model to adjust the output of off-line super-resolution model for better multi-view consistency. However, the CEM refinement model ruins the structure details of the images and inconsistencies still exist, leading to a smooth reconstruction result. [30, 32] mainly focus on the noise degradation of the input image. [30] modifies NeRF to reconstruct the scene in linear HDR color space by supervising directly on noisy raw input images to address the noise generated during post-processing from HDR images to LDR images, while [32] achieve NeRF restoration on noisy images by using noise-awarded encode to aggregate features across views. [52] considers solving the degradation that occurs in NeRF reconstruction by training a 2D refinement model using simulated degraded images for typical NeRF-style artifacts, but they can not achieve refinement on geometry since the restoration

only happens on rendered views. None of the existing approaches can restore NeRF in 3D space directly with flexible forms of degradation. NVSR [2] archives 3D geometry refine by upsampling tri-plane representation, but their training processing requires tremendous amounts of 3D data, which are extremely hard to obtain in practice. By contrast, our method has the ability to handle more flexible forms of degradation and restore 3D geometry and appearance with the only needs of an image set for an object or scene, making the 3D restoration more practical in real-world applications.

3 Method

In this section, we will elaborate the details of RaFE. We introduce how to refine the degraded views using pretrained 2D restoration model, to capture the high-quality appearance distribution in Sec. 3.1. Then, we describe our generative restoration framework including the neural representation, generator architecture and optimization in Sec. 3.2. The training strategy is introduced in Sec. 3.3. The overall pipeline could be found in Fig. 2.

3.1 High-quality Image Restoration

The recent success on 2D image restoration task is dominated by the denoising diffusion probabilistic model, benefiting from its powerful appearance generation capability and prior. Hence in this paper we mainly borrow the recent diffusion-based restoration methods [25, 34] to obtain the high-quality multi-view images from their low-quality counterparts. It should be noted that our 3D restoration framework could also work pretty well based on non-diffusion restoration approaches. Please check the experimental parts for more details.

To assist in the restoration of image details, we employ powerful image caption models [5, 22] to get an accurate textual prompt for the scene, denoted as P . Given a set of multi-view low-quality images I_l , the textual prompt is produced by selecting a view that contains as much information of the scene as possible, e.g. a side view of a synthetic Lego model. The prompt P and low-quality images I_l are then fed into the restoration model [25, 34] to achieve high-quality per-frame refinement. We denote the restored high-quality images as I_h .

The diffusion model-based image restoration methods can effectively reconstruct high-quality images from various real-world degradation types. Moreover, thanks to the powerful generalization ability of diffusion models preserved in these restoration methods, we can handle the scene restoration in open domain without the needs of re-training. However, though the restored images have better quality, more variations among different views occur due to the generative nature of these restoration models, leading to serious multi-view inconsistencies. Next, we focus on dealing with this issue with the proposed generative pipeline.

3.2 Generative NeRF Restoration

To deal with geometric and appearance inconsistencies across views, we treat these restored multi-view images distributed similarly to the rendered images

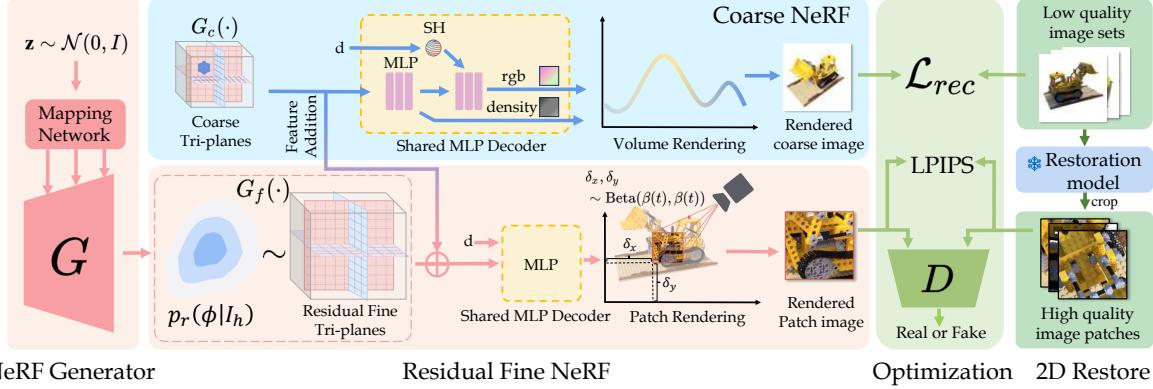


Fig. 2: Overview of our pipeline. Given multi-view degraded images, we utilize the off-the-shelf methods to restore high-quality multi-view images. Then, we train our Generative Restoration NeRF to generate a high-quality scene.

from multiple slightly different high-quality NeRF models. Consequently, instead of directly fitting a NeRF model using the refined high-quality views which usually leads to blurry reconstruction, we are trying to learn the distribution of these diverse NeRF models by leveraging a generative method, allowing us to sample distinct restored 3D under the same degraded inputs.

3D Representation. Following the recent 3D generative model [6, 9, 36], we adopt the hybrid explicit-implicit tri-plane representation for the feature field. This representation combines both explicit and implicit components to effectively model the density and RGB values. More specifically, to obtain the descriptor for any query location $\mathbf{x} \in \mathbb{R}^3$, we project the point onto three planes (P_{xy}, P_{yz}, P_{zx}) to retrieve the corresponding features (f_{xy}, f_{yz}, f_{zx}) using interpolation. Then we calculate the mean value of these three sampled features to obtain the final feature representation.

Once we obtain the features for a point along the ray, we use two MLPs to decode the density and RGB values. The first MLP, denoted as \mathcal{M}_{dens} , maps the features to the density value $\sigma \in \mathbb{R}$ and a color feature \mathbf{f}_{color} . The second MLP, denoted as \mathcal{M}_{color} , takes the color feature and the view direction as inputs and maps them to the RGB value $\mathbf{c} \in \mathbb{R}^3$. We empirically show that incorporating the view direction allows us to effectively model view-dependent effects, particularly when there are non-Lambertian surfaces in the scene. After obtaining the densities and RGB values for each sampled point along the emitted ray, we can apply the classic volumetric rendering to obtain the color value for each pixel:

$$C(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \quad (1)$$

where \mathbf{c}_i, σ_i represents RGB and density value of the i^{th} point along the ray, respectively. N is the sampled number and δ_i is the distance between samples.

NeRF Generator. Given restored multi-view images, we regard them as renderings from several diverse high-quality NeRF models with varied geometry and appearance. The distribution formed by these distinct NeRF models, formulated as $p_r(\phi|I_h)$, can be modeled by a generator. With the hybrid explicit-implicit tri-plane representation, we introduce a StyleGAN2 [15]-like CNN-based generator, which receives a latent code \mathbf{w} mapped from random code \mathbf{z} to generate fine

tri-planes \mathbf{P}_f for high-quality NeRF rendering. To fully leverage the degraded images and make the generator only focus on generating the necessary refinements, we also pre-train a coarse NeRF model with tri-plane features denoted as P_c using the input low-quality images. Specifically, the coarse NeRF model consists of a coarse tri-plane \mathbf{P}_c features and a decoder (where the decoder of coarse tri-planes is shared with the one of fine tri-planes). To obtain the final refined tri-planes P , we combine the reconstructed tri-plane representations with the residual tri-planes generated by the generator via: $\mathbf{f} = \mathbf{f}_c + \mathbf{f}_f$. By merging these components, RaFE effectively captures both the global geometric guidance provided by the coarse NeRF and the local refinements learned from 3D generator, enabling us to obtain the restored tri-planes that exhibit enhanced geometric accuracy and appearance fidelity.

Optimization. To supervise the generator and NeRF parameters, we propose to minimize the distribution discrepancy between the rendered images and the restored high-quality images. We adopt a saturate GAN [13] loss with an image level discriminator. Specifically, we treat the high-quality images restored by 2D model as the real samples while the rendered images as the fake samples, and utilize adversarial loss with R1 regularization between the real and fake images:

$$\begin{aligned}\mathcal{L}_{adv} = & \mathbb{E}_{\mathbf{z} \sim p_z, \boldsymbol{\theta} \sim p_{\theta}}[f(D(G(\mathbf{z}, \boldsymbol{\theta})))] \\ & + \mathbb{E}_{I_h \sim p_h}[f(-D(I_h)) + \lambda \|\nabla D(I_h)\|^2],\end{aligned}\quad (2)$$

where $\mathbf{z}, \boldsymbol{\theta}$ represent random code and view point, respectively, and I_h is the restored high quality image.

However, we observed relying solely on a GAN loss for training can lead to significant geometric mismatches between the restored images and rendered views. We argue that although the GAN loss helps align the distribution of 2D renderings, it still lacks geometry-level constraints. Therefore, we also incorporate a perceptual loss that encourages the rendered images to resemble the geometry of pre-frame restoration.

$$\mathcal{L}_{geometry} = LPIPS(I_h, G(\mathbf{z}, \boldsymbol{\theta})), \quad (3)$$

where $LPIPS(\cdot, \cdot)$ refer to the learned perceptual image patch similarity proposed in [49]. I_h is the restored high quality image paired with view point $\boldsymbol{\theta}$, and $G(\mathbf{z}, \boldsymbol{\theta})$ are the rendered image using the same training view $\boldsymbol{\theta}$ and a random sampled latent code \mathbf{z} . We also supervise the coarse NeRF branch by the input RGB degraded images:

$$\mathcal{L}_{rec} = \mathbb{E}_{\boldsymbol{\theta} \sim p_{\theta}}[\|G_c(\boldsymbol{\theta}) - I_l^{\boldsymbol{\theta}}\|^2], \quad (4)$$

where p_{θ} indicate the view point distribution, and $I_l^{\boldsymbol{\theta}}$ is the low quality image corresponding to view point $\boldsymbol{\theta}$. Overall, the complete training objective is:

$$\mathcal{L} = \lambda_{geometry} \mathcal{L}_{geometry} + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{rec} \mathcal{L}_{rec}, \quad (5)$$

where $\lambda_{geometry}, \lambda_{adv}, \lambda_{rec}$ are trade-off parameters.

3.3 Patch-based Training Strategy

During the training process, fully rendering the entire image, such as 256^2 or 512^2 pixels, and then feeding it into the discriminator can be computationally

intensive and resource-consuming. This is because volume rendering requires the computation of density and color values for sampling points along the ray for each pixel, which can become prohibitively expensive for a whole image. Therefore, we only render a patch of the view at a time (i.e. 64^2). The rendered patches and high-quality image patches are randomly selected, and the discriminator in Eq. 2 receives patches of rendered images and patches of high-quality images as fake images and real images respectively. For the perceptual loss in Eq. 3, the rendered patches and the high-quality patches have the same spacial coordinate on their original images to guarantee that the cropped patches have the same semantic and local structure.

One limitation of using patches instead of entire images for training is that it can be challenging to sample the local scenes evenly at the boundary regions of images, particularly for forward-facing data. The imbalanced training distribution will result in mode collapse during the training process. To mitigate this issue, we employ a beta sampling strategy to determine the positions of the sampled patches. This strategy ensures that patches in the boundary regions of the image are adequately sampled. More specifically, the beta sampling can be formulated as:

$$\delta_x, \delta_y \sim Beta(\beta(t), \beta(t)), \quad (6)$$

where δ_x and δ_y denote the position offset in x and y directions respectively, while $\beta(t)$ are linearly annealed from $\beta(0) = 1$ to some final value $\beta(T)$ smaller than 1. By using the beta sampling strategy, we can maintain a more balanced distribution of training data that focuses more on the boundary patches, alleviating mode collapse issue and improving the overall training stability.

4 Experiments

4.1 Setup

Datasets. We evaluate our model on the NeRF-Synthetic benchmark dataset [30], which contains 8 synthetic objects with images taken from different viewpoints uniformly distributed in the hemisphere. Following original setting, We hold 200 viewpoints for generating high-quality training data and 200 viewpoints for testing. Further, to demonstrate the generalization ability of our method, we also evaluate our method on complex real-world LLFF scenes [29] which consists of 8 scenes captured with roughly forward-facing images. We also demonstrate the superior performance of RaFE on real-world blur [26] and noise [28] data.

Evaluation Metrics. Following the common practice of 3D reconstruction, we firstly try to evaluate each method with two standard image quality metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [46], which however could not reflect the real 3D restoration performance according to our observations. Due to the generative characteristic of RaFE, the recovered radiance fields from RaFE is very high-quality, but may not faithfully follow the "ground-truth" 3D, since inverting a degraded signal is a highly ill-posed problem. Moreover, we also found the baselines with better scores over PSNR and SSIM still pose the degraded appearance with smooth texture details, as shown in

Figure 3b. Hence, a better way is to leveraging perceptual metric: learned perceptual image patch similarity (LPIPS) [49] which computes the mean squared error (MSE) between normalized features from all layers of a pre-trained VGG [35] encoder and is deemed to better correlate with human perception. Besides we also leverage the latest non-reference based image quality assessment metrics including LIQE [50] and MANIQA [47] to demonstrate the superior rendering quality of our method.

Implementation Details. We implement all the experiments by PyTorch. For the 2D generator and discriminator, we use a convolutional-based generator and discriminator used in StyleGAN2 [16]. In all experiments, we choose Adam optimizer for all the modules in our pipeline, with hyperparameters $\beta_1 = 0, \beta_2 = 0.99$. We use learning rate 2×10^{-3} for both generator and discriminator. For loss weights, we use $\lambda_{mimic} = 0.5$, $\lambda_{adv} = 1.0$, and $\lambda_{rec} = 1.0$ for almost all experiments. We evaluate RaFE framework on 4 different 3D restoration tasks:

- 4× Super-Resolution: On the blender dataset, we resize the image to 64×64 resolutions to get low-resolution images. As for LLFF data, we first center crop the training image to 188×252 to adapt to our 4× super-resolution task and then resize to 47×63 to get low-resolution images. And we use Deepfloyd [34] for 2D super-resolution.
- Deblur: On the LLFF dataset, we construct camera motion blur by applying the blur kernel following equation $I_{blur} = \mathbf{A} * I_{clear}$, where \mathbf{A} is the blur kernel and $*$ stands for the convolution operator. Different from the blur dataset proposed in Deblur-NeRF [26], which contains blurred images with varying degrees of blur, and even includes a certain amount of high-resolution images, we apply a large blur kernel size (e.g. 13) and a more complex camera motion path to all the images in the dataset. We additionally construct consistent blur datasets by using the same blur kernel to the training image in a scene, which means the camera motion trajectory is the same for the training image set. The resolution of images is the same as the super-resolution task. We use HiDiff [10] to recover high-quality images.
- Denoise: On the LLFF dataset, we follow the noise model used in [27, 32] and we get the noisy version of a clean image I according to equation $I_{noisy}(x) = \mathcal{N}(I(x), \delta_r^2 + \delta_s^2 I^2(x))$, where σ_r is the signal-independent read-noise parameter, σ_s is the signal-dependent shot-noise. Following [27, 32], we use Gain level to represent the noise strength. We use gain levels = 8 to get our noisy image. We use DiffBIR [25] to get high-quality images.
- Mixed degradation: The degradation pipeline consists of three stages: **blur**, **noise**, and **JPEG compression**. First, we utilize Gaussian blur with a radius of 7 for the blender dataset and 3 for the LLFF dataset. Second, we add noise with std 25. And then, we apply JPEG compression. The quality of JPEG compression is 50. The resolution of images is the same as the super-resolution task. We use DiffBIR [25] to get high-quality images.

4.2 Results

Baseline Methods. For general tasks, we try a baseline that firstly restores the degraded images and uses the restored high-quality images to reconstruct a NeRF directly, denoted as NeRF-Perframe. We also use the 2D-based restoration model SwinIR [24] to do the per-view refinement for the renderings of NeRF trained by degraded image, denoted as NeRF-SwinIR. Note that we do not evaluate NeRF-SwinIR for the deblur task since there are no corresponding checkpoints.

To more thoroughly test the effectiveness of our method, we also select some task-specific competitors. For super-resolution task, we choose NeRF-SR [42], Neural Volume Super-Resolution (NVSR) [1] as baselines. For mixed degradation, since there is no existing method tailored for mixed degradation, we choose NeRFLiX, which tries to solve the NeRF-like degradation by training a 2D refinement model using degradation images constructed by a degradation simulator for typical NeRF-style artifacts. We consider this to be the most relevant method. For the deblur task, we compare with two state-of-the-art methods Deblur-NeRF [26] and BAD-NeRF [44]. Deblur-NeRF designs a learnable blur kernel and applies it to rays to simulate the degrading process and BAD-NeRF directly models the camera trajectories to solve motion blur. For the denoising task, we compare with NAN [32], which uses a noise-aware encoder to aggregate the feature of multi-view images for restoration.

Quantitative Results. We conduct extensive quantitative comparisons with various baselines across different restoration task in Tab. 1a for super-resolution, Tab. 2b and Tab. 2a for deblurring, Tab. 2c for denoising and Tab. 1b for mixed degradation. As analyzed before, although most of the time our method falls slightly behind on the reconstruction metrics like PSNR and SSIM compared with other baselines, which only measure local pixel-aligned similarity between the rendered novel views and the ground truth images, are less indicative since uncertainties naturally exist in generative procedure. Taking the super-resolution results on Blender data as an example (Tab. 1a), the simplest baseline NeRF-Perframe have already achieved the best reconstruction metrics, but as shown in Figure. 3a, its visual quality is vastly inferior compared with our results. Through the error map, we found the mis-alignment between the generated 3D and input 3D causes the drop of PSNR and SSIM. By contrast, on the perceptual metrics like LPIPS and non-reference based metrics including LIPE and MANIQA, which could more effectively reflect the restoration performance, our method consistently achieves better results when compared with other baselines, demonstrating its clear advantages.

For mixed degradation tasks, the best results for LPIPS metrics are achieved by NeRFLix w. ref [52]. This is because NeRFLiX can see two high-quality ground-truth images from the nearest two viewpoints when inference, which leads to information leakage. However, high-quality ground-truth information is not accessed in our setting or any real-world cases. After eliminating the impact of ground truth (NeRFLiX w/o. ref) by replacing the reference ground truth images with degraded images, our method performs better than NeRFLiX.

Method	Blender					LLFF					Method	Blender					LLFF				
	PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑	PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑		PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑	PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑
NeRF-SR	26.85	0.912	0.135	2.372	0.366	22.65	0.702	0.327	1.320	0.220	NeRFlix W. Ref	27.31	0.933	0.066	2.643	0.358	28.18	0.885	0.145	1.793	0.233
NVSР	26.13	0.879	0.132	2.301	0.343	21.29	0.606	0.442	2.108	0.177	NeRFlix W/O. Ref	25.78	0.905	0.107	1.102	0.166	26.28	0.821	0.290	1.129	0.183
NeRF-SwinIR	24.07	0.878	0.119	3.06	0.381	21.73	0.655	0.365	1.808	0.267	NeRF-SwinIR	27.42	0.922	0.086	2.441	0.317	25.94	0.813	0.249	2.013	0.193
NeRF-Perframe	27.39	0.922	0.083	3.276	0.386	23.68	0.745	0.261	1.417	0.257	NeRF-Perframe	26.79	0.927	0.088	2.289	0.355	24.45	0.812	0.267	1.41	0.257
Ours	24.99	0.901	0.062	4.621	0.543	23.85	0.752	0.197	2.397	0.322	Ours	25.28	0.907	0.076	3.947	0.541	24.81	0.832	0.217	2.210	0.296

(a) Super-resolution

(b) Mixed degradation

Table 1: Quantitative comparisons on super-resolution and mixed degradation tasks. The best result without using reference is highlighted. Our method achieves great superiorities on perceptual metrics and image quality when compared with others.

Method	LLFF					Method	LLFF					Method	LLFF				
	PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑		PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑		PSNR↑	SSIM↑	LPIPS↓	LIQE↑	MANIQA↑
Deblur-NeRF	23.75	0.799	0.307	1.365	0.157	Deblur-NeRF	21.71	0.749	0.333	1.104	0.122	NAN	25.99	0.822	0.3208	1.3032	0.241
BAD-NeRF	24.027	0.788	0.313	1.094	0.151	BAD-NeRF	25.40	0.836	0.278	1.140	0.173	NeRF-SwinIR	24.37	0.778	0.346	1.579	0.210
NeRF-Perframe	21.02	0.695	0.362	0.362	0.150	NeRF-Perframe	21.57	0.771	0.310	1.103	0.199	NeRF-Perframe	23.66	0.786	0.281	1.095	0.214
Ours	23.23	0.811	0.294	1.144	0.177	Ours	22.93	0.789	0.252	1.143	0.224	Ours	23.78	0.791	0.2561	1.607	0.257

(a) Deblurring

(b) Deblurring (consistent blur)

(c) Denoising with Gain 8

Table 2: Quantitative comparisons for deblurring and denoising. The best result without using reference is highlighted. Our method performs the best on perceptual metrics.

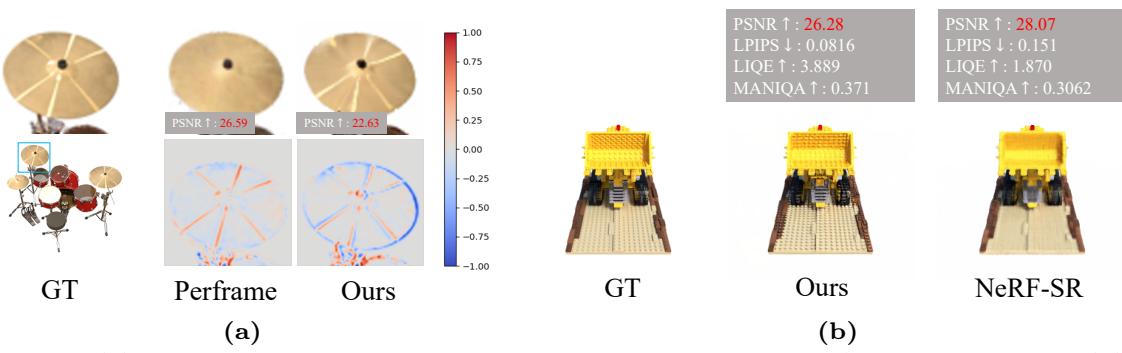


Fig. 3: (a) Error map between ground truth and our method/NeRF-Perframe. (b) we showcase that the visual quality can be much better even with lower PSNR scores.

Qualitative Results. To further verify the restoration capability, we present visual results for different degradations in Fig. 4. Our method is able to generate realistic details while other methods tend to generate smooth results which lack high-quality details. For example, for the super-resolution task, we show the drum restoration for each method, as can be seen in Fig. 4, our method generates high-fidelity drum surfaces with sharp edges, while other methods suffer from severe blur on the drum surfaces and edges. For mixed degradation tasks, our method successfully restores the intricate golden textures on the chair surface while other methods only rendered extremely blurred texture.

Geometry Refinement. Compared with previous NeRF restoration methods, most of which focusing on using 2D refinement to resolve frame-wise defects, our method firstly leverages the general priors of large foundation models and GAN, achieving the open-domain 3D-based restoration. Thus, our restoration algorithm could not only recover the high-quality and view-consistent images, but also refine the 3D geometry. We demonstrate this advantage using the Ficus data of Blender, as shown in Fig. 5a, through comparing the extracted mesh from the restored NeRF models, our method effectively learns the restoration in the 3D tri-plane space and recover better geometry.

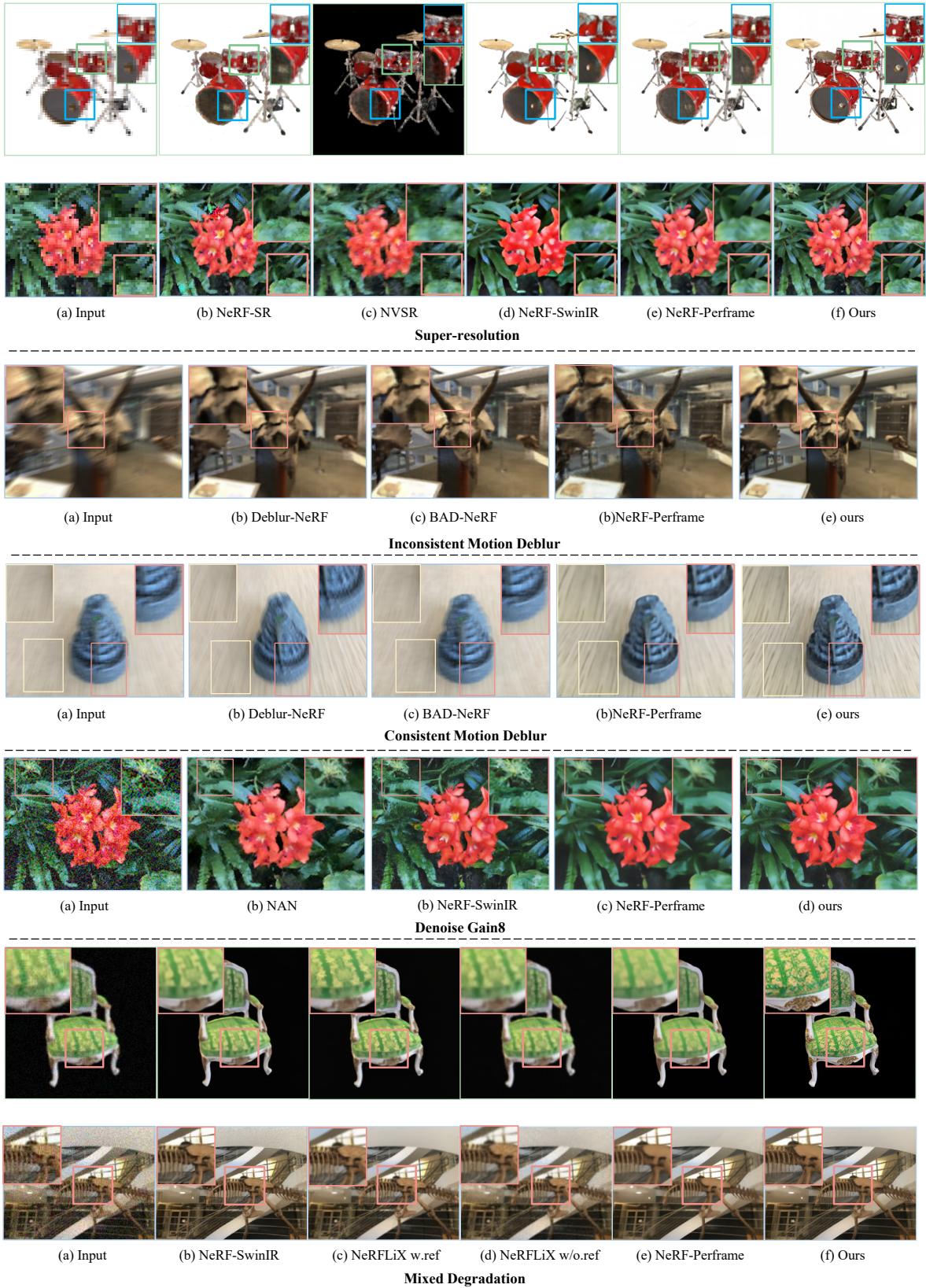


Fig. 4: Visual results for super-resolution and mixed-degradation tasks. We show that our method is capable of generating detailed geometry and texture while other methods tend to be smooth in both geometry and texture. We recommend zooming in for better visualization.



Fig. 5: (a) Geometry comparisons between NeRF-SWINIR/NeRFLIX and our method. (b) Comparisons between using different 2D restoration models.

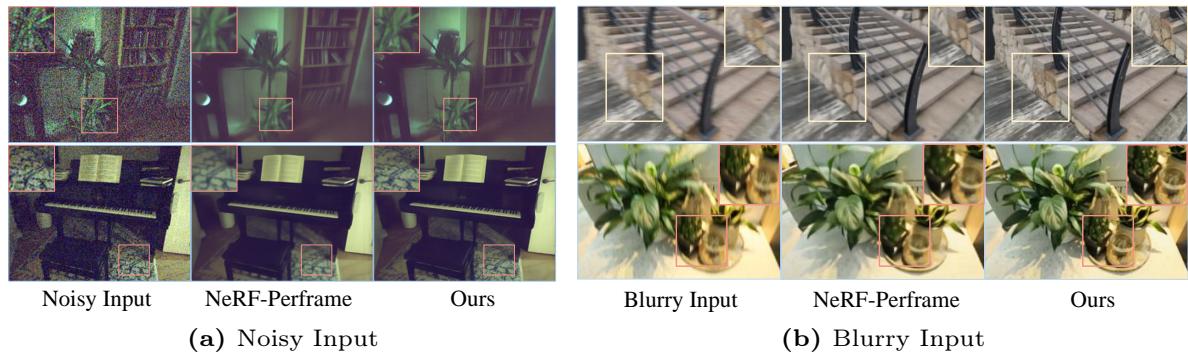


Fig. 6: The performance of our method under real world scenario. The results indicate that our method could also generalize to the real-world setting pretty well.

4.3 Real-world Restoration

To validate that our method also works well on the real-world setting, we also test RaFE using real-noise and real-blur datasets proposed in [28] and [26]. As shown in Fig. 6a and Fig. 6b, benefiting from the powerful 2D restoration models, NeRF-Perframe could effectively remove the existing degradations like noise or blur. Nonetheless, simply averaging the view-inconsistent 2D restored frames results in very smooth 3D reconstruction. By contrast, through modeling the 3D space using a generative model, the sampled 3D model from our method could achieve significantly better rendering quality with realistic and degradation-free texture details, demonstrating its superiorities on the real-world 3D restoration.

4.4 Discussion

Effects of different restoration models. To investigate the influence of different 2D restoration models on our method, we tested two additional off-the-shelf restoration models for the super-resolution task, including diffusion-based DiffBIR [25] and non-diffusion-based SRFormer [53]. As shown in Fig. 5b, diffusion-based model DiffBIR shows larger restoration diversity over SRFormer as expected by measuring the diversity score. When the repaired images exhibit diversity, direct reconstruction inevitably leads to the blurriness to varying degrees due to the existing multi-view inconsistency. By contrast, through modeling the distribution of the potential high-quality NeRFs, our method successfully accommodates these inconsistencies and consistently achieves better performance over NeRF-Perframe, demonstrating the great generalization capability of RaFE to different 2D restoration model.

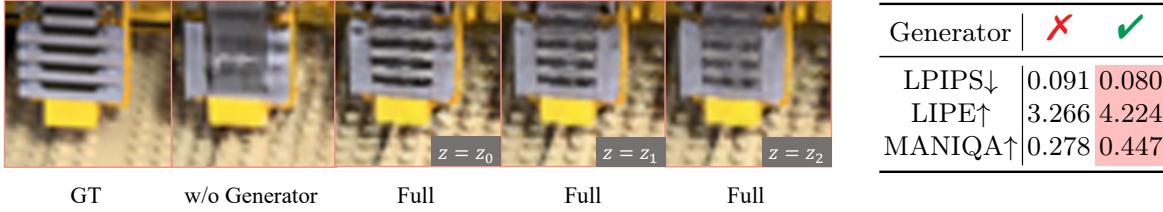


Fig. 7: Effectiveness of the tri-plane generator. **Left:** image rendered by NeRF without generator and images rendered by generative NeRF under different random codes $z(z_0, z_1, z_2)$. **Right:** numerical metrics to evaluate the efficacy of the generator. The results show the effectiveness of using the generator to model the distribution.

Effects of the generator. In this ablation study, we examine the influence of the generator by comparing with the baseline that directly optimizes the NeRF parameters using GAN loss and LPIPS mentioned above on the Lego dataset. As we can observe in Fig. 7, the image rendered by generative NeRF exhibits varied fine-textured details under different random code z . Once the generator is removed, the rendered images will contain blurry and smooth appearance, showing the importance of using generator to model the distribution, which can also be demonstrated by the numerical metrics in the right part of Fig. 7.

Effects of geometry loss & GAN loss. In this experiment, we examine the influence of geometry and GAN losses on the performance by training the model on the drums dataset, shown in Fig. 8. We conduct this experiment by setting the weights of unused losses to 0. As can be seen, removing the geometry loss results in severe geometry mismatches (e.g. the edges of the drums exhibit distortion), which can be also demonstrated by the drop of PSNR. Meanwhile, using perceptual loss only makes the rendered image to be very smooth with fewer details (e.g. the light and reflection), resulting in a significant decline on perceptual metrics. RaFE trained with both objectives achieves the best performance.

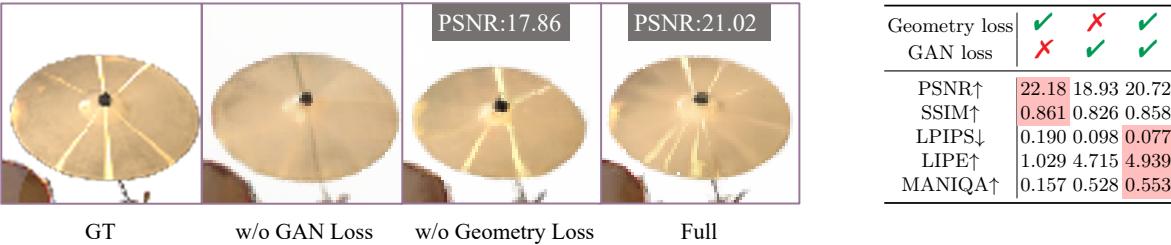


Fig. 8: Effectiveness of geometry loss & GAN loss term. We show the visual results (left) and numerical metrics (right) of using different loss terms. RaFE trained with both objectives achieves the best performance.

5 Conclusions

This paper proposes a novel generic NeRF restoration method that applies to various types of degradations, such as low resolution, blurriness, noise, and mixed degradation. The proposed method leverages the off-the-shelf image restoration methods to restore the multi-view input images individually. To tackle the geometric and appearance inconsistencies presented in multi-view images due to individual restoration, we propose to train a GAN for NeRF generation, where a two-level tri-plane structure is adopted. The coarse-level tri-plane pre-trained by low-quality images remains fixed, while the fine-level residual tri-plane to be

added to the coarse level is modeled by a GAN-based generator to capture variations in restoration. Extensive experiments on various restoration tasks with both synthetic and real cases demonstrate the superior performance of our method.

Limitations and Future Work: One limitation of our method is instability when performing the restoration at extremely high resolutions, such as 4k, due to the patch-rendering strategy. Moreover, due to the inherent slow efficiency of NeRF rendering, currently the long training time also needs to be optimized. To overcome these limitations, a potential solution would be to integrate more efficient rendering techniques like Gaussian splatting [18] into RaFE, enabling the rendering of entire images instead of using patches in real-time. We plan to resolve these issues in the future work.

References

1. Bahat, Y., Michaeli, T.: Explorable super resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2716–2725 (2020) [4](#), [10](#)
2. Bahat, Y., Zhang, Y., Sommerhoff, H., Kolb, A., Heide, F.: Neural volume super-resolution. arXiv preprint arXiv:2212.04666 (2022) [2](#), [4](#), [5](#)
3. 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 (2021) [1](#)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022) [1](#)
5. Beyer, L., Zhai, X., Kolesnikov, A.: Big vision. https://github.com/google-research/big_vision (2022) [5](#)
6. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16123–16133 (2022) [3](#), [6](#)
7. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorrf: Tensorial radiance fields. In: European Conference on Computer Vision (ECCV) (2022) [1](#)
8. Chen, W.T., Yifan, W., Kuo, S.Y., Wetzstein, G.: Dehazenerf: Multiple image haze removal and 3d shape reconstruction using neural radiance fields. arXiv preprint arXiv:2303.11364 (2023) [4](#)
9. Chen, X., Deng, Y., Wang, B.: Mimic3d: Thriving 3d-aware gans via 3d-to-2d imitation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) [3](#), [6](#)
10. Chen, Z., Zhang, Y., Ding, L., Bin, X., Gu, J., Kong, L., Yuan, X.: Hierarchical integration diffusion model for realistic image deblurring. In: NeurIPS (2023) [4](#), [9](#)
11. Chen, Z., Zhang, Y., Gu, J., Kong, L., Yang, X., Yu, F.: Dual aggregation transformer for image super-resolution. In: ICCV (2023) [4](#)
12. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023) [1](#)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) [7](#)

14. Han, Y., Yu, T., Yu, X., Wang, Y., Dai, Q.: Super-nerf: View-consistent detail generation for nerf super-resolution. arXiv preprint arXiv:2304.13518 (2023) [4](#)
15. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021) [6](#)
16. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proc. CVPR (2020) [9](#)
17. Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising diffusion restoration models. In: Advances in Neural Information Processing Systems (2022) [3](#), [4](#)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> [15](#)
19. Lee, D., Lee, M., Shin, C., Lee, S.: Dp-nerf: Deblurred neural radiance field with physical scene priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12386–12396 (2023) [2](#), [4](#)
20. Lee, D., Oh, J., Rim, J., Cho, S., Lee, K.M.: Exblurf: Efficient radiance fields for extreme motion blurred images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17639–17648 (2023) [2](#), [4](#)
21. Li, H., Zhang, Z., Jiang, T., Luo, P., Feng, H., Xu, Z.: Real-world deep local motion deblurring. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1314–1322 (2023) [4](#)
22. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022) [5](#)
23. Li, J., Zhang, Z., Liu, X., Feng, C., Wang, X., Lei, L., Zuo, W.: Spatially adaptive self-supervised learning for real-world image denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9914–9924 (June 2023) [4](#)
24. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. arXiv preprint arXiv:2108.10257 (2021) [10](#)
25. Lin, X., He, J., Chen, Z., Lyu, Z., Fei, B., Dai, B., Ouyang, W., Qiao, Y., Dong, C.: Diffbir: Towards blind image restoration with generative diffusion prior. arXiv preprint arXiv:2308.15070 (2023) [3](#), [4](#), [5](#), [9](#), [13](#)
26. Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblurnerf: Neural radiance fields from blurry images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12861–12870 (2022) [2](#), [4](#), [8](#), [9](#), [10](#), [13](#)
27. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2502–2510 (2018) [9](#)
28. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: NeRF in the dark: High dynamic range view synthesis from noisy raw images. CVPR (2022) [1](#), [8](#), [13](#)
29. 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 Transactions on Graphics (TOG) (2019) [8](#)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) [1](#), [2](#), [4](#), [8](#)
31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127> [1](#)

32. Pearl, N., Treibitz, T., Korman, S.: Nan: Noise-aware nerfs for burst-denoising. In: CVPR (2022) [1](#), [4](#), [9](#), [10](#)
33. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–10 (2022) [3](#), [4](#)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022) [3](#), [4](#), [5](#), [9](#)
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) [9](#)
36. Skorokhodov, I., Tulyakov, S., Wang, Y., Wonka, P.: Epigraf: Rethinking training of 3d gans. Advances in Neural Information Processing Systems **35**, 24487–24501 (2022) [3](#), [6](#)
37. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022) [1](#)
38. Tian, K., Jiang, Y., Yuan, Z., Bingyue, P., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. arXiv preprint arXiv:2404.02905 (2024) [4](#)
39. Wan, Z., Paschalidou, D., Huang, I., Liu, H., Shen, B., Xiang, X., Liao, J., Guibas, L.: Cad: Photorealistic 3d generation via adversarial distillation. arXiv preprint arXiv:2312.06663 (2023) [3](#)
40. Wan, Z., Richardt, C., Božič, A., Li, C., Rengarajan, V., Nam, S., Xiang, X., Li, T., Zhu, B., Ranjan, R., Liao, J.: Learning neural duplex radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8307–8316 (June 2023) [1](#)
41. Wan, Z., Zhang, B., Chen, D., Zhang, P., Chen, D., Liao, J., Wen, F.: Bringing old photos back to life. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2747–2757 (2020) [4](#)
42. Wang, C., Wu, X., Guo, Y.C., Zhang, S.H., Tai, Y.W., Hu, S.M.: Nerf-sr: High quality neural radiance fields using supersampling. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 6445–6454 (2022) [2](#), [4](#), [10](#)
43. Wang, P., Zhao, L., Ma, R., Liu, P.: BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4170–4179 (June 2023) [2](#), [4](#)
44. Wang, P., Zhao, L., Ma, R., Liu, P.: Bad-nerf: Bundle adjusted deblur neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4170–4179 (2023) [10](#)
45. Wang, Y., Yu, J., Zhang, J.: Zero-shot image restoration using denoising diffusion null-space model. The Eleventh International Conference on Learning Representations (2023) [3](#), [4](#)
46. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) [8](#)
47. Yang, S., Wu, T., Shi, S., Lao, S., Gong, Y., Cao, M., Wang, J., Yang, Y.: Maniq: Multi-dimension attention network for no-reference image quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1191–1200 (2022) [9](#)
48. Zhang, K., Liang, J., Van Gool, L., Timofte, R.: Designing a practical degradation model for deep blind image super-resolution. In: IEEE International Conference on Computer Vision. pp. 4791–4800 (2021) [3](#), [4](#)

49. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) [7](#), [9](#)
50. Zhang, W., Zhai, G., Wei, Y., Yang, X., Ma, K.: Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 14071–14081 (2023) [9](#)
51. Zhang, W., Li, X., Chen, X., Qiao, Y., Wu, X.M., Dong, C.: Seal: A framework for systematic evaluation of real-world super-resolution. arXiv preprint arXiv:2309.03020 (2023) [4](#)
52. Zhou, K., Li, W., Wang, Y., Hu, T., Jiang, N., Han, X., Lu, J.: Netflix: High-quality neural view synthesis by learning a degradation-driven inter-viewpoint mixer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12363–12374 (2023) [2](#), [4](#), [10](#)
53. Zhou, Y., Li, Z., Guo, C.L., Bai, S., Cheng, M.M., Hou, Q.: Srformer: Permuted self-attention for single image super-resolution. arXiv preprint arXiv:2303.09735 (2023) [4](#), [13](#)

RaFE: Generative Radiance Fields Restoration

Supplementary Material

Zhongkai Wu¹, Ziyu Wan², Jing Zhang¹, Jing Liao², and Dong Xu³

¹ College of Software, Beihang University, China

² City University of Hong Kong, China

³ The University of Hong Kong, China

ZhongkaiWu@buaa.edu.cn

RaFE.github.io

1 Overview

In this supplementary material, additional training details and more experimental results are provided, including:

- Training & rendering details of the whole pipeline in Sec. 2.
- Training details of the pre-trained coarse NeRF in Sec. 3.
- More Discussion and experimental results in Sec. 4.
- Details of the calculation of diversity score in Sec. 5.
- More real-world results in Sec. 6.

2 Training & Rendering Details

The input images and corresponding viewpoints are randomly selected during the training process. We also randomly sample random code z from the normal distribution. And we use a discriminator learning rate of 0.002 and a generator learning rate of 0.0025. In the early stage of training, we blur the image to stabilize the training process and reduce the blur kernel size to zero gradually. Furthermore, we use a density regularization which minimizes the density differences between adjacent sampled points.

For the NeRF-Synthetic benchmark dataset (Blender), we sample 192 points for each ray, with 128 stratified sampling points and 48 importance sampling points. We assume that the blender object is in a $[-1.5, 1.5]^3$ cube and set the near and far plane of the ray to 2 and 6 respectively. For training, we restore 10K high-quality images randomly selected in 200 training views and we set the batch size to 32 and the minibatch std group size to 4 for optimizing the generator.

For the forward-facing datasets, we sample 192 points for each ray, with 128 stratified sampling points and 48 importance sampling points. We use normalized device coordinates (NDC) and set the near plane to 0 and the far plane to 1. The derivation of NDC can be found in NeRF [?]. The size of the dataset, the batch size, and the minibatch standard deviation group size are the same as those in the Blender dataset.

3 Details of Pre-trained Coarse NeRF

As mentioned in the main submission, we pre-train a coarse NeRF as the foundation of our residual fine NeRF. In this section, we describe our training details about coarse NeRF. We use NeRFStudio [?] as our codebase. We re-implement the hybrid explicit-implicit tri-plane representation within this framework and use degraded images for training supervision. We minimize the L2 loss, denoted as L_{rec} , between the rendered image and the degraded image. Inspired by K-planes [?], we incorporate additional losses: the TV loss (L_{tv}), which minimizes feature differences between adjacent coordinates on the tri-planes, and the distortion loss (L_{dis}), which pushes the density of a ray within a specific range. The total loss for optimizing the coarse NeRF is:

$$\mathcal{L}_{coarse} = \lambda_{rec}L_{rec} + \lambda_{tv}L_{tv} + \lambda_{dis}L_{dis}, \quad (1)$$

where λ_{rec} , λ_{tv} and λ_{dis} denote the trade-off parameters. Specifically, we use $\lambda_{rec} = 1$, $\lambda_{tv} = 0.01$ and $\lambda_{dis} = 0.001$ in our experiments.

4 More Discussions

Effects of residual coarse NeRF. In this ablation study, we show that the residual coarse NeRF could facilitate the model to be aware of the geometry and help better render highly detailed images, as shown in Fig. 1. We directly drop the coarse NeRF and train the generator from scratch with GAN loss and LPIPS loss, forcing the generator to learn the coarse structure and details jointly. We can observe that with the addition of coarse NeRF, RaFE could better model the structural information (like the Lego baseplate) and transparent material (like the drumhead).

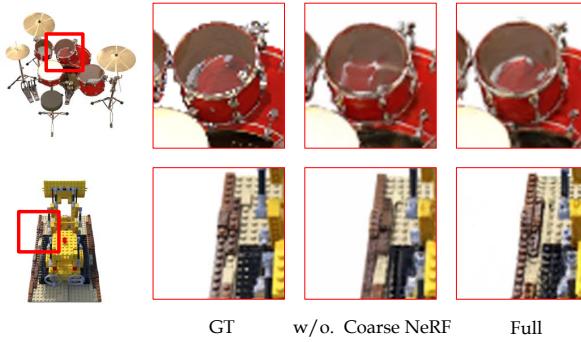


Fig. 1: Ablation study of residual coarse NeRF. Our full pipeline renders the images with better quality, demonstrating the great effectiveness of the residual coarse NeRF.

Effects of view direction. In this ablation study, we examine the effects of view direction conditioning. We drop the second MLP \mathcal{M}_{color} in the decoder and

let the RGB value directly be decoded by the first MLP jointly with density. As shown in Fig. 2, objects with non-Lambertian surfaces exhibit significant light reflections. In contrast, without view direction information, the object in the figure appears to have no reflections.

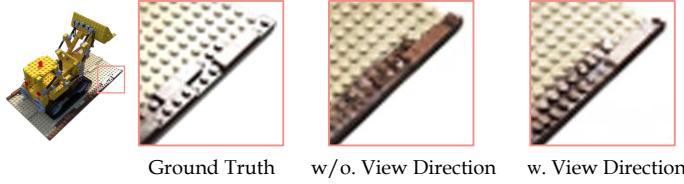


Fig. 2: Ablation of view direction. Reflections can be observed when using view direction conditioning.

Effects of patch sampling strategy. In this experiment, we examine the effects of our patch-sampling strategy by training the model on the room scene. To highlight the significance of the Beta sampling strategy, we consider the case where the value of δ_x, δ_y are sampled from a uniform distribution: $\delta_x, \delta_y \sim \mathcal{U}(0, 1)$. This approach essentially obtains image patches through uniform cropping, which offers a baseline for comparison. The results are shown in Fig. 3. As can be seen, the model trained with uniform sampling suffers from unstable optimization and severe artifacts, which effectively demonstrates the efficacy of beta sampling in maintaining the rendering quality.



Fig. 3: Ablation of patch sampling strategy. The training process becomes unstable without patch sampling strategy and causes severe artifacts

Performance on NeRF-like degradation. In this experiment, we additionally examine our method’s performance on NeRF-like degradation [?] caused by the reconstruction process of NeRF. We first use high-quality images to train a NeRF. Although trained sufficiently, artifacts can still be presented in the images rendered by NeRF models, as discussed in [?]. Then we collect the rendered images as a low-quality set and use our restoration method to recover an artifact-free NeRF. As can be seen in Fig. 4, Although not specifically tailored

for the NeRF-like degradation, our method still demonstrates clearly improved performance.



Fig. 4: Performance on NeRF-like degradation. Although not specifically tailored for the NeRF-like degradation, our method still demonstrates satisfactory performance.

5 Details of the diversity score

In this section, we describe the details of the diversity score used to evaluate the diversity of the generated image set. In particular, we follow the computational methods described in [?]. We first calculate the minimal LPIPS score for each image with other images in the image set. Then we average the per-image scores to get the overall score for the whole image set. The higher diversity score means the greater diversity of the image set. Here we provide pseudo-code in Algorithm 1 to explain our computational method better.

Algorithm 1: Calculation process of the diversity score

Input: Image set I
Output: diversity score s

```

 $s_{sum} \leftarrow 0$ 
foreach  $i_s$  in  $I$  do
     $s_{min} \leftarrow \inf$ 
    foreach  $i_d$  in  $I$  do
        if  $i_s \neq i_d$  then
             $s_{lpipl} \leftarrow LPIPS(i_s, i_d)$ 
             $s_{min} \leftarrow \min(s_{min}, s_{lpipl})$ 
        end
    end
     $s_{sum} \leftarrow s_{sum} + s_{min}$ 
end
 $s \leftarrow s_{sum}/|A|$ 
return  $s$ 

```

6 More results of real-world scenario

In this section, we provide more experiments on real-world scenarios to show the capability of our method in addressing multiple types of degradation in the real world in Fig. 5.

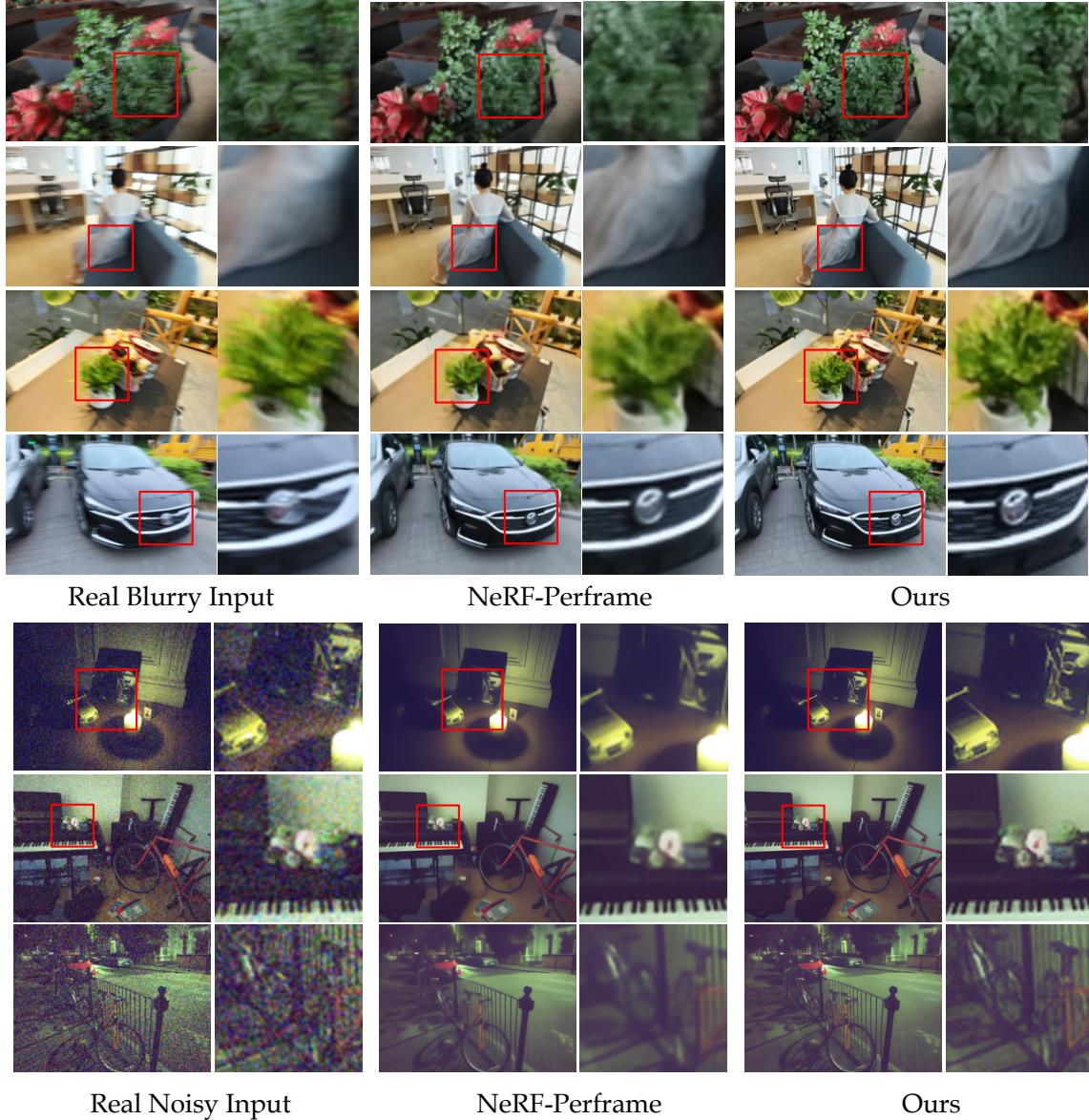


Fig. 5: More qualitative results on different real-world 3D restoration tasks.