RePaint-NeRF: NeRF Editting via Semantic Masks and Diffusion Models

Xingchen Zhou[1,2], **Ying He**[1,2], **F. Richard Yu**[1,2], **Jianqiang Li**[1,3], **You Li**[2]
[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
[3]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University
zhouxingchen2021@email.szu.edu.cn, {heying, yufei, lijq}@szu.edu.cn, liyougis@gmail.com

Abstract

The emergence of Neural Radiance Fields (NeRF) has promoted the development of synthesized high-fidelity views of the intricate real world. However, it is still a very demanding task to repaint the content in NeRF. In this paper, we propose a novel framework that can take RGB images as input and alter the 3D content in neural scenes. Our work leverages existing diffusion models to guide changes in the designated 3D content. Specifically, we semantically select the target object and a pre-trained diffusion model will guide the NeRF model to generate new 3D objects, which can improve the editability, diversity, and application range of NeRF. Experiment results show that our algorithm is effective for editing 3D objects in NeRF under different text prompts, including editing appearance, shape, and more. We validate our method on both real-world datasets and synthetic-world datasets for these editing tasks. Please visit https://repaintnerf.github.io for a better view of our results.

1 Introduction

High-quality reconstruction of a complex 3D world is a critical challenge in computer vision [Aharchi and Ait Kbir, 2020]. Neural Radiance Fields (NeRF) [Mildenhall *et al.*, 2021] is an advanced approach for reconstructing the photo-realistic view of real 3D scenes. Nevertheless, most NeRF models implicitly encode the 3D scene by multiple layer perceptions (MLP) [Mildenhall *et al.*, 2021] or spherical harmonics [Fridovich-Keil *et al.*, 2022], the shape and appearance of the scene can only be seen after rendering, which means its content cannot be edited as we do in explicit scenarios. In the scenario of automatic driving, an automatic driving model requires a large amount of realistically simulated data for training [Li *et al.*, 2019]. NeRF can provide large volumes of high-fidelity data for self-driving training to alleviate the gap of Sim2Real [Tancik *et al.*, 2022]. However, there is still a certain distance for the current NeRF to produce numerous simulated data among the implicit scene, which greatly limits the scope of the application of NeRF. Therefore, editing within NeRF is essential in many cases.

In prior research studies [Kobayashi *et al.*, 2022; Yang *et al.*, 2021; Kundu *et al.*, 2022] that decompose the implicitly encoded scene for editing the objects in the scene by assigning semantic labels to each point in three-dimensional space. Then they can edit the content in NeRF by manipulating the collection of labeled points. For example, by setting the density of a car on the street to zero, it can be removed from the scene, or a red car can be set to blue by modifying its RGB value. However, these operations cannot meet creative editing needs, such as turning a pickup truck into a sedan, which requires models with strong generalization capabilities to change the shape and appearance of objects in space.

Recent works [Ramesh *et al.*, 2022; Rombach *et al.*, 2022] have shown very promising results in generating image content through text prompts. Users are now free to edit 2D images using text prompts and generate new images at a higher resolution. One of the keys is the thousands of rich images on the Internet, which enables the model to understand the content in the image and align with the abstract concepts in the language. In the three-dimensional domain, the lack of diverse 3D data limits the development of such generative models [Lin *et al.*, 2022].

A recent approach, DreamFusion [Poole *et al.*, 2022] integrates a 2D pre-trained diffusion model [Saharia *et al.*, 2022] with NeRF to generate 3D objects from text prompts. In more detail, they use an optimized gradient from the denoising process of diffusion model [Ho *et al.*, 2020] to update the NeRF model in the direction of the text prompt and finally obtain a 3D model that conforms to the description of the text prompt and ensures view-consistency. However, DreamFusion's high memory and time consumption limits its scalability, making it impractical for generating complex 3D scenes.

To tackle these problems, we propose a new framework for editing the content in NeRF from text prompts. More specifically, we first mask the area to be edited, under the guidance of the pre-trained text-to-image diffusion model [Poole *et al.*, 2022], we can modify the specified area from text prompts. However, manually smearing the mask on the two-dimensional training images cannot guarantee view consistency, and consumes a lot of time. Thus, we split our framework into two stages. In the first stage, based on the vanilla NeRF [Mildenhall *et al.*, 2021] encoding color and density, we additionally extend a semantic feature module to provide users with semantic target selection. These semantic features

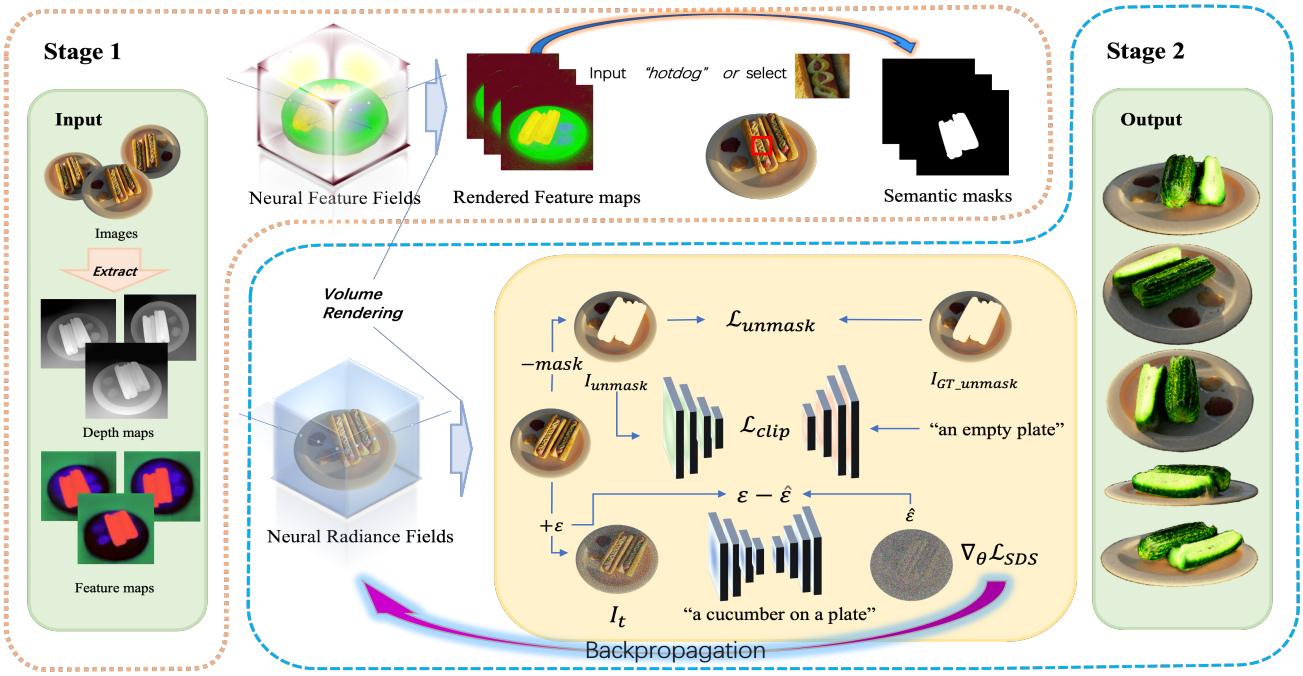


Figure 1: **Overview of RePaint-NeRF.** We present an editing method in NeRF. In the first stage, we additionally optimize a feature field along with the color module and density module to extract the content mask by using text or patch. In another way of speaking, we separate the part we want to change for a generation. Then, we use the mask and text prompt to generate the new content guided by the pre-trained diffusion model and CLIP model. After optimization of the generation, we can finally repaint a pre-trained NeRF model with view consistency and scene integrity.

are extracted from a pre-trained large-scale model, such as CLIP [Radford *et al.*, 2021]. After the first stage of training, users can select a patch to get the target object mask and generate new training data with mask information. In the second stage, based on DreamFusion [Poole *et al.*, 2022], we gradually modify the mask area to conform to the shape and appearance of the text prompts under the guidance of the diffusion model. Moreover, we find that if the newly generated object is smaller than the original object, the previously covered area will be exposed, but since this part of the content is unknown to the model, this area will become a black hole. To alleviate this problem, we additionally add a background prompt to guide the generation of the content of the black hole. However, we find that only adding a background prompt is not enough. Inspired by [Mirzaei *et al.*, 2022], we use CLIP [Radford *et al.*, 2021] to encourage the filling of this part. We name our method, RePaint-NeRF, which means that based on a pre-trained NeRF, we can recreate the content inside it. Our experiments on both real-world and synthetic datasets demonstrate the effectiveness of our method in changing the content in different scenes under various text prompts.

In summary, our contributions include:

- We propose a new framework that is capable of editing 3D content in NeRF through text prompts. To the best of our knowledge, we are the first work to propose editing NeRF using a diffusion model in complex scenes.
- Our method can greatly expand the scope of the application of NeRF model and apply it to most existing NeRF model architectures.
- Our approach enables practical semantic-masked object editing, making it possible for guiding editing in continuous NeRF scenes by diffusion models.

2 Related Work

Our method mainly utilizes a pre-trained text-to-image diffusion model [Rombach *et al.*, 2022] for NeRF editing. In this section, we mainly summarize some recent NeRF editing research and text-to-content generation works.

2.1 Neural Radiance Fields Editing

Neural Radiance Fields (NeRF) [Mildenhall *et al.*, 2021] uses a multi-layer perceptual layers network to encode complex scenes in a coordinate system-based manner and render high-quality 3D views in an end-to-end manner. A large amount of variant works [Barron *et al.*, 2021; Müller *et al.*, 2022; Fridovich-Keil *et al.*, 2022; Pumarola *et al.*, 2021] were released, setting off a wave of neural rendering. However, most NeRF variant works are based on implicit neural representations, which makes NeRF not as easy to edit as traditional explicit primitives, such as mesh. Some NeRF editing studies [Wang *et al.*, 2022; Liu *et al.*, 2021; Yuan *et al.*, 2022; Xu and Harada, 2022; Kobayashi *et al.*, 2022] recently are proposed to address this challenging issue.

Object-level NeRF Editing. Some of the NeRF editing works [Wang *et al.*, 2022; Liu *et al.*, 2021] focus on a single class of objects. For example, Editing-NeRF [Liu *et al.*, 2021] can change the shape or color of some parts of a certain class of objects, such as a chair or a car. Specifically, Editing-NeRF trains a neural network on a large number of a single category of objects, which is designed to learn the shape code and appearance code of these 3D models. Editing-NeRF can edit the shape and appearance of an object by adjusting these two latent codes. However, this method can only be operated on similar objects and can hardly extend the editing operation to complex scenes.

Neural Scene Decomposition. Another research direction [Kobayashi *et al.*, 2022; Kundu *et al.*, 2022; Zhi *et al.*, 2021] is to decompose the neural scene first and add semantic labels to each 3D coordinate point by additionally training a semantic branch so that a certain class of object can be selected to edit during rendering. However, such methods are limited in that they cannot extend editing to invisible content, such as turning a rock into an apple. Our work relies on a pre-trained diffusion model, which endows the power to regenerate a selected object in a complex scene.

2.2 Text-to-3D Generation

Recently, some methods [Poole *et al.*, 2022; Lin *et al.*, 2022; Metzer *et al.*, 2022; Jain *et al.*, 2022] have been proposed to transfer knowledge from pre-trained 2D diffusion models to 3D fields. DreamField [Jain *et al.*, 2022] uses a pre-trained CLIP [Radford *et al.*, 2021] model to supervise the gap between the views rendered from different perspectives represented by NeRF and a text prompt. However, the generated 3D models are still not photo-realistic. The recently proposed DreamFusion [Poole *et al.*, 2022] and its variants [Lin *et al.*, 2022; Metzer *et al.*, 2022] use pre-trained diffusion models to guide the generation of 3D models. The diffusion model generates gradients through its denoising mechanism [Nichol and Dhariwal, 2021; Ho *et al.*, 2020] by randomly looking at the 3D field, the gradient is then passed directly to the NeRF model for optimization. However, these approaches can not extend to scene-level generation due to memory limitations. In our paper, we use the diffusion model for the editing of different objects in the scene, achieving scene-level generation in a sense.

3 Method

The first part of our method is to mask the places we want to modify. However, it is too time-consuming to manually paint masks from different angles in a 3D scene. Thus, to get a view consistent semantic mask, we additionally train a semantic feature module to obtain a relative view continuous mask. In detail, we encode the 3D coordinate points to a high-dimension feature space. Moreover, we also add depth maps predicted from training images to supervise the depth of the predicted view, which could effectively reduce noise and speed up training [Deng *et al.*, 2022]. The second part of our framework is about how to generate a new object by text guidance based on the target mask. We also need to preserve the other existing content, so we add background prompts and a CLIP loss to monitor the plausibility of background content. We extract the pose information of the training views via COLMAP [Schonberger and Frahm, 2016]. For the supervised depth information, we use an existing robust model [Ranftl *et al.*, 2022] which could predict a rough depth of a monocular image. Note that the depth estimation model can be replaced by other models that predict depth relatively accurately. For the supervised semantic features, we refer to DFF [Kobayashi *et al.*, 2022], which distills the feature maps from a pre-trained CLIP model [Li *et al.*, 2022]. The overview of our framework is shown in Fig. 1.

3.1 Preliminaries

Neural Radiance Fields. NeRF [Mildenhall *et al.*, 2021] implicitly encodes the color $\boldsymbol{c}$ and density σ of each 3D point $\boldsymbol{x}$ from different view direction $\boldsymbol{d}$ by utilizing multiple perceptual layers $g_\theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$ weighted by θ. Considering a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ is emitted to sample the points along the ray in 3D space, where $\boldsymbol{o}$ is the origin of the ray, t is the distance from the origin $\boldsymbol{o}$ to the sample point $\boldsymbol{x}$ along the ray. The color of a pixel can be obtained by volume rendering:

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))c(\boldsymbol{r}(t), \boldsymbol{d})\mathrm{d}t, \tag{1}$$

$$\text{where} \quad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(t))\mathrm{d}t\right), \tag{2}$$

where $T(t)$ can be regarded as transparency, t_n and t_f are the near plane and the far plane of the sampling boundary. NeRF is obtained by optimizing the following loss function:

$$\mathcal{L}_{\text{color}} = \sum_{\boldsymbol{r}\in\mathcal{R}} \left\| C(\boldsymbol{r}) - \hat{C}(\boldsymbol{r}) \right\|^2, \tag{3}$$

where $\mathcal{R}$ represents all rays emitted from the pixels of training images.

DreamFusion. A recent work DreamFusion [Poole *et al.*, 2022] shows that under the guidance of a 2D diffusion model ϕ, a 3D implicit object represented by the NeRF g_θ can be generated from scratch according to a text prompt. In their method, the NeRF model g_θ renders an image I at a random viewing angle, the pre-trained diffusion model ϕ sample noise ϵ at time-step t to generate noisy image $I_t = I + \epsilon$. The main contribution of DreamFusion [Poole *et al.*, 2022] is that they proposed a gradient calculation by Score Distillation Sampling (SDS) loss [Poole *et al.*, 2022] to guide the update direction of NeRF:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, g_\theta) = \mathbb{E}_{t,\epsilon}\left[w(t)\left(\epsilon_\phi\left(I_t; y, t\right) - \epsilon\right)\frac{\partial I}{\partial \theta}\right], \tag{4}$$

where $w(t)$ is a weighted function correspond to time-step t, y is a text embedding, ϵ_ϕ is a learned denoising function. $\nabla_\theta \mathcal{L}_{\text{SDS}}$ is used to update the NeRF network g_θ instead of propagating to the diffusion model ϕ.

3.2 Semantic Mask Extraction

In our approach, the first step in modifying a 3D scene is to mask the target regions. Manually labeling each point in the training data or 3D field is very time-consuming. Obtaining the semantic information of a single two-dimensional image directly will lose the view consistency. Therefore, to extract accurate and consistent semantic information, we encode the feature $\boldsymbol{f}$ of view-independent point $\boldsymbol{x}$ in three-dimensional space. Note that the feature $\boldsymbol{f}$ is a high-dimensional vector with semantic information, which is different from explicit semantic labels in semantic segmentation. Here we follow DFF [Kobayashi *et al.*, 2022]. We first utilize an existing pre-trained model, such as CLIP [Radford *et al.*, 2021], to extract the feature $\boldsymbol{f}$ of each training data. We additionally trained a feature network $s : \boldsymbol{x} \rightarrow \boldsymbol{f}$ and finally obtained the features from different views but keep view consistency through volume rendering equation:

$$F(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))f(\boldsymbol{r}(t), d)\mathrm{d}t. \tag{5}$$

Similar to Eq. 3, we defined the loss function for optimizing the feature module:

$$\mathcal{L}_{\text{feature}} = \sum_{\boldsymbol{r}\in\mathcal{R}} \left\| F(\boldsymbol{r}) - \hat{F}(\boldsymbol{r}) \right\|^2. \tag{6}$$

However, in some views, there will always be some noise affecting NeRF's depth estimation, resulting in imprecise semantic mask segmentation (see Fig. 2). Thus, we add depth information that is predicted by a pre-trained model [Ranftl *et al.*, 2022] as a coarse depth supervision to mitigate this problem. At the same time, this measure also speeds up the convergence of the feature field:

$$\mathcal{L}_{\text{depth}} = \sum_{\boldsymbol{r}\in\mathcal{R}} \left\| D(\boldsymbol{r}) - \hat{D}(\boldsymbol{r}) \right\|^2, \tag{7}$$

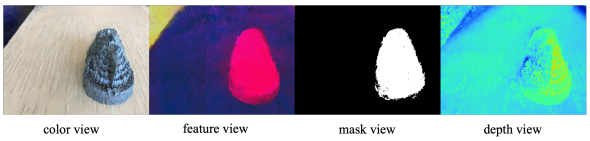
where $D(\boldsymbol{r})$ is the depth predicted by a pre-trained model as the depth ground truth, and $\hat{D}(\boldsymbol{r})$ is the depth predicted by NeRF. Therefore, the final loss function for the first stage of our method is:

$$\mathcal{L}_{\text{first-stage}} = \mathcal{L}_{\text{color}} + \lambda_{\text{feature}}\mathcal{L}_{\text{feature}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}. \tag{8}$$

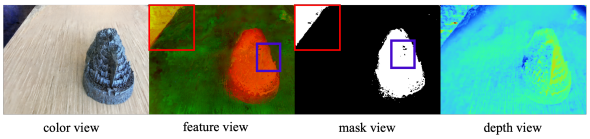
We do not obtain the semantic mask by comparing the features $\boldsymbol{f}$ of each 3D point $\boldsymbol{x}$ with the target feature. Instead, we compare the patch features with pixel features after rendering the feature map F_I. We prefer to use patch features instead of text features because we find that a higher threshold can be set to control more accurate segmentation. The mask is obtained using the following equation:

$$I_{\text{mask}}^{H\times W\times 1} = \mathbb{1}(Sim(F_{\text{patch}}, F_I) > \alpha), \tag{9}$$

where F_{patch} is the mean feature of a selected patch, Sim is a similarity function, and α is a threshold.



(a) fortress scene **without** depth supervision.



(b) fortress scene **with** depth supervision.

Figure 2: **Ablation study of depth supervision.**

3.3 Text-to-3D Content Editing

In the second stage, based on DreamFusion [Poole *et al.*, 2022] and the previously extracted masks, we modify the 3D content inside it on a pre-trained NeRF. Our goal is to keep the surrounding content unchanged while editing NeRF content. Thus, we keep optimizing the unmasked region by minimizing:

$$\mathcal{L}_{\text{unmask}} = \sum_{\boldsymbol{r}\in\mathcal{R}} \left\| C(\boldsymbol{r})_{\text{unmask}} - \hat{C}(\boldsymbol{r})_{\text{unmask}} \right\|^2, \tag{10}$$

$$\text{where} \quad C(\boldsymbol{r})_{\text{unmask}} = C(\boldsymbol{r}) \times \mathbb{1}(I_{\text{mask}} < 0.5), \tag{11}$$

$$\hat{C}(\boldsymbol{r})_{\text{unmask}} = \hat{C}(\boldsymbol{r}) \times \mathbb{1}(I_{\text{mask}} < 0.5). \tag{12}$$

However, we find that when the diffusion model [Rombach *et al.*, 2022] is used to guide the masked part to be modified, the newly generated small target object will expose the part covered by the previous object, which is largely unseen. Thus, we add a background prompt (BGT) that could partially solve this issue. For example, we use prompt *"a blue rose in leaves"* instead of *"a blue rose"*, so the BGT here is *"leaves"*. (see the first-row example in Fig. 3). Inspired by [Weder *et al.*, 2022], we also add a CLIP loss function to guide the masked region for generating a background that could fill the black hole around the mask edge. The CLIP loss function is defined as:

$$\mathcal{L}_{\text{clip}} = -Sim\left(Z_I, Z_{\text{BGT}}\right), \tag{13}$$

where Z_I and Z_{BGT} are the latent feature of rendering image I and background prompt encoded by pre-trained CLIP model [Li *et al.*, 2022].

The insight of our method is that keep the diffusion model watch the whole scene by using $\nabla_\theta\mathcal{L}_{\text{SDS}}$ to guide the update direction of NeRF g_θ for optimizing the target region, and using $\mathcal{L}_{\text{unmask}}$ to ensure the unmasked region of the target keep stable. Besides, the background prompt and CLIP loss function $\mathcal{L}_{\text{clip}}$ is to make the unseen region to be filled.

Now the final loss function of our second stage is defined as:

$$\mathcal{L}_{\text{repaint}} = \lambda_{\text{unmask}}\mathcal{L}_{\text{unmask}} + \lambda_{\text{clip}}\mathcal{L}_{\text{clip}}, \tag{14}$$

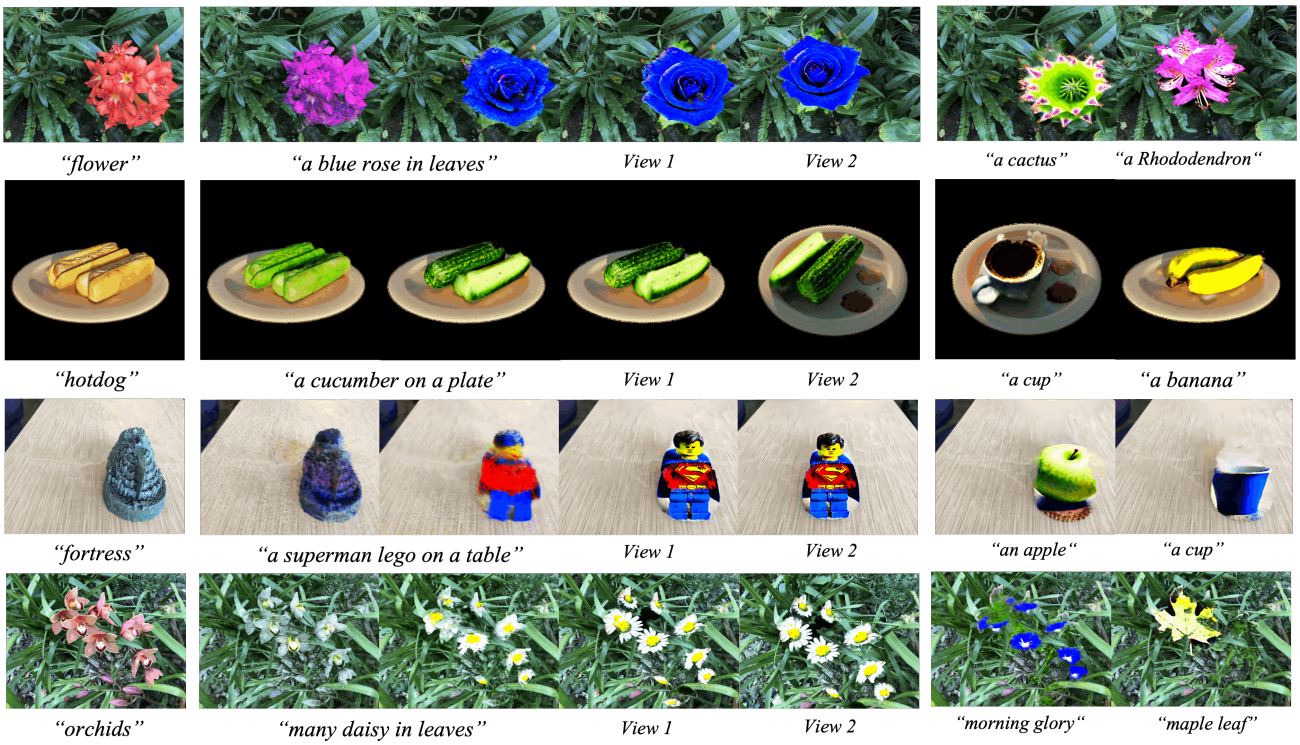


Figure 3: **Qualitative editing results.** We test our method on the Blender and LLFF. Our method can change the shape and appearance of objects in both real-world and synthetic-world datasets with a simple text prompt while maintaining almost high fidelity.

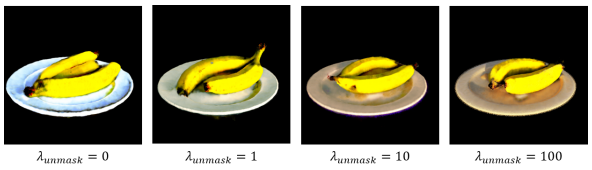


Figure 4: **Comparison of different λ_{unmask}.** The text prompt and background prompt we use here are *"a banana in a plate"* and *"an empty plate"*.

where the λ_{unmask} and λ_{clip} is set here for balancing the change of the target object and other regions.

3.4 Implementation Details

Our implementation is divided into two parts, based on DFF [Kobayashi *et al.*, 2022] and DreamFusion [Poole *et al.*, 2022] separately. Each training iteration needs to render a whole view, this undoubtedly consumes a huge amount of memory, and the training speed is also crucial to user experience. Thus, we use Instant-NGP [Müller *et al.*, 2022] as our NeRF model, which is based on a multi-resolution hash grid structure to accelerate the training and rendering process. We test our method on a single NVIDIA RTX 3090 GPU. It is worth noting that the actual generation time depends on the appearance and shape of editing before and after. For example, if you only change the color of one car, it might only take 5 minutes. But turning a car into a chocolate candy car can take more than 40 minutes to achieve decent results. Please refer to our supplementary material for more details.

Mask Extraction. In the first stage, we first extract the semantic masks of the training images. CLIP [Radford *et al.*, 2021] is a text-image pair-based multimodal model for self-supervised training, which is trained for aligning text information and image information. However, the feature maps extracted by CLIP are not at the pixel level, so here we use the LSeg [Li *et al.*, 2022] to extract the feature maps as our training set. LSeg [Li *et al.*, 2022] is a semantic segmentation model that is trained based on the CLIP weights. All feature maps are interpolated to the size of image size $H \times W \times 512$. We visualize the high dimensional feature via PCA, which can reflect the distribution of semantic information to a certain extent. In addition, for depth information, we use MiDAS [Ranftl *et al.*, 2022], a robust monocular depth estimation model, to extract rough depth information. The size of each depth image is $H \times W \times 1$, and the depth information is normalized to a range of 0 to 1. Thanks to Instant-NGP [Müller *et al.*, 2022] and the depth information we added as supervision, we only need to train 2000 steps for each scene to get a clear semantic mask. We use Adam [Kingma and Ba, 2014] to optimize our NeRF model in the first stage, with a learning rate 1e-2 and batch size 4096.

3D Object Editing. In the second stage, we only use the color images and the masks extracted in the first stage as inputs. We first train a NeRF by optimizing Eq. 3 as our base model. In the pre-training phase, we sample a whole image in each iteration, so that we can observe the memory usage

(a) Mask + CLIP

(b) Mask + SDS

Figure 5: **Comparison of different guidance.** We compare the ability of generations under two different guidance. Here we use text prompt *"a blue rose"* in both tests.

and adjust the hyperparameters in time to make compromises for the pre-trained diffusion model and CLIP model in the generation phase. For example, in the fortress scene, we set the bound to 1.4 and the learning rate to 1e-3 to ensure that the GPU does not exceed the video memory space as much as possible. For the training of generation, each scene will require an input of an object text prompt with a background text prompt. We use Stable Diffusion [Rombach *et al.*, 2022] to supervise the update direction of the NeRF model and use the CLIP [Radford *et al.*, 2021] model to compare the similarity between the unmasked image (an image that includes the surrounding content and a masked blacked hole) and the BGT. We train these two phases using Adan with a learning rate of 1e-3 decaying to 1e-4 and a batch size of 1. These two phases are optimized for 3000 steps and 10000 steps respectively, more steps will be added for better visual effect.

4 Experiments

In this section, we focus on evaluating our method on different scenes with different prompts, we show qualitative editing results on Local Light Field Fusion (LLFF) [Mildenhall *et al.*, 2019] and Blender, followed by comparison experiments and ablation studies.

4.1 Datasets

We use Local Light Field Fusion (LLFF) [Mildenhall *et al.*, 2019] and Blender for testing. The LLFF [Mildenhall *et al.*, 2019] is collected from the real world in the form of shooting forward-facing, and its capture resolution is 4032×3024. The Blender comes from the synthetic world by rendering on 3D models, its resolution is 800×800. In order to save the memory of the GPU, we resize the image size to 504×378 and the image size of Blender to 400×400. Our experiments show that our method is very effective for object editing in both worlds.

4.2 Semantic Mask Extraction

The role of the first part is mainly to replace the operation of manual masking and provides view-consistent masks for the second part. Thus, we mainly focus on how to get a clean and accurate mask view faster. In the Fig. 2, we take the fortress scene as an example, and our goal is to extract the objects of the fortress. The result shows that training for only 2000 steps without depth supervision produces inaccurate and noisy semantic masks. However, in our proposed model with depth informative supervision, 2000-step training yields more accurate and clean semantic masks.

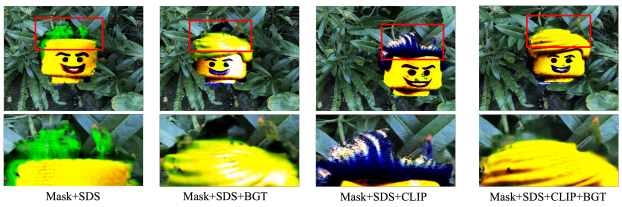


Figure 6: **Ablation study of background prompt guidance.** For **Mask+SDS**, the text prompt is *"a lego man head"*; For **Mask+SDS+BGT**, the text prompt is *"a lego man head in leaves"*; For **Mask+SDS+CLIP**, the text prompt is *"a lego man"*, and the prompt for CLIP is *"leaves"*; For **Mask+SDS+BGT+CLIP**, the text prompt is *"a lego man head in leaves"*, and prompt for CLIP is *"leaves"*.

4.3 Text-to-3D Editing

We have conducted a lot of experiments on LLFF [Mildenhall *et al.*, 2019] and Blender, as shown in Fig. 3, our method can recreate a photo-realistic target area while ensuring that the other original scene does not change or changes slightly. The first column of Fig. 3 is the view rendered by pre-trained NeRF. The text under the view of the first column is the target object we desire to change. Columns 2, 3, and 4 represent the process of gradually modifying the object. The text prompt is set below those three views, which includes the target prompt and the BGT. The BGT we send for the CLIP model is to encourage background generation. Columns 4 and 5 are different views that are rendered by modified NeRF. Columns 6 and 7 contain our other editing results under the same scene, the text below there is the target generation prompt, and the background prompts are consistent with the previous ones. We find that the generation of similar shapes works better and faster, such as turning a flower into a blue rose in the first row and a hotdog into a cucumber in the second row. For shapes that are not too similar, the generated effect will decrease, but it will gradually approach the content of the prompt, such as turning a hotdog into a cup of coffee in the first row and a fortress into an apple in the third row. At the same time, in our method, the model can perceive the surrounding objects and generate areas that have not been seen before. For example, in the showcase of the fortress to an apple, the edge of the fortress that previously covered part becomes material and color similar to the desktop. In the example of orchids to maple leaves, except for the modified maple leaf, other content that was orchids before has become the content of the grass in the background.

Comparison of CLIP guidance and SDS guidance. We compare the ability of Stable Diffusion [Rombach *et al.*,

2022] and CLIP [Radford *et al.*, 2021] to guide object changes under the same mask. Fig. 5 shows the editing effect of the two models during the optimization process. The target text prompt here we set is *"a blue rose"*, and the red flower is the source object. We first set Eq. 10 to ensure that the surrounding content remains unchanged. For **Mask+CLIP**, we make sure that the CLIP model [Radford *et al.*, 2021] will only see the region of the flower and we train the NeRF model by optimizing the similarity loss function of Eq. 13. We find that the CLIP model is weak to guide the NeRF model to the target direction very well. At the beginning of the training, the CLIP model is trying to make the flower move to a little bit blue, while after several epochs, the flower becomes an unreasonable texture. For **Mask+SDS**, we set the SDS loss [Poole *et al.*, 2022] to guide the masked part to become a blue rose. The result of **Mask+SDS** in the Fig. 5 shows that the model generates an excellent texture and shape. The comparison shows that the **Mask+SDS** can produce a much better effect than **Mask+CLIP**.

Ablation study of background prompt guidance. We perform background-cued ablation experiments (see Fig. 6). For **Mask+SDS**, we only set an SDS loss [Poole *et al.*, 2022] to guide the masked region with an object text prompt, here we set is *"a lego man head"*. The result shows that the model generates a shape like green hair on the Lego head, and the generated shape of the hair is still the shape of the petal. For **Mask+SDS+BGT**, here we set the text prompt for the diffusion model as *"a lego man head in leaves"*, which means we explicitly tell the diffusion model what is the content around the Lego man. And the result is that hair of the Lego man is more reasonable, but the previously covered part on the top of the hair becomes a black hole. For **Mask+SDS+CLIP**, we remove the background prompt in the text prompt and provide a background prompt to CLIP, we are inspired by [Mirzaei *et al.*, 2022] here. We give a view of the unmasked region to CLIP to fill the black hole. The result of this way is that the Lego man owns reasonable hair but with the shape of a petal. For the final **Mask+SDS+BGT+CLIP**, we set the text prompt includes the background prompt, which is *"a lego man head in leaves"*. And the CLIP will receive a view of the unmasked image and a background prompt *"leaves"*. The final results show that the hair becomes a normal shape, and the black hole is filled with a kind of green material. We also could see the result of **Mask+SDS+BGT+CLIP** in Fig. 3, for example, some green leaves appeared around the blue rose in the example of flower to a blue rose.

Effects of Different Mask Weight. In addition, we compare the effect of different λ_{unmask} in Eq. 14 (see Fig. 4). The Stable Diffusion [Poole *et al.*, 2022] model we use in our method receives full-resolution images without any mask. Thus, we set a loss function Eq. 10 to strongly control the unmasked part to avoid unwanted content. We use the hotdog scene as an example, and the goal is to change the hotdog to a banana. In the first experiment, we set $\lambda_{\text{unmask}} = 0$, which means no constraints on the background part (i.e., the plate), and the result is the diffusion model totally changes the shape and appearance of the plate. Then we gradually increase the weight λ_{unmask} to control the shape of the plate, the result shows that $\lambda_{\text{unmask}} = 100$ could retain the original shape and appearance of the plate well. Thus, in the practice of all scenes, we usually set $\lambda_{\text{unmask}} = 100$, we also find a weight too larger will extremely influence the speed of generation, which means more constraint to the ability of the diffusion model.

Figure 7: **Limitation of restricted view angle.** The daisy in the **red** frame is composed of two previous orchids; the daisy in the **yellow** frame is normal.

5 Limitations

Although our method can modify the content of NeRF through text prompts, there are still some limitations. The first and most obvious problem with our method is that the whole process is time-consuming and space-consuming. For this reason, we cannot extend to a larger resolution of data or a larger scene. Several recent works [Lin *et al.*, 2022; Metzer *et al.*, 2022] have introduced how to solve the time problems of DreamFusion [Poole *et al.*, 2022], but it still takes more than 30 minutes to generate a relatively high-quality 3D model at the object level. In another aspect, we find that in some cases, especially the new object shape is hugely different from the old one, the training process will be very tough for the model (see Fig. 3, the cup from the hotdog is not completely changed). The last limitation is that angle constraint will influence the shape of the generation. For example, in Fig. 7, the generated daisy on the edge of the view is actually combined by two orchids, while the generated daisy in the central location is much more normal.

6 Conclusion

In this paper, we proposed a NeRF editing framework based on the pre-trained 2D diffusion model guidance. Our method mainly obtained the edited scene by regenerating the masked part through the gradient guidance of the diffusion model. Moreover, we added a background prompt and a CLIP loss to ease the problem of invisible background. At the same time, we also added depth information as supervision in the semantic mask acquisition part of the task for a faster training speed and better semantic masks. Our method combined the semantic information of the scene and used a text-driven method to modify the content of the scene. Compared with explicitly modifying the 3D scene, our method would make the NeRF more diverse and promote the development of the simulation environment based on neural rendering. We believe that there will be faster and better editing methods in the future to further improve the editability of NeRF models.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62002238 and 62271324, Shenzhen Science and Technology Program under Grant ZDSYS20220527171400002, the Guangdong "Pearl River Talent Recruitment Program", and the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant GML-KF-22-26.

References

[Aharchi and Ait Kbir, 2020] M Aharchi and M Ait Kbir. A review on 3d reconstruction techniques from 2d images. In *Proceedings of the Third International Conference on Smart City Applications*, pages 510–522, 2020.

[Barron *et al.*, 2021] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[Deng *et al.*, 2022] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[Fridovich-Keil *et al.*, 2022] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[Jain *et al.*, 2022] 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.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kobayashi *et al.*, 2022] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022.

[Kundu *et al.*, 2022] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022.

[Li *et al.*, 2019] Wei Li, CW Pan, Rong Zhang, JP Ren, YX Ma, Jin Fang, FL Yan, QC Geng, XY Huang, HJ Gong, et al. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28):eaaw0863, 2019.

[Li *et al.*, 2022] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[Lin *et al.*, 2022] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

[Liu *et al.*, 2021] 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.

[Metzer *et al.*, 2022] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022.

[Mildenhall *et al.*, 2019] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[Mildenhall *et al.*, 2021] 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.

[Mirzaei *et al.*, 2022] Ashkan Mirzaei, Yash Kant, Jonathan Kelly, and Igor Gilitschenski. Laterf: Label and text driven object radiance fields. In *Proceedings of European Conference on Computer Vision*, pages 20–36, 2022.

[Müller *et al.*, 2022] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171, 2021.

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

[Pumarola *et al.*, 2021] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[Radford *et al.*, 2021] 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 *Proceedings of International Conference on Machine Learning*, pages 8748–8763, 2021.

[Ramesh *et al.*, 2022] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[Ranftl *et al.*, 2022] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[Rombach *et al.*, 2022] 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.

[Saharia *et al.*, 2022] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[Schonberger and Frahm, 2016] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer vVsion and Pattern Recognition*, pages 4104–4113, 2016.

[Tancik *et al.*, 2022] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[Wang *et al.*, 2022] 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.

[Weder *et al.*, 2022] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. *arXiv preprint arXiv:2212.11966*, 2022.

[Xu and Harada, 2022] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *European Conference on Computer Vision*, pages 159–175, 2022.

[Yang *et al.*, 2021] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.

[Yuan *et al.*, 2022] 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.

[Zhi *et al.*, 2021] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.