GenN2N: Generative NeRF2NeRF Translation

Xiangyue Liu[1] Han Xue[2] Kunming Luo[1] Ping Tan[1†] Li Yi[2,3,4†]

[1] Hong Kong University of Science and Technology [2] Tsinghua University
[3] Shanghai Artificial Intelligence Laboratory [4] Shanghai Qi Zhi Institute

Figure 1. We introduce GenN2N, a unified framework for NeRF-to-NeRF translation, enabling a range of 3D NeRF editing tasks, including text-driven editing, colorization, super-resolution, inpainting, etc. We show at least two rendering views of edited NeRF scenes at inference time. Given a 3D NeRF scene, GenN2N can produce high-quality editing results with suitable multi-view consistency.

Abstract

We present GenN2N, a unified NeRF-to-NeRF translation framework for various NeRF translation tasks such as text-driven NeRF editing, colorization, super-resolution, inpainting, etc. Unlike previous methods designed for individual translation tasks with task-specific schemes, GenN2N achieves all these NeRF editing tasks by employing a plug-and-play image-to-image translator to perform editing in the 2D domain and lifting 2D edits into the 3D NeRF space. Since the 3D consistency of 2D edits may not be assured, we propose to model the distribution of the underlying 3D edits through a generative model that can cover all possible edited NeRFs. To model the distribution of 3D edited NeRFs from 2D edited images, we carefully design a VAE-GAN that encodes images while decoding NeRFs. The latent space is trained to align with a Gaussian distribution and the NeRFs are supervised through an adversarial loss on its renderings. To ensure the latent code does not depend on 2D viewpoints but truly reflects the 3D edits, we also regularize the latent code through a contrastive learning scheme. Extensive experiments on various editing tasks show GenN2N, as a universal framework, performs as well or better than task-specific specialists while possessing flexible generative power. More results on our project page: https://xiangyueliu.github.io/GenN2N/.

1. Introduction

Over the past few years, Neural radiance fields (NeRFs) [23] have brought a promising paradigm in the realm of 3D reconstruction, 3D generation, and novel view synthesis due to their unparalleled compactness,

†Corresponding authors.

high quality, and versatility. Extensive research efforts have been devoted to creating NeRF scenes from 2D images [4, 20, 22, 36, 38] or just text [13, 26] input. However, once the NeRF scenes have been created, these methods often lack further control over the generated geometry and appearance. NeRF editing has therefore become a notable research focus recently.

Existing NeRF editing schemes are usually task-specific. For example, researchers have developed NeRF-SR [34], NeRF-In [19], PaletteNeRF [16] for NeRF super-resolution, inpainting, and color-editing respectively. These designs require a significant amount of domain knowledge for each specific task. On the other hand, in the field of 2D image editing, a growing trend is to develop universal image-to-image translation methods to support versatile image editing [25, 29, 43]. By leveraging foundational 2D generative models, e.g., stable diffusion [28], these methods achieve impressive editing results without task-specific customization or tuning. We then ask the question: can we conduct universal NeRF editing leveraging foundational 2D generative models as well?

The first challenge is the representation gap between NeRFs and 2D images. It is not intuitive how to leverage image editing tools to edit NeRFs. A recent text-driven NeRF editing method [9] has shed some light on this. The method adopts a "render-edit-aggregate" pipeline. Specifically, it gradually updates a NeRF scene by iteratively rendering multi-view images, conducting text-driven visual editing on these images, and finally aggregating the edits in the NeRF scene. It seems that replacing the image editing tool with a universal image-to-image translator could lead to a universal NeRF editing method. However, the second challenge would then come. Image-to-image translators usually generate diverse and inconsistent edits for different views, e.g. turning a man into an elf might or might not put a hat on his head, making edits aggregation intricate. Regarding this challenge, Instruct-NeRF2NeRF [9] presents a complex optimization technique to pursue unblurred NeRF with inconsistent multi-view edits. Due to its complexity, the optimization cannot ensure the robustness of the outcomes. Additionally, the unique optimization outcome fails to reflect the stochastic nature of NeRF editing. Users typically anticipate a variety of edited NeRFs just like the diverse edited images.

To tackle the challenges above, we propose GenN2N, a unified NeRF-to-NeRF translation framework for various NeRF editing tasks such as text-driven editing, colorization, super-resolution, inpainting (see Fig. 1). In contrast to Instruct-NeRF2NeRF which adopts a "render-edit-aggregate" pipeline, we first render a NeRF scene into multi-view images, then exploit an image-to-image translator to edit different views, and finally learn a generative model to depict the distribution of NeRF edits. Instead of aggregating all the image edits to form a single NeRF edit, our key idea is to embrace the stochastic nature of content editing by modeling the distribution of the edits in the 3D NeRF space.

Specifically given a NeRF model or its multi-view images, along with the editing goal, we first generate edited multi-view images using a plug-and-play image-to-image translator. Each view corresponds to a unique 3D edit with some geometry or appearance variations. Conditioned on the input NeRF, GenN2N trains a conditional 3D generative model to reflect such content variations. At the core of GenN2N, we design a 3D VAE-GAN that incorporates a differentiable volume renderer to connect 3D content creation with 2D GAN losses, ensuring that the inconsistent multi-view renderings can still help each other regarding 3D generation. Moreover, we introduce a contrastive learning loss to ensure that the 3D content variation can be successfully understood just from edited 2D images without being influenced by the camera viewpoints. During inference, users can simply sample from the conditional generative model to obtain various 3D editing results aligned with the editing goal. We have conducted experiments on human, items, indoor and outdoor scenes for various editing tasks such as text-driven editing, colorization, super-resolution and inpainting, demonstrating the effectiveness of GenN2N in supporting diverse NeRF editing tasks while keeping the multi-view consistency of the edited NeRF.

We summarize the contribution of this paper as follows,

- A generative NeRF-to-NeRF translation formulation for the universal NeRF editing task together with a generic solution;
- a 3D VAE-GAN framework that can learn the distribution of all possible 3D NeRF edits corresponding to the a set of input edited 2D images;
- a contrastive learning framework that can disentangle the 3D edits and 2D camera views from edited images;
- extensive experiments demonstrating the superior efficiency, quality, and diversity of the NeRF-to-NeRF translation results.

2. Related Work

NeRF Editing. Previous works such as EditNeRF [21] propose a conditional neural field that enables shape and appearance editing in the latent space. PaletteNeRF [16, 37] focuses on controlling color palette weights to manipulate appearance. Other approaches utilize bounding boxes [41], meshes [39], point clouds [6], key points [45], or feature volumes [18] to directly manipulate the spatial representation of NeRF. However, these methods either heavily rely on user interactions or have limitations in terms of spatial deformation and color transfer capabilities.

NeRF Stylization. Images-referenced stylization [7, 12, 42] often prioritize capturing texture style rather than de-

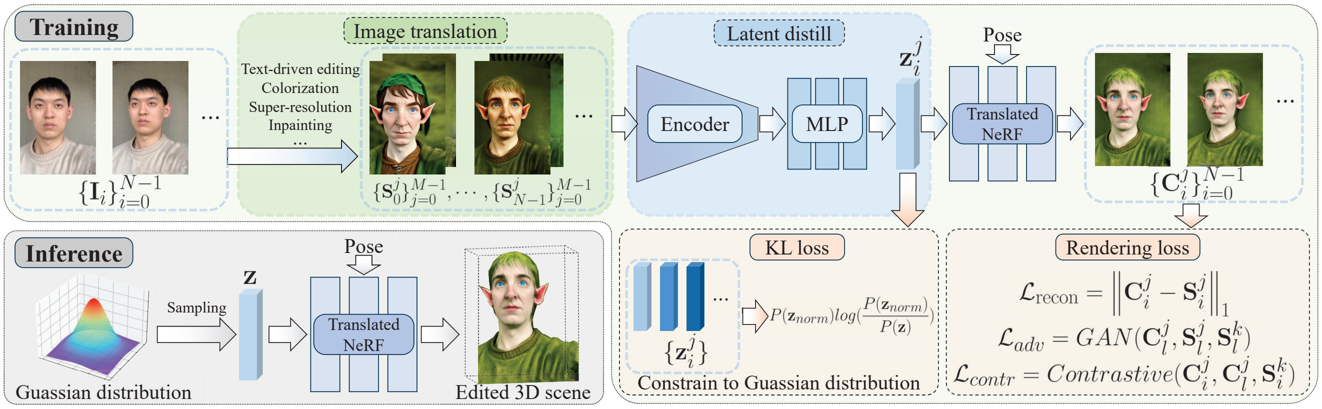

Figure 2. **Overview of GenN2N.** We first edit the source image set $\{\mathbf{I}_i\}_{i=0}^{N-1}$ using 2D image-to-image translation methods, e.g., text-driven editing, colorization, zoom out, etc. For each view $i \in [0, N-1]$, we generate M edited images, resulting in a group of translated image set $\{\{\mathbf{S}_i^j\}_{j=0}^{M-1}\}_{i=0}^{N-1}$. Then we use the Latent Distill Module to learn $M \times N$ edit code vectors from the translated image set, which serve as the input of the translated NeRF. To optimize our GenN2N, we design four loss functions: a KL loss to constrain the latent vectors to a Gaussian distribution; and $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{contr}}$ to optimize the appearance and geometry of the translated NeRF. At inference, we can sample a latent vector $\mathbf{z}$ from Gaussian distribution and render a corresponding multi-view consistent 3D scene with high quality.

tailed content, resulting in imprecise editing appearance of NeRF only. Text-guided works [33, 35], on the other hand, apply contrastive losses based on CLIP [27] to achieve the desired edits. While text references usually describe the global characteristics of the edited results, instructions offer a more convenient and precise expression.

Instruct-driven NeRF editing. Among numerous image-to-image translation works, InstructPix2Pix [2] stands out by efficiently editing images following instructions. It leverages large pre-trained models in the language and image domains [3, 28] to generate paired data (before and after editing) for training. While editing NeRF solely based on edited images is problematic due to multi-view inconsistency. To address this, an intuitive yet heavy approach [9] is to iteratively edit the image and optimize NeRF. In addition, NeRF-Art [35] and DreamEditor [46] adopt a CLIP-based contrastive loss [27] and score distillation sampling [26] separately to supervise the optimization of editing NeRF. Inspired by Generative Radiance Fields [5, 30], We capture various possible NeRF editing in the generative space to solve it.

3. Method

Given a NeRF scene, we present a unified framework GenN2N to achieve various editing on the 3D scene leveraging geometry and texture priors from 2D image editing methods, such as text-driven editing, colorization, super-resolution, inpainting, etc. While a universal image-to-image translator can theoretically accomplish these 2D editing tasks, we actually use a state-of-the-art translator for each task. Therefore, we formulate each 2D image editing method as a plug-and-play image-to-image translator and all NeRF editing tasks as our universal NeRF-to-NeRF translation, in which the given NeRF is translated into NeRF scenes with high rendering quality and 3D geometry consistency according to the user-selected editing target. The overview of GenN2N is illustrated in Fig. 2, we first perform image-to-image translation in the 2D domain and then lift 2D edits to 3D and achieve NeRF-to-NeRF translation.

Given N multi-view images $\{\mathbf{I}_i\}_{i=0}^{N-1}$ of a scene, we first use Nerfstudio [32] to train the original NeRF. Then we use a plug-and-play image-to-image translator to edit these source images. However, the content generated by the 2D translator may be inconsistent among multi-view images. For example, using different initial noise, the 2D translator [1] may generate different content for image editing, which makes it difficult to ensure the 3D consistency between different view directions in the 3D scene.

To ensure the 3D consistency and rendering quality, we propose to model the distribution of the underlying 3D edits through a generative model that can cover all possible edited NeRFs, by learning an edit code for each edited image so that the generated content can be controlled by this edit code during the NeRF-to-NeRF translation process.

For each view $\in [0, N-1]$, we generate M edited images, resulting in a group of the translated image set $\{\{\mathbf{S}_i^j\}_{j=0}^{M-1}\}_{i=0}^{N-1}$. Then we design a Latent Distill Module described in Sec. 3.1 to map each translated image $\mathbf{S}_i^j$ into an edit code vector $\mathbf{z}_i^j$ and design a KL loss $\mathcal{L}_{\text{KL}}$ to constrain those edit code vectors to a Gaussian distribution. Conditioned on the edit code $\mathbf{z}_i^j$, we perform NeRF-to-NeRF translation in Sec. 3.2 by rendering multi-view images $\{\mathbf{C}_i\}_{i=0}^{N-1}$ and optimize the translated NeRF by three loss functions: the reconstruction loss $\mathcal{L}_{\text{recon}}$, the adversarial

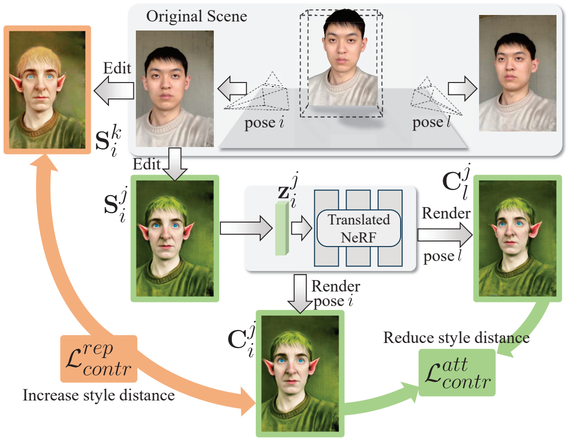

Figure 3. **Illustration of our proposed contrastive loss functions.** Regarding the multi-view rendered images $\mathbf{C}_i^j$ and $\mathbf{C}_l^j$ sharing the same edit code, we resend them to our Latent Distill Module to extract $\mathbf{z}_i^j$ and $\mathbf{z}_l^j$, and aggregate them via $\mathcal{L}_{\text{contr}}^{\text{att}}$. In addition, for $\mathbf{S}_i^k$ whose editing style vary from $\mathbf{S}_i^j$, $\mathcal{L}_{\text{contr}}^{\text{rep}}$ increase the distance between edit codes of them.

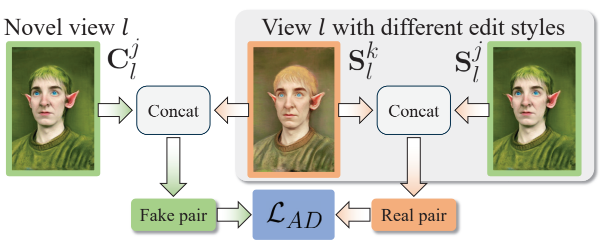

Figure 4. **Illustration of our proposed conditional adversarial loss functions.** Our conditional discriminator distinguishes artifacts such as blur and distortion in novel-view rendered image $\mathbf{C}_l^j$ compared with target image $\mathbf{S}_l^j$. $\mathbf{S}_l^j$ and $\mathbf{S}_l^k$ are edited with same view but various styles, the latter serves as the condition to concatenate with $\mathbf{C}_l^j$ and $\mathbf{S}_l^j$ and manufacture fake and real pairs.

loss $\mathcal{L}_{\text{AD}}$, and the contrastive loss $\mathcal{L}_{\text{contr}}$. After the optimization of the translated NeRF, as described in Sec. 3.3, we can sample an edit code $\mathbf{z}$ from Gaussian distribution and render the corresponding edited 3D scene with high quality and multi-view consistency in the inference stage.

3.1. Latent Distill Module

Image Translation. As illustrated in Fig. 2, GenN2N is a unified framework for NeRF-to-NeRF translation, in which the core is to perform a 2D image-to-image translation and lift 2D edits into universal 3D NeRF-to-NeRF translation. Given the source multi-view image set $\{\mathbf{I}_i\}_{i=0}^{N-1}$ of a NeRF scene, we first perform image editing M times for each view using a plug-and-play 2D image-to-image translator, producing a group of translated image set $\{\{\mathbf{S}_i^j\}_{j=0}^{M-1}\}_{i=0}^{N-1}$. In this paper, we use several 2D translation tasks to show the adaptability of our GenN2N: text-driven editing, super-resolution, colorization and inpainting. For more details about those 2D image editing methods, please refer to the supplementary materials.

Edit Code. Since 2D image-to-image translation may generate different content even with the same editing target, causing the inconsistency problem in the 3D scene. We propose to map each edited image $\mathbf{S}_i^j$ into a latent feature vector named edit code to characterize these diverse editings. We employ the off-the-shelf VAE encoder from stable diffusion [28] to extract the feature from $\mathbf{S}_i^j$ and then apply a tiny MLP network to produce this edit code $\mathbf{z}_i^j \in \mathbb{R}^{64}$. During the training process, we keep the pre-trained encoder fixed and only optimize the parameters of the tiny MLP network. This mapping process can be formulated as follows:

$$\mathbf{z}_i^j = \mathcal{D}(\mathbf{S}_i^j) = \mathcal{M}(\mathcal{E}(\mathbf{S}_i^j)), \tag{1}$$

where $\mathcal{D}$ represent this mapping process, $\mathcal{E}$ is the fixed encoder, and $\mathcal{M}$ is the learnable tiny MLP.

KL loss. In order to facilitate effective sampling of the edit code so as to control the editing diversity of our NeRF-to-NeRF translation, we need to constrain the edit code to a well-defined distribution. Thus we design a KL loss to encourage $\mathbf{z}_i^j$ to approximate a Gaussian distribution:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{\mathbf{S}\in\{\{\mathbf{S}_i^j\}_{j=0}^{M-1}\}_{i=0}^{N-1}}[P(\mathbf{z}_{normal})log(\frac{P(\mathbf{z}_{normal})}{P(\mathcal{D}(\mathbf{S}))})], \tag{2}$$

where $P(\mathbf{z}_{normal})$ denotes probability distribution of the standard Gaussian distribution in $\mathbb{R}^{64}$ and $P(\mathcal{D}(\mathbf{S}))$ means probability distribution of the extracted edit codes.

Contrastive loss. It is not assured that edit codes $\mathbf{z}$ obtained from the Latent Distill Module contain only the editing information while excluding viewpoint-related effects. However, since the translated NeRF utilizes $\mathbf{z}$ to edit scenes, it yields instability if $\mathbf{z}$ violently changes given images that are similar in appearance but different in viewpoints. To ensure the latent code does not depend on 2D viewpoints but truly reflects the 3D edits, we regularize the latent code through a contrastive learning scheme. Specifically, we reduce the distance between edit codes of different-view rendered images from a translated NeRF that share the same edit code, while increasing the distance between same-view images that are multi-time edited by the 2D image-to-image translator. As illustrated in Fig. 3, given an edit code $\mathbf{z}_i^j$ extracted from the i-th input view at the j-th edited image $\mathbf{S}_i^j$, we render multi-view images $\{\mathbf{C}_l^j\}_{l=0}^{N-1}$ using the translated NeRF conditioned on $\mathbf{z}_i^j$. Then we employ contrastive learning to encourage the edit code $\mathbf{z}_i^j$ to be close to $\{\mathbf{z}_l^j\}_{l=0}^{N-1}$ extracted from $\{\mathbf{C}_l^j\}_{l=0}^{N-1}$, while being distinct from the edit codes $\{\mathbf{z}_i^k\}_{k=0}^{M-1}$ extracted from $\{\mathbf{S}_i^k\}_{k=0}^{M-1}$, where $k \neq j$.

Figure 5. **Text-Driven Editing.** We sample 4 inference results for both text-driven editing tasks. The diversity of geometry and appearance showcases awesome generative ability of GenN2N, on the premise of maintaining the 3D consistency between different viewpoints.

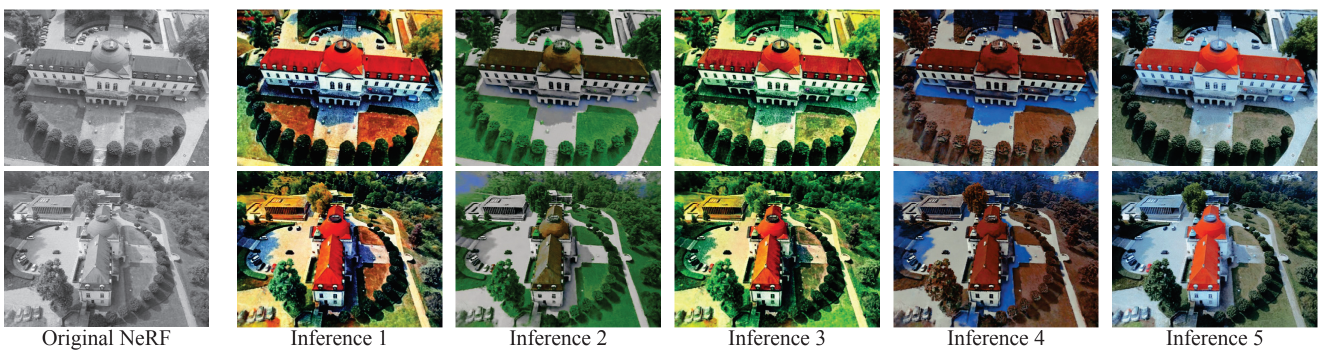

Figure 6. **Colorization.** Our method colorizes the gray-scale 3D scene consistently across views. By changing the edit code during inference, diverse colorized scenes can be rendered with satisfying photorealism and reasonably rich colors.

Specifically, our contrastive loss is designed as follows:

$$\begin{aligned}\mathcal{L}_{\text{contr}} &= \mathcal{L}_{\text{contr}}^{\text{att}} + \mathcal{L}_{\text{contr}}^{\text{rep}} \\ &= \sum_{l=0}^{N-1} ||\mathbf{z}_i^j - \mathbf{z}_l^j||_2^2 + \sum_{k=0}^{M-1} max(0, \alpha - ||\mathbf{z}_i^j - \mathbf{z}_i^k||_2^2),\end{aligned} \tag{3}$$

where α represents the margin that encourages the difference in features, and $k \neq j$.

3.2. NeRF-to-NeRF translation

Translated NeRF. After 2D image-to-image translation, we need to lift these 2D edits to the 3D NeRF. For this purpose, we propose to modify the original NeRF as a translated NeRF that takes the edit code $\mathbf{z}$ as input and generates the translated 3D scene according to the edit code. We refer readers to the supplementary for more details about the network architecture.

Reconstruction loss. Given an edit code $\mathbf{z}_i^j$ extracted from the edited image $\mathbf{S}_i^j$, we can generate a translated NeRF to render $\mathbf{C}_i^j$ from the same viewpoint. Then we define the reconstruction loss as the L1 normalization and Learned Perceptual Image Patch Similarity (LPIPS) [44] between the rendered image $\mathbf{C}_i^j$ and the edited image $\mathbf{S}_i^j$ as follows:

$$\begin{aligned}\mathcal{L}_{\text{recon}} &= \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{LPIPS}} \\ &= \left\|\mathbf{C}_i^j - \mathbf{S}_i^j\right\|_1 + LPIPS[\mathcal{P}(\mathbf{C}_i^j) - \mathcal{P}(\mathbf{S}_i^j)],\end{aligned} \tag{4}$$

where $\mathcal{P}$ means a patch sampled from the image. Note that due to the lack of 3D consistency of the edited multi-view image, the supervision of the edited image from other viewpoints $\{\mathbf{S}_l^j\}_{l \neq i}$ will lead to conflicts in pixel-space optimization. Therefore, we only employ reconstruction loss on the same view image $\mathbf{S}_i^j$ to optimize the translated NeRF.

Adversarial loss. Since the 3D consistency of edited multi-view images is not assured, relying solely on the reconstruction loss on the same view often leads to blurry or distorted

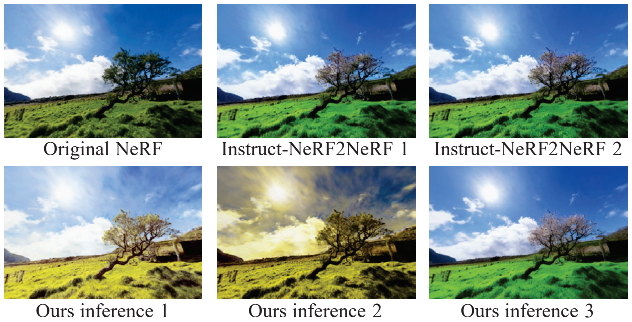

Figure 7. **Comparisons with baselines of text-driven NeRF editing.** We compare our method with Instruct-NeRF2NeRF [9] in the editing by using the text prompt *"Make it Spring"*.

Method	CLIP Text-Image Direction Similarity↑	CLIP Direction Consistency ↑	FID ↓
InstructPix2Pix [2]+NeRF	0.1669	0.8475	270.542
Instruct-NeRF2NeRF	0.2021	0.9828	148.021
Ours w/o $\mathcal{L}_{\text{adv}}$	0.1920	0.9657	162.275
Ours w/o $\mathcal{L}_{\text{contr}}$	0.2007	0.9749	156.524
Ours	**0.2089**	**0.9864**	**137.740**

Table 1. **Quantitative results on text-driven editing.** We compare our method with the naive method of directly combining InstractPix2Pix [2] with NeRF and the state-of-the-art method Instruct-NeRF2NeRF [9].

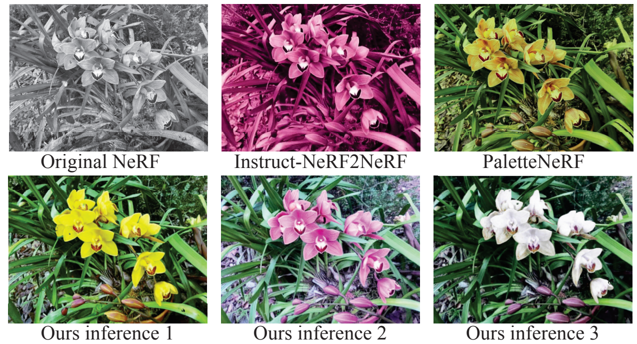

Figure 8. **Comparisons with baselines of NeRF colorization.** We compare with PaletteNeRF[16] in colorization.

Method	CF ↑	FID ↓
DDColor [14]+NeRF	40.435	148.957
Instruct-NeRF2NeRF	45.599	201.456
PaletteNeRF [16]	39.654	–
Ours w/o $\mathcal{L}_{\text{adv}}$	35.031	137.740
Ours w/o $\mathcal{L}_{\text{contr}}$	34.829	105.750
Ours	**65.099**	**35.041**

Table 2. **Quantitative results on colorization.** We colorize images with the translator, DDcolor [14], and edit NeRF by directly optimizing NeRF, Instruct-NeRF2NeRF [9] and our NeRF translation method. The quantitative comparison is conducted between these methods as well as PaletteNeRF [16].

artifacts on novel views. Previous research demonstrates the effectiveness of conditional adversarial training in preventing the production of blurry rendered images resulting from conflicts that arise from noise in the camera extrinsic when performing image supervision from different viewpoints [11]. The function of the condition is to guide discriminator with fine-grained information from the same viewpoint, thus preventing GAN mode collapse.

It inspires us to incorporate conditional adversarial loss on rendered images from the translated NeRF, which is conducive to distinguish artifacts in rendered images. As illustrated in Fig.4, the discriminator $\mathbf{D}$ takes into real pairs and fake pairs. Each real pair $\mathbf{R}$ consists of $\mathbf{S}^j$ and $\mathbf{S}^j - \mathbf{S}^k$ where $\mathbf{S}^j \in \{\mathbf{S}_i^j\}_{i=0}^{N-1}$ and $\mathbf{S}^k \in \{\mathbf{S}_i^k\}_{i=0}^{N-1}$ are from two sets of edited images from the image translation. Similarly, each fake pair $\mathbf{F}$ consists of $\mathbf{C}^j$ and $\mathbf{C}^j - \mathbf{S}^k$ in which $\mathbf{C}^j \in \{\mathbf{C}_i^j\}_{i=0}^{N-1}$ is generated by translated NeRF. Note that the images in the same pair come from the same viewpoint. The pairs are concatenated in RGB channels and fed into the discriminator. We optimize the discriminator $\mathbf{D}$ and translated NeRF with the objective functions below:

$$\begin{aligned} \mathcal{L}_{\text{AD-D}} &= \mathbb{E}_{\mathbf{R}}[-log(\mathbf{D}(\mathbf{R}))] + \mathbb{E}_{\mathbf{F}}[-log(1-\mathbf{D}(\mathbf{F}))], \\ \mathcal{L}_{\text{AD-G}} &= \mathbb{E}_{\mathbf{F}}[-log(\mathbf{D}(\mathbf{F}))]. \end{aligned} \tag{5}$$

Optimization. During the training process, we jointly optimize the loss functions mentioned above: $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{contr}}$ for the edit code, $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{AD-G}}$ for the translated NeRF, and $\mathcal{L}_{\text{AD-D}}$ for the discriminator. The total loss formula is expressed as follows:

$$\mathcal{L} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{AD-G}} + \mathcal{L}_{\text{AD-D}} + \mathcal{L}_{\text{contr}}. \tag{6}$$

where we assign each regularization term the weight of 1.0, 1.0, 0.1, 0.1, 0.1 in all of our experiments. The weights can be adjusted to prioritize different aspects of the training objective, such as reconstruction accuracy, adversarial training, and perceptual quality.

3.3. Inference

After the optimization of our GenN2N, the translated NeRF is optimized to be able to render the target scene conditioned on the edit code. As shown in Fig. 1, users can simply sample an edit code from the Gaussian distribution and use the translated NeRF to render the 3D scene with high-quality and multi-view 3D consistency.

4. Experiments

Our proposed GenN2N is a unified NeRF-to-NeRF translation framework which can support various NeRF editing tasks. In this paper, we demonstrate the effectiveness of GenN2N by a suite of challenging tasks:

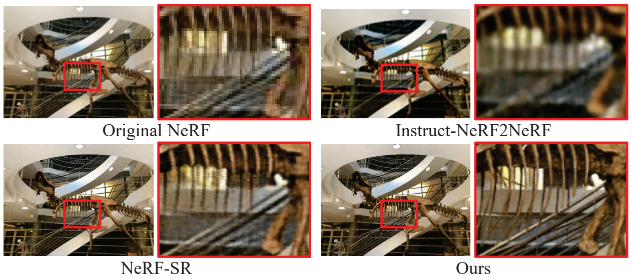

Figure 9. **Comparisons with baselines of NeRF super-resolution.** We compare with NeRF-SR [34] in super-resolution.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
ResShift [40]+NeRF	19.978	0.535	0.1156
Instruct-NeRF2NeRF	20.299	0.642	0.2732
NeRF-SR [34]	27.957	0.897	0.0937
Ours w/o $\mathcal{L}_{\text{adv}}$	12.555	0.663	0.2001
Ours w/o $\mathcal{L}_{\text{contr}}$	15.372	0.662	0.1834
Ours	**28.501**	**0.913**	**0.0748**

Table 3. **Quantitative results on super-resolution.** We improve image resolution with ResShift [40] and edit NeRF by directly optimizing NeRF, Instruct-NeRF2NeRF [9] and our NeRF translation method. The quantitative comparison is conducted between these methods as well as NeRF-SR [34].

(1) **Text-driven Editing** edits the given NeRF scene to a set of NeRF scenes according to the text instruction.
(2) **Colorization** transforms a gray-scale NeRF scene to a set of plausible color NeRF scenes.
(3) **Super-resolution** enhances the resolution of NeRF and enables multiple plausible outcomes.
(4) **Inpainting** fills in user-specified masked regions in the NeRF scene with realistic content.

We achieve those tasks by simply changing the plug-and-play 2D image translator in our framework, without any additional task-specific design. Previous studies have extensively explored some of these issues like text-driven editing, colorization, super-resolution, and inpainting. However, there is rarely a unified framework that can achieve all these tasks with strong performance, high quality, and plausible multi-view consistent 3D structure. Furthermore, GenN2N can also perform zooming out and text-driven inpainting in NeRF-to-NeRF translation, which were not explored in prior research. We refer readers to the supplementary materials for detailed experiment settings, dataset settings and implementation details.

4.1. Comparisons

Text-driven Editing. We achieve text-driven editing of the given NeRF by using InstructPix2Pix [2] as the 2D image-to-image translator in our framework. We compared our approach to a naive solution, which involves optimizing a NeRF with edited images via InstructPix2Pix. However, this naive approach leads to a 3D inconsistency problem among different edits. While Instruct-NeRF2NeRF [9] proposed an iterative updating mechanism to address this issue, it falls short in capturing the diversity of different edits, making it challenging to ensure the quality of the outcomes. To evaluate our method, we conducted experiments on the *Face* [9] and *Fangzhou* [35] self-portrait datasets, and *Farm* [9] dataset, comparing GenN2N with the state-of-the-art NeRF editing method Instruct-NeRF2NeRF [9].

Quantitative results are presented in Table 1, where we employed CLIP Text-Image Direction Similarity [9], CLIP Direction Consistency [9], and Fréchet Inception Distance (FID) [10] as evaluation metrics. The results highlight the superior performance of GenN2N over other methods, demonstrating its effectiveness in producing high-quality 3D text-driven editing results. Furthermore, we provide a qualitative comparison between GenN2N and Instruct-NeRF2NeRF [9] in Figure 7. Notably, Instruct-NeRF2NeRF was trained twice, but the results are nearly identical. In contrast, GenN2N can render edited scenes with various effects, consistent with the input text instruction, by inferring with different edit codes sampled from a Gaussian distribution. To further illustrate the generative capability of GenN2N, we showcase additional results of our inference renderings in Figure 5, demonstrating its diverse generative ability in terms of appearance and geometry.

Colorization. For NeRF colorization, GenN2N uses DDColor [14] as the 2D image-to-image translator. CoRF [8] and Palette-NeRF [16] do a similar task and we compare with Palette-NeRF in Table 2. The Fréchet Inception Distance (FID) [10] and colorfulness score (CF) [17] are used to measure the distribution similarity and vividness of generated images. We show visual comparison results in Fig. 6 and Fig. 8. We can find that with different edit codes, the scene can be rendered in different color styles. It is noticeable that with the same edit code, the color rendered from different views is consistent. This strongly demonstrates the effectiveness of our method in translating NeRF while keeping the 3D consistency of the scene.

Super-resolution. When only low-resolution images are available, our methods can boost NeRF in reconstructing scenes at higher resolution, while keeping view consistency and avoiding blurry outputs. We achieve this by employing ResShift [40] as the 2D image-to-image translator in GenN2N. Following state-of-the-art method NeRF-SR [34], we conduct experiments on *LLFF* dataset [23], using PSNR, SSIM, and LPIPS as evaluation metrics. As shown in Table 3, GenN2N obtains NeRF-to-NeRF translation with higher performance than NeRF-SR [34]. Moreover, we also provide qualitative comparison results in Fig. 9, where GenN2N produces clearer and more realistic rendering results than previous methods.

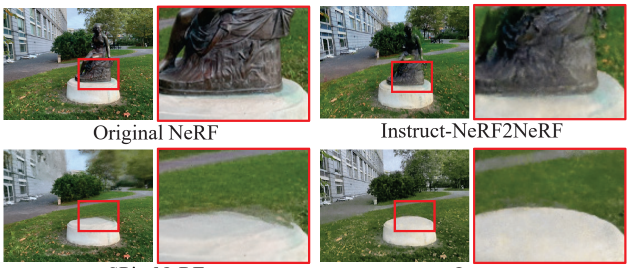

Figure 10. **Comparisons with baselines of NeRF inpainting.** We compare with SPIn-NeRF [24] in inpainting.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
LaMa [31]+NeRF	18.983	0.3706	0.1730
Instruct-NeRF2NeRF	16.734	0.3088	0.2750
SPin-NeRF [24]	24.369	0.7217	0.1754
Ours	**26.868**	**0.8137**	**0.1284**

Table 4. **Quantitative results on NeRF inpainting.**

Inpainting. The goal of NeRF Inpainting is to fill the 3D content of regions specified by users. SPIn-NeRF [24] achieves this through a multi-step process: it employs SAM [15] for object segmentation, utilizes LaMa [31] to paint the background content in multi-view images, and subsequently trains the NeRF model with color, depth, and perceptual cues. In our experiments, we use SAM and LaMa as the 2D image-to-image translator in our GenN2N, which is the same setting as SPin-NeRF [24]. Quantitative comparisons on *statue* dataset [24] are shown in Table 4, where GenN2N achieves superior PSNR and SSIM scores than SPin-NeRF [24], highlighting the effectiveness of our GenN2N framework. In addition, qualitative results are showcased in Fig. 10 revealing that while SPin-NeRF [24] fails to generate reasonable content behind the masked object, our GenN2N produces realistic content in the same region with fine multi-view consistency.

4.2. Ablation Studies

We conduct comprehensive ablation experiments to validate the designs of each component in GenN2N. Due to space limitations, we only highlight the essential aspects below. Please refer to supplementary for more details.

The Contrastive Loss. We demonstrate the advantages of incorporating our proposed contrastive loss in Table 1, 2, 3. The motivation is to disentangle the camera view and edit information present in the latent space. We achieve this by reducing the distance between edit codes of different-view rendered images from a translated NeRF that shares the same edit code, while increasing the distance between same-view images that are edited by the 2D translator with diverse edit styles. As demonstrated in Tables, the absence of contrastive loss leads to the generation of blurry areas in the rendered images, resulting in a decrease in the metric scores. This blurriness can be attributed to the inclusion of pose information within the edit code $\mathbf{z}$. By incorporating the contrastive loss, our method successfully achieves a uniform appearance with different observing views under the same style latent $\mathbf{z}$.

Discriminator for Novel Views. We demonstrate the effectiveness of employing a conditional discriminator to address artifacts caused by inconsistent cross-view edited images and to enhance the quality of novel view rendering images, as depicted in Table 1, 2, 3. The removal of this conditional discriminator results in blurry novel view images with artifacts in the background region. We attribute these undesirable effects to the inability of current image-to-image translation methods, such as InstructPix2Pix [2], to produce image editing consistently across multi-view images. To mitigate these issues, we introduce a conditional discriminator between rendered images from the translated NeRF and edited images from the 2D image-to-image translator. This inclusion successfully eliminates artifacts and enhances the image quality of rendered images from the translated NeRF.

4.3. Applications

We demonstrate the versatility and robustness of GenN2N by exploring two translation applications: Zoom Out and Text-Driven Inpainting. While existing 2D translators [1, 29] can complete these tasks, 3D editing has not been explored. We achieve these tasks by incorporating Blended Latent Diffusion [1] as the 2D image-to-image translator, enabling us to generate diverse and high-quality content with multi-view consistency. Please refer to our supplementary materials for the results of these applications due to space constraints.

5. Conclusions

We introduce GenN2N, a unified NeRF-to-NeRF translation framework that can handle various NeRF editing tasks. Unlike previous task-specific approaches, our framework uses an image-to-image translator for 2D editing and integrates the results into 3D NeRF space. To address the challenge of ensuring 3D consistency, we propose modeling the distribution of 3D edited NeRFs from 2D edited images using our novel techniques. After optimization, users can sample from the conditional generative model to obtain diverse 3D editing results with high rendering quality and multi-view consistency. Our experiments demonstrate that GenN2N outperforms existing task-specific methods on various editing tasks, including text-driven editing, colorization, super-resolution, and inpainting, in terms of efficiency, quality, and diversity.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 3, 8

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 3, 6, 7, 8

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[4] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3981–3990, 2022. 2

[5] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 3

[6] Jun-Kun Chen, Jipeng Lyu, and Yu-Xiong Wang. Neuraleditor: Editing neural radiance fields via manipulating point clouds. *arXiv preprint arXiv:2305.03049*, 2023. 2

[7] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1475–1484, 2022. 2

[8] Ankit Dhiman, R Srinath, Srinjay Sarkar, Lokesh R. Boregowda, and R Venkatesh Babu. Corf: Colorizing radiance fields using knowledge distillation. *ArXiv*, abs/2309.07668, 2023. 7

[9] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 2, 3, 6, 7

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[11] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, Thomas Funkhouser, et al. Adversarial texture optimization from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020. 6

[12] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022. 2

[13] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2

[14] Xiaoyang Kang, Tao Yang, Wenqi Ouyang, Peiran Ren, Lingzhi Li, and Xuansong Xie. Ddcolor: Towards photo-realistic and semantic-aware image colorization via dual decoders. *arXiv preprint arXiv:2212.11613*, 2022. 6, 7

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 8

[16] Zhengfei Kuang, Fujun Luan, Sai Bi, Zhixin Shu, Gordon Wetzstein, and Kalyan Sunkavalli. Palettenerf: Palette-based appearance editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20691–20700, 2023. 2, 6, 7

[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 577–593. Springer, 2016. 7

[18] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4340–4350, 2023. 2

[19] Hao-Kang Liu, I Shen, Bing-Yu Chen, et al. Nerf-in: Free-form nerf inpainting with rgb-d priors. *arXiv preprint arXiv:2206.04901*, 2022. 2

[20] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023. 2

[21] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5773–5783, 2021. 2

[22] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. *arXiv e-prints*, pages arXiv–2302, 2023. 2

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 7

[24] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 8

[25] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image

translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[26] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 3

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 4

[29] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2, 8

[30] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 3

[31] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 8

[32] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 3

[33] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 3

[34] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6445–6454, 2022. 2, 7

[35] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 3, 7

[36] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022. 2

[37] Qiling Wu, Jianchao Tan, and Kun Xu. Palettenerf: Palette-based color editing for nerfs. *arXiv preprint arXiv:2212.12871*, 2022. 2

[38] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[39] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2

[40] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *arXiv preprint arXiv:2307.12348*, 2023. 7

[41] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–18, 2021. 2

[42] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 717–733. Springer, 2022. 2

[43] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[45] Chengwei Zheng, Wenbin Lin, and Feng Xu. Editablenerf: Editing topologically varying neural radiance fields by key points. *arXiv preprint arXiv:2212.04247*, 2022. 2

[46] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. *arXiv preprint arXiv:2306.13455*, 2023. 3