Edit-DiffNeRF: Editing 3D Neural Radiance Fields using 2D Diffusion Model

Lu Yu
James Cook University
lu.yu@my.jcu.edu.au

Wei Xiang*
La Trobe University
w.xiang@latrobe.edu.au

Kang Han
La Trobe University
k.han@latrobe.edu.au

Abstract

Recent research has demonstrated that the combination of pretrained diffusion models with neural radiance fields (NeRFs) has emerged as a promising approach for text-to-3D generation. Simply coupling NeRF with diffusion models will result in cross-view inconsistency and degradation of stylized view syntheses. To address this challenge, we propose the Edit-DiffNeRF framework, which is composed of a frozen diffusion model, a proposed delta module to edit the latent semantic space of the diffusion model, and a NeRF. Instead of training the entire diffusion for each scene, our method focuses on editing the latent semantic space in frozen pretrained diffusion models by the delta module. This fundamental change to the standard diffusion framework enables us to make fine-grained modifications to the rendered views and effectively consolidate these instructions in a 3D scene via NeRF training. As a result, we are able to produce an edited 3D scene that faithfully aligns to input text instructions. Furthermore, to ensure semantic consistency across different viewpoints, we propose a novel multi-view semantic consistency loss that extracts a latent semantic embedding from the input view as a prior, and aim to reconstruct it in different views. Our proposed method has been shown to effectively edit real-world 3D scenes, resulting in 25% improvement in the alignment of the performed 3D edits with text instructions compared to prior work.

1 Introduction

With the growing prevalence of efficient 3D reconstruction techniques like Neural Radiance Fields (NeRFs), generating photo-realistic synthetic views of real-world 3D scenes has become increasingly feasible [10]. Meanwhile, the demand for manipulating 3D scenes is rapidly increasing due to the broad range of content re-creation use cases [26]. A number of recent works have introduced extensions for editing NeRFs by e.g., operating on explicit 3D representations [30], training models to enable color modifications or the removal of certain objects [9], or utilizing diffusion models to perform text-to-3D synthesis [15]. These advancements have expanded the versatility of implicit volumetric representations. However, they are currently limited to changing the color and shape, or require complex and computationally expensive training procedures for each editing task.

Editing NeRFs is a crucial and challenging task. As NeRF is an implicit representation optimized per scene, the standard tools used for editing explicit representations cannot be directly employed [24, 27]. Moreover, in contrast to 2D image editing, where a single image input is able to provide sufficient guidance for manipulation, NeRFs require multi-view information to guide their content modification [20]. EditingNeRF [9] was the first attempt to enable editing on the implicit radiance field. To learn the geometry and appearance of various models, they incorporate the shape code and color code for better disentanglement. Nevertheless, EditingNeRF is limited to shape and color manipulation, or the removal of local parts of an object. Yang et al. [28] introduce a technique that

---

*Corresponding author.

combines NeRF with explicit 3D geometry representations, to enable more efficient and accurate reconstruction of 3D objects. Yuan et al. [30] further improve the performance by enabling users to perform user-controlled shape deformation in a 3D scene. However, their rendered results still suffer from significant shortcomings in terms of photorealism and geometric accuracy.

Recently, diffusion models have been proposed as a promising approach in incorporating NeRF-like models to produce multi-view consistent images [17]. This is done either by optimizing NeRFs via a pretrained diffusion model such as DreamFusion [15], or via an unconditional generative diffusion model that can be trained with 2D images [7]. Although these approaches can generate 3D models from any given text prompts, they currently lack fine-grained control over the synthesized views. More importantly, they cannot be directly used to edit real-captured NeRFs of fully observed 3D scenes. Haque et al. [3] propose Instruct-NeRF2NeRF, which extracts shape and appearance features from a pretrained 2D diffusion model (i.e., InstructPix2Pix [1]) to gradually edit the rendered images while optimizing the underlying 3D scene. This method, however, shares several limitations with InstructPix2Pix [1], such as significant multi-view inconsistencies and notable rendering artifacts including noise and blurring.

Our method. To solve the aforementioned problems and perform high-fidelity text-to-3D editing with pretrained NeRFs, we propose a framework dubbed Edit-DiffNeRF to edit the semantic latent space within frozen pretrained diffusion models from an input prompt. Which enables us to make fine-grained modifications to the rendered views of NeRF, offering enhanced control and accuracy in the optimization process of an underlying scene. Our Edit-DiffNeRF is composed of a frozen diffusion model, a proposed delta module to edit the latent space of a diffusion model, and a NeRF. Our proposed framework for text-to-3D editing involves two key steps. First, in order to perform the editing for the rendered views, we utilize the diffusion prior to generate a latent semantic embedding for each view and then apply our proposed delta module to edit the embedding. This delta module is optimized using a CLIP distance loss function. After training, it is able to produce edited images based on the input text instruction. Second, we proceed to train the NeRF using those edited images, leveraging the modifications made through the delta module. In order to ensure the multi-view consistency of a 3D scene, we propose a multi-view semantic consistency loss to reconstruct consistent latent features in the latent space from different views. To evaluate its effectiveness, we evaluate our Edit-DiffNeRF on a variety of real-captured NeRF scenes published by [3]. Extensive experimental results demonstrate 25% improvement in the alignment of the performed 3D edits with the text instructions compared to Instruct-NeRF2NeRF [3].

2 Related work

2.1 Editing NeRFs

NeRFs [10] have emerged as a prominent approach for rendering photo-realistic views from multiple views of 3D scenes. Recently, there has been ongoing research focusing on advancing the capabilities of editing NeRFs. One strategy is to integrate physics-based inductive biases into the training procedure of NeRFs, thereby enabling effective modifications in materials or scene lighting [22, 25, 12]. Alternatively, different approaches have been proposed involving the utilization of bounding boxes and conditions encoded on NeRFs, to allow easy manipulations of different objects [13, 29]. For instance, EditingNeRF [9] was the first attempt to enable editing on the implicit radiance field, where the 3D scene is conditioned via a shape code and an appearance code. This approach enables modifications to be made to the shape of objects in the scene as well as the colors. NeRF-Editing [30] is another work for performing geometry editing on 3D scenes, which provides an intuitive interface for making fine-grained modifications to the 3D shape and structure of NeRF scenes. A recent work, Instruct-NeRF2NeRF [3], proposes a method for editing NeRF scenes with text prompts based on a pretrained diffusion framework InstructPix2Pix [1]. However, it employs a diffusion model that operates on a single view at a time, which can lead to similar artifacts encountered in InstructPix2Pix, such as the presence of duplicated or distorted objects. In this work, we instead focus on enabling the manipulation of the latent semantic space of a pretrained diffusion model, thereby allowing the editing of rendered views of NeRF and facilitating the optimization of a 3D NeRF scene.

2.2 3D content generation

The recent advancements in pretrained large-scale models have significantly accelerated progress in the field of generating 3D content from scratch, allowing for the fast and effective creation of 3D scenes. Some works optimize NeRFs by vision-language models such as CLIP [8]. CLIP-NeRF [26] introduces a disentangled conditional NeRF that allows individual control over both shape and appearance via a learned deformation field. Dream Fields [6] combines the powerful CLIP with an optimization-based process to train NeRFs. CLIP has also been used in the recent state-of-the-art model DreamFusion [15] with a loss derived from the distillation of a 2D diffusion model to render high-fidelity coherent 3D objects. While these approaches are able to generate 3D scenes based on arbitrary text inputs, they often fall short of generalizing to real-world scenes. In parallel, recent works like SparseFusion [32] have delved into the concept of grounding by utilizing one or a few input views, effectively hallucinating the unseen parts of a scene. However, all of these generative models encounter the same challenge, i.e., consolidating the diverse and inconsistent outputs of a 2D diffusion model into a unified and consistent 3D scene. In our work, we aim to edit a real-captured 3D scene by an edited latent semantic embedding. Furthermore, we propose a novel multi-view semantic consistency loss to ensure 3D consistency across different viewpoints of a scene.

3 Preliminaries

3.1 Denoising diffusion probabilistic models (DDPMs)

Given a set of training views $\left\{\boldsymbol{x}^i\right\}_{i=1}^N \in \mathcal{I}$ for a 3D scene, The goal of generative models, e.g., Denoising Diffusion Probabilistic Models (DDPMs) [5] is to optimize the parameters θ of a model that closely approximates the data distribution $p(\boldsymbol{x})$. DDPM proposes to learn the data distribution by gradually transforming a sample from a tractable noise distribution toward a target distribution. Diffusion models typically include a deterministic forward process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that gradually adds noise to the sample such that:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \quad t \in (0, T], \tag{1}$$

where β_t is the t-th variance schedule. The model then learns the reverse (denoising) process with a neural network $\mathcal{D}_\theta(\boldsymbol{x}_t, t)$, which performs denoising steps by progressively removing noise and predicts $\hat{\boldsymbol{x}}_0$ from $\boldsymbol{x}_t$. The denoising process similarly uses a Gaussian distribution:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)), \tag{2}$$

where μ_θ and Σ_θ are the mean and variance, respectively.

3.2 Latent diffusion models

Latent diffusion models [17] obtain efficiency improvements compared to DDPMs [5] by leveraging the latent space of a pretrained variational autoencoder. In particular, given an input image $\boldsymbol{x}^i$, the forward process adds noise to the encoded latent embedding $\boldsymbol{z} = \varepsilon(\boldsymbol{x}^i)$, where $\varepsilon(\cdot)$ is an encoder and $\boldsymbol{z}_t$ is the noisy latent embedding at timestep t. The neural network $\mathcal{D}_\theta(\cdot)$ is optimized to predict the presented noise based on image and text instruction conditioning inputs. Formally, the latent diffusion objective is expressed as follows:

$$\mathcal{L} = \mathbb{E}_{\varepsilon(\boldsymbol{x}^i), \varepsilon(\boldsymbol{C}_I), \boldsymbol{c}_T, \epsilon \sim \mathcal{N}(0,1), t} \left[||\epsilon - \mathcal{D}_\theta(\boldsymbol{z}_t, t, \varepsilon(\boldsymbol{C}_I), \boldsymbol{c}_T)||_2^2 \right], \tag{3}$$

where $\boldsymbol{C}_I$ is the conditioned image, $\boldsymbol{C}_T$ is the text editing instruction, and $\hat{\epsilon}_t = \mathcal{D}_\theta(\boldsymbol{z}_t, t, \varepsilon(\boldsymbol{C}_I), \boldsymbol{c}_T)$ is the predicted noise at timestep t. Once trained, the estimated latent $\hat{\boldsymbol{z}}_{t-1}$ can be derived with a noisy input $\boldsymbol{z}_t$ and a predicted noise $\hat{\epsilon}_t$ at timestep t.

4 Method

Given a pretrained NeRF scene along with the input text instruction, we aim to edit the NeRF scene by manipulating a pretrained and frozen diffusion model as a score function estimator. Which allows us to produce an edited version of the NeRF scene that matches the provided edit instruction. Our pipeline of Edit-DiffNeRF is illustrated in Fig. 1, which is a two-stage framework. In the first stage, we train the proposed delta module to edit the latent space of a pretrained diffusion model. Which enables us to make fine-grained modifications to the rendered views and effectively consolidate these instructions in a 3D scene via NeRF training in the second stage.
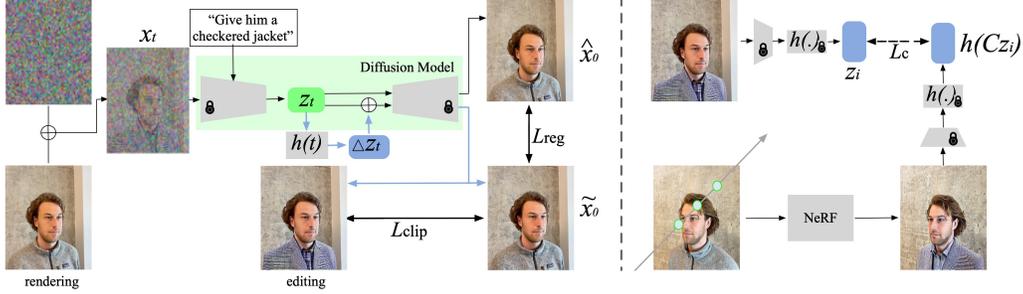
Figure 1: Our pipeline of Edit-DiffNeRF, which is a two-stage framework consisting of a frozen diffusion model, a proposed delta module, and a NeRF. In the first stage, we train the delta module $h(t)$ to edit the latent space of a pretrained diffusion model. After training, it is able to produce edited images based on the input text instruction. Then we freeze the weights of the delta module and train the NeRF using those edited images, leveraging the modifications made through the delta module.

4.1 Problem

In order to supervise the NeRF reconstruction process for editing a 3D scene, many researchers have proposed to utilize a 2D diffusion model for synthesizing the edited outputs [15]. Alternatively, others train a conditional diffusion model by learning $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{C}_T)$ [11]. The experiment results have demonstrated that $\mu_\theta(\boldsymbol{x}_t, t, \boldsymbol{C}_T)$ is closer to the target mean $\widehat{\mu}_\theta(\boldsymbol{x}_t, t)$ than $\mu_\theta(\boldsymbol{x}_t, t)$ [31]. This empirical evidence suggests that incorporating conditional inputs can be beneficial to bridge the gap between the predicted posterior mean and the corresponding ground-truth value. However, training diffusion models in image space for each condition can be a cumbersome and impractical task.

Furthermore, an intuitive idea to edit $\boldsymbol{x}$ is simply updating $\boldsymbol{x}_t$ to optimize NeRF based on the score distillation sampling loss $\nabla \mathcal{L}_{SDS}$ as proposed in DreamFusion [15]. However, it results in a 3D scene with more artifacts or distorted views (see Fig. 2). We believe this is due to a pretrained diffusion model with the fixed semantic latent space may not achieve feasible results in realistic scenarios [21].
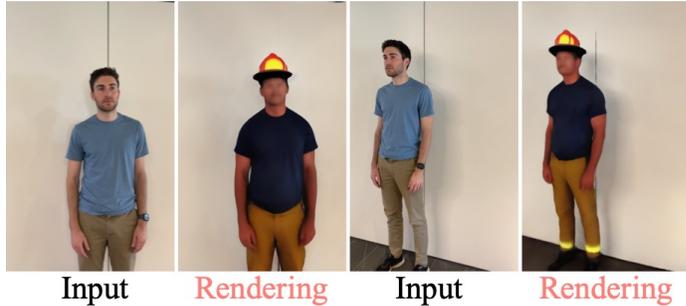

Figure 2: Rendering results with a trained NeRF based on score distillation loss. Edits were performed with a text instruction "Turn him into a firefighter with a hat".

4.2 Editing semantic latent space in diffusion models

In order to manipulate the latent semantic space with the input text instruction and circumvent the process of training an entire diffusion model, we propose to utilize a delta module $h(t)$, which learns the shifted latent semantic space $\Delta \boldsymbol{z}_t$ given a frozen and pretrained diffusion model and offers significantly reduced computational demands. $h(t)$ is implemented as a small compact neural network with two convolutional layers. Which have the same number of channels as the bottleneck layer of the U-Net in the diffusion model. Formally, the parameterization for $\mu_\theta(\boldsymbol{x}_t, t)$ given the delta module $h(\cdot)$ becomes:

$$\mu_\theta(\boldsymbol{x}_t, t, \Delta \boldsymbol{z}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\mathcal{D}_\theta(\boldsymbol{z}_t|\Delta \boldsymbol{z}_t, t, \boldsymbol{\varepsilon}(\boldsymbol{C}_I), \boldsymbol{c}_T)\right). \quad (4)$$

Essentially, by introducing the information of $\boldsymbol{c}_T$ to the latent semantic space with $\Delta \boldsymbol{z}_t$, the predicted $\hat{\epsilon} = \mathcal{D}_\theta(\boldsymbol{z}_t|\Delta \boldsymbol{z}_t, t, \boldsymbol{\varepsilon}(\boldsymbol{C}_I), \boldsymbol{c}_T)$ is modified. Which, in turn, produces the shifted mean value

$\mu_\theta(\boldsymbol{x}_t, t, \Delta \boldsymbol{z}_t)$ to bridge the gap and facilitate the reverse process of reconstructing the provided instructional information in the samples.

We utilize the architecture of CLIP [16], which comprises an image encoder $\boldsymbol{\varepsilon}_I$ and a text encoder $\boldsymbol{\varepsilon}_T$ that project inputs to a shared latent space. Building on this, we propose a cross-modal CLIP distance function to evalute the cosine similarity between the input text instruction and the edited image:

$$\mathcal{L}_{\mathrm{clip}} = 1 - \langle \boldsymbol{\varepsilon}_T(\boldsymbol{t}_{\mathrm{src}}) - \boldsymbol{\varepsilon}_I(\boldsymbol{t}_{\mathrm{tgt}}), \boldsymbol{\varepsilon}_I(\boldsymbol{x}_{\mathrm{src}}) - \boldsymbol{\varepsilon}_I(\boldsymbol{x}_{\mathrm{tgt}}) \rangle, \tag{5}$$

where the input image and its text description are represented by $\boldsymbol{x}_{\mathrm{src}}$ and $\boldsymbol{t}_{\mathrm{src}}$. The text instruction and the edited image are denoted as $\boldsymbol{t}_{tgt}$ and $\boldsymbol{x}_{tgt}$, respectively, and $\langle \cdot \rangle$ is the cosine similarity operator. In addition, we add an L1 loss to regulate the produced $\boldsymbol{x}_0$ from the original input and the edited one by:

$$\mathcal{L}_{\mathrm{reg}} = \lambda_{\mathrm{reg}} |\hat{\boldsymbol{x}}_0 - \widetilde{\boldsymbol{x}}_0|, \tag{6}$$

where $\hat{\boldsymbol{x}}_0$ is obtained via the frozen diffusion prior, $\widetilde{\boldsymbol{x}}_0$ is generated with the modified latent embedding, and λ_{reg} is the hyper-parameter.

4.3 View-consistent rendering

Another fundamental challenge when it comes to editing a 3D scene with a 2D diffusion model is that the diffusion models lack 3D awareness, which leads to a generated NeRF with inconsistent and distorted views. Seo et al. [20] tried to close this gap by utilizing viewpoint-specific depth maps from a coarse 3D structure. However, this embedding needs to optimize a 3D model for each text prompt. This is a computationally intensive process that can still produce blurred and distorted images despite optimization.

To account for the inconsistency challenge, we propose to encode the latent semantic embedding for each view using the pretrained diffusion model and our proposed delta function $\boldsymbol{h}(\cdot)$. Specifically, given a pretrained diffusion model and the optimized $\boldsymbol{h}(\cdot)$, each rendered image is encoded into a latent embedding $\boldsymbol{z}_i$. Inspired by conditional NeRFs [26], the color $\boldsymbol{c}$ in NeRF [10] is extended to a latent-dependent emitted color $\boldsymbol{c}_{\boldsymbol{z}}$:

$$(\sigma, \boldsymbol{c}_{\boldsymbol{z}_i}) = F_\theta(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{z}_i), \tag{7}$$

where the embedding $\boldsymbol{z}_i$ is a conditional input, and F_θ is the NeRF model.

4.4 Multi-view semantic consistency loss

To achieve our goal, the optimized delta module $\boldsymbol{h}(\cdot)$ is supposed to produce consistent latent semantic embeddings as much as possible across different camera poses. Therefore, we propose a novel multi-view semantic consistency loss, denoted as $\mathcal{L}_c$. Which involves using a latent embedding $\boldsymbol{z}_i$ extracted from image $\boldsymbol{x}^i$ as the conditional input to reconstruct images from different views. The formal expression of our proposed loss $\mathcal{L}_c$ is given as follows:

$$\mathcal{L}_c = ||\boldsymbol{h}(\boldsymbol{C}_{\boldsymbol{z}_i}) - \boldsymbol{z}_i||_1, \tag{8}$$

where $\boldsymbol{C}_{\boldsymbol{z}_i}$ is the rendered output based on Eq. (7). Thus the total loss for training a NeRF becomes:
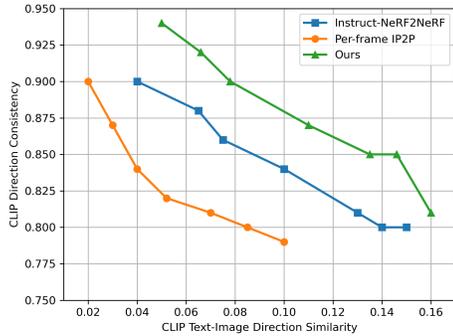
$$\mathcal{L} = \mathcal{L}_{\mathrm{photo}} + \lambda_c \mathcal{L}_c, \tag{9}$$

where λ_c is the hyper-parameter. By leveraging this strategy, our method ensures consistency across different views while also preserving the integrity of the underlying latent embeddings.
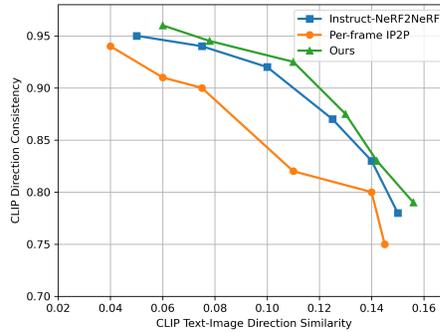
5 Experiments

In this section, we first introduce the datasets and implementation details in Section 5.1. Subsequently, we evaluate the performance of our approach against other methods in Section 5.2. Finally, to further understand the impact of the key designs, we conduct ablation studies in Section 5.3.

5.1 Experimental setup

| (a) Results with larger training datasets | (b) Results with smaller training datasets (less than 100 images) |

Figure 3: We plot the trade-off between the CLIP Direction Consistency and the CLIP Text-Image Direction Similarity. For both metrics, higher is better.

Table 1: Quantiative evaluation on real-captured scenes

	CLIP Text-Image Direction Similarity	CLIP Direction Consistency
Per-frame IP2P [1]	0.1603	0.8185
SDS w/ IP2P [1]	0.0266	0.9160
One-time DU [3]	0.1157	0.8823
Instruct-NeRF2NeRF [3]	0.1600	0.9191
Ours	**0.2031**	**0.9376**

Datasets. We conduct 3D editing on a set of scenes with varying degrees of complexity, including 360-degree scenes of environments and objects, faces, and full-body portraits that are released by [3]. These scenes were captured using two types of cameras: a smartphone and a mirrorless camera. We use the camera poses that were extracted via the COLMAP [18]. Following CLIP-NeRF [26], we also evaluate the effectiveness of our approach on two publicly available datasets: Photoshape [14] with a collection of 150K chairs and Carla [2, 19] consisting of 10K cars.

Table 2: FID scores

	Chairs		Cars	
	Before	After	Before	After
EditNeRF [9]	36.8	40.2	102.8	118.7
CLIP-NeRF [26]	47.8	48.4	66.7	67.8
Ours	**32.1**	**32.5**	**48.6**	**49.0**

Implementation details. We choose the official InstructPix2Pix [1] as our diffusion prior, which contains a large-scale text-to-image latent diffusion model StableDiffusion [17]. During the training stage, we uniformly sample timesteps ranging from $t = 1$ to $T = 1000$ for all experiments. The variances of the diffusion process are linearly increased, starting from $\beta_1 = 0.00085$ and reaching $\beta_1 = 0.012$. We optimize our proposed delta module $h(t)$ using 50 steps for each view. The training process requires approximately 15 minutes and is performed on four RTX 3090 GPUs. After training, we use the edited images as the supervision to train our NeRF. As the underlying NeRF implementation, we use the nerfacto model from NeRFStudio [23], which is a recommended real-time model tuned for real captures. We follow the training strategy in NeRFStudio and the NeRFs are optimized for 30000 steps with L1 and directional CLIP losses in [16].

Metrics. It should be noted that unlike dynamic NeRF methods, acquiring ground truth views for view synthesis results after editing poses significant challenges, particularly when performing with real scenes. This is primarily because the edited views as a product of user manipulation do not physically exist. Following the evaluation metrics employed in Instruct-NeRF2NeRF [3], we evaluate two crucial quantitative metrics, namely (1) CLIP Text-Image Direction Similarity, i.e., the alignment between the edited 3D views and the corresponding text instruction, and (2) CLIP

Direction Consistency, the temporal consistency of the edit across multiple views [1]. Besides, we compute the Fréchet Inception Distance (FID) scores [4] for 2000 rendered images before and after the editing process, which allows us to quantitatively assess the quality and fidelity of the edited scenes.
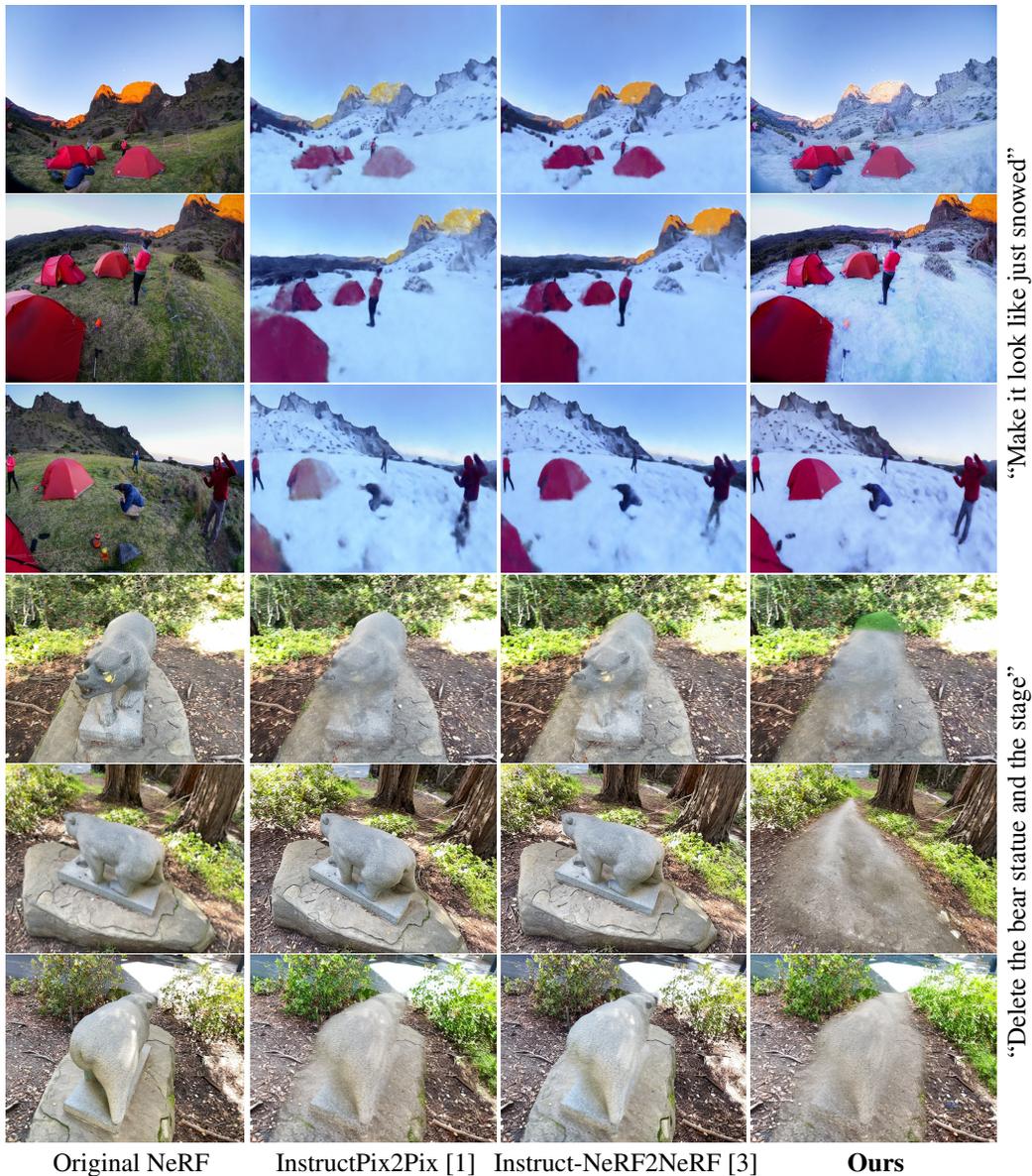

Figure 4: Visual comparisons with a collection of recent state-of-the-art methods.

5.2 Quantiative evaluation

Table 1 and Table 2 show the superiority of Edit-DiffNeRF over other state-of-the-art methods on real scenes. We observe that by learning the optimal delta module to edit the latent semantic space, our Edit-DiffNeRF can have notably higher CLIP Text-Image Direction Similarity and Consistency. In Fig. 3, we also plot the trade-off between the CLIP Direction Consistency and the CLIP Text-Image Direction Similarity over two scenes. As these two metrics compete with each other, when the degree to which the output images align with the desired edit increases, the consistency with the input image decreases. As is observed from Fig. 3, compared to the recent state-of-the-art method Instruct-

NeRF2NeRF [3], the CLIP Direction consistency obtained by our Edit-DiffNeRF is significantly higher, even with similar CLIP Text-Image Direction Similarity values. Furthermore, we observe that when the training dataset for a scene is smaller (consisting of less than 100 images), both the Instruct-NeRF2NeRF [3] and our method yield similar results, with lower directional similarity.

In Table 2, we report the FID scores for measuring the image quality of synthesized views before and after editing. To calculate the FID scores for rendered images, we employ a set of 2000 randomly selected test images. Subsequently, we apply various edit instructions similar to CLIP-NeRF [26] to these images and recompute the FID scores for the edited results. On the chair dataset, When evaluated on the chair dataset, EditNeRF [9] demonstrates improved performance in terms of reconstruction compared to CLIP-NeRF [26]. However, it is worth noting that the quality of the edited images noticeably decreases after the editing process. When evaluated on the car dataset, CLIP-NeRF [26] exhibits a significant improvement over EditNeRF [9] in terms of reconstruction quality before and after editing. Finally, compared to those two methods, our Edit-DiffNeRF not only greatly improves the overall quality but also effectively preserves the image quality after the editing process.

Editing results. We show edited results rendered from different views in Fig. 4 for real-captured scenes. For comparison, we also show the original NeRF rendering results under the same views before editing. In Fig. 4, the first set is a campsite scene. We edited it by a text instruction "Make it look like just snowed".

Figure 5: Comparisons of editing results between CLIP-NeRF [26] and our Edit-DiffNeRF.

On the one hand, as is observed from Fig. 4, the generated results via the InstructPix2Pix [1] are with ambiguity on what exactly to edit and exhibit considerable inconsistencies across different views. On the other hand, although the Instruct-NeRF2NeRF [3] appears to produce feasible views, some of them tend to show significant variance, resulting in a 3D scene that is blurry and highly distorted. Instead, the rendered multiple views obtained via our Edit-DiffNeRF show its capability to effectively edit real-world scenes, surpassing the achievements of previous approaches by delivering photo-realistic results while maintaining 3D consistency. In the second set of Fig. 4, the images were edited by instruction "Delete the bear statue and the stage". According to the figure, neither InstructPix2Pix [1] nor Instruct-NeRF2NeRF [3] are able to obtain the intended editing results effectively. This limitation primarily arises from the incapacity of InstructPix2Pix to handle large spatial manipulations.

Moreover, we present additional experimental results along with NeRF rendering results in Fig. 5 and Fig. 6 to further illustrate our findings. The comparisons in Fig. 5 were performed with a text instruction "Turn it into a car with red front wheels and green rear wheels". It can be noted that CLIP-NeRF [26] is not able to handle fine-grained edits. Instead, our edited result (right column) is quite consistent with the instruction. In Fig. 6, we can further observe the presence of in-
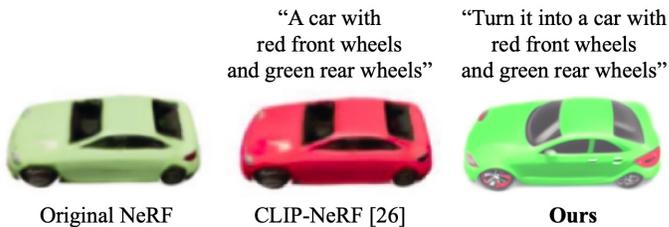

Figure 6: Comparison with Instruct-NeRF2NeRF [3]. Edits were performed with a text instruction "Give him a checkered jacket".

8

consistent edits within Instruct-NeRF2NeRF [3] that fail to consolidate in a 3D scene (middle column). Instead, our model provides a superior solution (right column), rendering consistent results and allows for significant visual manipulations.

5.3 Ablation study

We validate the effectiveness of our Edit-DiffNeRF by conducting a comprehensive comparative analysis between our approach and several other variants.

Impact of delta module. We first compare our model with against Instruct-NeRF2NeRF [3]. We use the official code released by [1] and fine-tune the entire UNet model in InstructPix2Pix [1] to edit real images. Table 3 demonstrates that our pro-

Table 3: Ablation study results

	CLIP Text-Image Direction Similarity	CLIP Direction Consistency
Instruct-NeRF2NeRF [3]	0.1120	0.7805
Ours w/o $\mathcal{L}_c$	0.1703	0.9198
Ours	**0.2031**	**0.9376**

posed Edit-DiffNeRF outperforms InstructPix2Pix in all aspects. We attribute this superiority to the fact that fine-tuning the entire model for each scene is challenging, thereby resulting in inferior results.

Impact of multi-view semantic consistency loss. In addition, we perform experiments where we exclude the consistency loss $\mathcal{L}_c$. As is observed from Table 3, even in the absence of the multi-view semantic consistency loss $\mathcal{L}_c$, our method still surpasses Instruct-NeRF2NeRF [3] with a trained delta module. Nevertheless, without this loss, our method still lacks 3D consistency and only achieves a slight performance gain. In contrast, incorporating this loss yields substantial improvements in the results.

6 Conclusion

In this paper, we have outlined the underlying challenges in achieving accurate NeRF scene modifications with pretrained 2D diffusion models. To address this limitation, we introduce the Edit-DiffNeRF framework, which specifically targets editing the semantic latent space within pretrained diffusion models. Specifically, the Edit-DiffNeRF framework is devised to learn latent semantic directions using a delta module, guided by provided text instructions, which allows for the effective consolidation of these instructions within a 3D scene through NeRF training. Furthermore, we introduce a multi-view semantic consistency loss to ensure semantic consistency across different views. Extensive experiments demonstrate that our approach consistently and effectively enables edits across a wide range of real-captured scenes. Moreover, it significantly improves the text-image consistency of the edited results.

7 Limitations

Despite the encouraging performance gained by our work, it suffers from two limitations. First, our model is subject to the visual quality of the rendered images generated using NeRF techniques, as well as the diffusion model's ability to generalize to arbitrary edits. Second, the quality of edited images decreases when the input images have low resolution or are out of focus.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–15, 2023.

[2] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *CoRL*, pages 1–16, 2017.

[3] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3d scenes with instructions. *arXiv preprint 2303.12789*, 2023.

[4] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–38, 2018.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–12, 2020.

[6] 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 (CVPR)*, pages 857–866, 2022.

[7] Animesh Karnewar, Andrea Vedaldi, David Novotný, and Niloy J. Mitra. HOLODIFFUSION: training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–13, 2023.

[8] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH*, pages 25:1–25:8, 2022.

[9] 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 (ICCV)*, pages 5753–5763, 2021.

[10] 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. In *ECCV*, pages 405–421, 2020.

[11] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. DiffRF: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–11, 2023.

[12] Jacob Munkberg, Wenzheng Chen, Jon Hasselgren, Alex Evans, Tianchang Shen, Thomas Müller, Jun Gao, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2856–2865, 2021.

[14] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: photorealistic materials for large-scale shape collections. *ACM Trans. Graph.*, 37(6):192, 2018.

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

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[17] 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 (CVPR)*, pages 10674–10685, 2022.

[18] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[19] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[20] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023.

[21] Changhao Shi, Haomiao Ni, Kai Li, Shaobo Han, Mingfu Liang, and Martin Renqiang Min. Exploring compositional visual generation with latent classifier guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2023.

[22] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7495–7504, 2021.

[23] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.

[24] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas J. Guibas. Deformation-aware 3d model embedding and retrieval. In *ECCV*, pages 397–413, 2020.

[25] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd E. Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490, 2022.

[26] 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 (CVPR)*, pages 3825–3834, 2022.

[27] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Hang Yu, Wei Liu, Xiangyang Xue, and Yu-Gang Jiang. Pixel2Mesh: 3d mesh model generation via image guided deformation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3600–3613, 2021.

[28] 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 (ICCV)*, pages 13759–13768, 2021.

[29] Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *ICLR*, 2022.

[30] 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 (CVPR)*, pages 18332–18343, 2022.

[31] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[32] Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling view-conditioned diffusion for 3d reconstruction. *arXiv preprint arXiv:2212.00792*, 2022.