Efficient-NeRF2NeRF: Streamlining Text-Driven 3D Editing with Multiview Correspondence-Enhanced Diffusion Models

Liangchen Song[1], Liangliang Cao[1], Jiatao Gu[1], Yifan Jiang[12], Junsong Yuan[3], Hao Tang[1]
[1]Apple [2]UT Austin [3]University at Buffalo

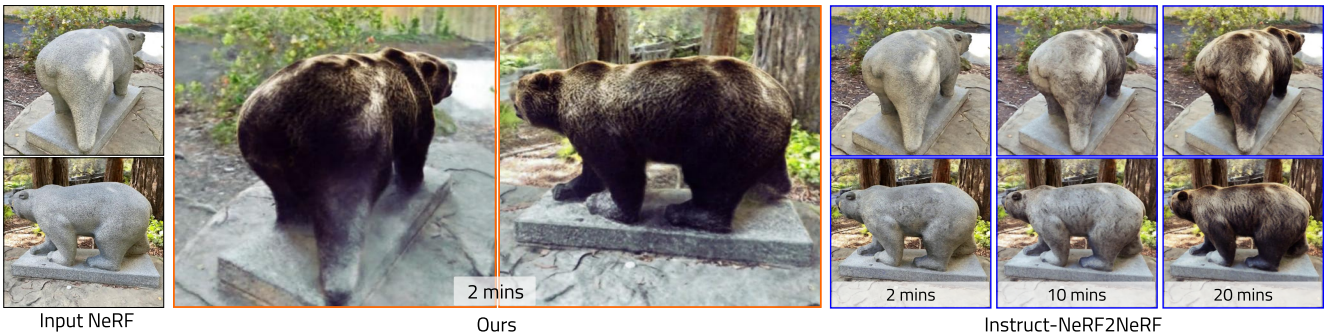


Figure 1. **Efficient NeRF editing within 2 minutes**. We present a framework that aims to enhance the efficiency of editing NeRF models using text-based instructions. The key factor contributing to this efficiency is our regularized diffusion scheme, which enables the direct generation of multiview-consistent images. (Prompt: *"Turn the bear into a grizzly bear"*.)

Abstract

The advancement of text-driven 3D content editing has been blessed by the progress from 2D generative diffusion models. However, a major obstacle hindering the widespread adoption of 3D content editing is its time-intensive processing. This challenge arises from the iterative and refining steps required to achieve consistent 3D outputs from 2D image-based generative models. Recent state-of-the-art methods typically require optimization time ranging from tens of minutes to several hours to edit a 3D scene using a single GPU. In this work, we propose that by incorporating correspondence regularization into diffusion models, the process of 3D editing can be significantly accelerated. This approach is inspired by the notion that the estimated samples during diffusion should be multiview-consistent during the diffusion generation process. By leveraging this multiview consistency, we can edit 3D content at a much faster speed. In most scenarios, our proposed technique brings a 10× speed-up compared to the baseline method and completes the editing of a 3D scene in 2 minutes with comparable quality. Project page: https://lsongx.github.io/projects/en2n.html.

1. Introduction

The recent accomplishments in foundational 2D editing methods [3, 6, 56] have enabled us to personalize and modify 3D captured scenes [22], which hold great appeal and significant practical value. However, despite the impressive results obtained through these recent developments, the existing 3D editing methods [22, 59, 79] still suffer from the drawback of prolonged optimization duration, which often takes tens of minutes.

Achieving efficient 3D editing using text-driven 2D image editors presents a challenging task. This difficulty primarily arises from the fact that text-driven image editors typically operate on a per-image basis. Consequently, applying these editors to multiview images often results in ineffective correspondence among views, hence the edited 2D images cannot be directly leveraged for updating 3D content. Although significant advancements have been made in enhancing multiview correspondence, such as score-distillation sampling [50], existing approaches frequently necessitate retraining and backpropagation, thereby making multiview editing inefficient.

This paper proposes an approach wherein we apply regularization to the diffusion denoising process to enhance its multiview correspondence. Performing direct edits on a collection of multi-view images leads to efficient 3D editing capabilities compared to existing works like Instruct-

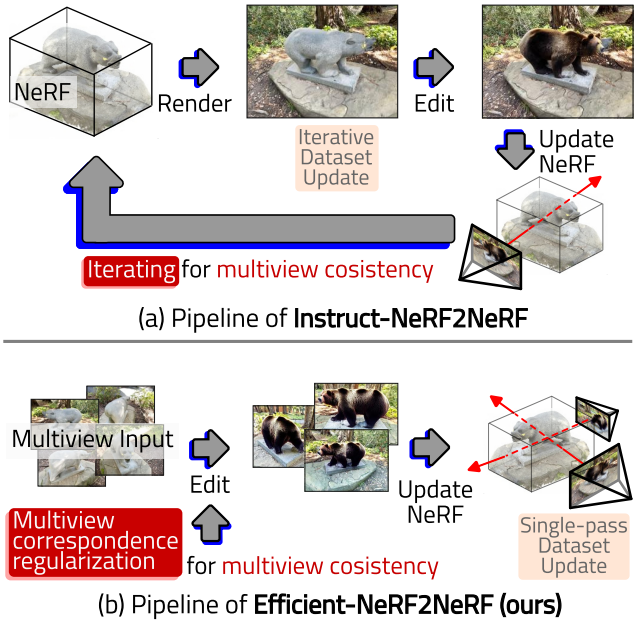


Figure 2. **Comparison of editing frameworks.** Our approach significantly enhances the speed of the editing process by directly updating the NeRF using edited multiview images. In contrast, the SoTA method, Instruct-NeRF2NeRF, needs to iteratively edit and re-render images.

NeRF2NeRF [22] (see Fig. 2). This is because the iterative updating framework [22] can now be simplified by treating the process as a text-driven editing of 2D images and then updating the 3D representation.

Our key insight is that in 3D editing, where multiple-view images serve as inputs, we benefit from the observations of the existing correspondences between these images. If we can progressively rectify the inconsistencies observed across multiple views during the diffusion sampling process, we can effectively generate a collection of images that exhibit strong multiview consistency and subsequently update the 3D contents in a much more efficient way. Note that the idea of multiview consistency is also used in 3D generation tasks like text-to-3D [50] or image-to-3D [86] where correspondence needs to be generated.

However, ensuring consistent prediction across various perspectives is a challenging problem due to two reasons. Firstly, the color of a 3D point may vary depending on the viewing angle. Consequently, naively imposing identical estimations leads to unrealistic outcomes. Secondly, prevalent 2d diffusion models denoise in the latent space, where each latent vector represents an image patch. However, such latent representation cannot best capture the variations of image patches across different viewpoints.

We address the aforementioned challenges through a two-fold approach: mitigating the impact of regularization for diffusion and minimizing the influence of inconsistent edits for training radiance fields. To tackle the regularization issue, we draw inspiration from recent studies that reveal how the diffusion process prioritizes overall image structure in the early denoising stages while focusing on texture details in the later stages [45]. Based on this insight, we propose a regularization strategy that places more emphasis on reducing inconsistency during the early steps of diffusion, and gradually removes the regularization towards the later steps. By softening the regularization, we can generate visually appealing images. However, the generated images may still not strictly consistent across multiviews. To overcome this, we adopt a loss function that encourages the 3D content optimization process to align the style of image patches using the Gram matrix [19] and randomly switch between the photometric loss and the style loss. In our method, this loss function guides the edited 3D to minimize the impact of multiview inconsistency outputs, such as shifts or deformations. To sum up, our contributions are as follows:

- This paper proposes a text-driven 3D content editing framework, utilizing the power of 2D image editing techniques. This framework grants users the ability to efficiently edit 3D content, 10 times faster than current radiance field editing techniques such as Instruct-NeRF2NeRF [22], which may require approximately 20 minutes.
- This paper develops a regularization technique to preserve the multiview correspondence across a collection of images, eliminating the requirements for retraining the diffusion network. Moreover, to mitigate the impact of inconsistent generation on the tuning of the radiance field, we propose the incorporation of a style matching loss.

2. Related Works

Generating 3D Contents with Foundational 2D Generative Models. 3D editing can be conceptualized as a conditioned generation challenge. In recent years, foundational 2D generative models [3] have showcased remarkable prowess in open-set generation [55–57]. The utilization of these 2D generative foundational models for open-set 3D content generation has garnered significant attention. DreamFusion [28, 50] proposed Score Distillation Sampling (SDS) based on probability density distillation to get 3D generation priors from 2D generative models. Score Jacobian Chaining [80] applied chain rule on the learned gradients and back-propagate the score of a 2D diffusion model through the Jacobian of a differentiable renderer for 3D content. Given that 2D images are conventionally generated on a view-by-view basis, the central quandary in this avenue of research revolves around the identification of a multifaceted supervisory signal that ensures 3D consistency from the purview of 2D models, which can be improved in various aspects such as on the 2D generation conditioning

[2, 9, 11, 26, 38, 51, 60, 81, 84, 97] and on the underlying 3D representation [13, 37, 61, 73, 76, 91]. Besides pure text as conditioning, some methods are developed to take a single image as the input [43, 51, 54, 62, 74, 85, 86]. The aforementioned methods primarily tackle a common challenge: the task of updating 3D representations in the presence of potentially multiview inconsistent images. Within our approach, we incorporate a style loss component to mitigate the issue of multiview inconsistency. This strategy is employed with the specific aim of facilitating 3D editing, which diverges notably from the task of generating content from scratch.

Multiview Consistent Image Generation. Since multiview consistency stands as the cornerstone in producing high-quality 3D content, several methods are dedicated to refining multiview consistency in the process of 2D image generation. A prevalent research direction posits the augmentation of 2D image generation models with a heightened awareness of the imaging process, such as camera poses. 3DiM [82] proposed to use camera poses as a conditioning input to the denoising model in diffusion. Zero-1-to-3 [40] further improved this line of work with foundational 2D generative models and effective advanced multiview consistency modeling. One-2-3-45 [39] further accelerated the sampling and reconstruction process of Zero-1-to-3. Instead of modeling the view-conditioned distribution of images, Viewset Diffusion [71] and SyncDreamer [42] proposed to model the distribution of multiview images directly. MVDiffusion [75] and Consistent-1-to-3 [89] used epipolar geometry based attention across the views to improve multiview consistency, while MVDream [63] demonstrated the effectiveness of directly using the self-attention in Stable Diffusion [56]. Our research distinguishes itself from the methodologies above by focusing on 3D editing, wherein we presuppose correspondence as an input to the generative process. In contrast, the previously mentioned approaches necessitate the generation of correspondence due to the absence of 3D inputs.

Radiance Fields Editing. Radiance Fields [1] serve as a means for representing 3D content, and they can be parameterized through neural networks [46, 48], multiplane images [83, 96], point clouds [30, 87], and others [8, 12, 58, 69, 90]. Since a radiance field represents a 3D scene, editing it can be achieved with human-designed graphics knowledge, such as volume geometry [36, 52, 77, 88, 92–94], color [20, 32, 33] and lighting [4, 5, 68]. Some works [14, 16, 24, 27, 49, 95] proposed to stylize a radiance field with 2D image style adaptors [19, 25], with an emphasis on the texture generation and leaving the geometry untouched. EditNeRF [41] proposed to update the latent embeddings of NeRF for editing both shape and appearance. ClipNeRF [78], NeRF-Art [79] and Blended-NeRF [21] adopted CLIP [53] to maximize the similarity between NeRF and the given text descriptions. DFFs [31] distilled image feature from DINO [7] and LSeg [34] into radiance fields and allows direct editing of properties such as shape, size and color. More closely related to our work, Instruct-NeRF2NeRF [22] adopted the 2D instruction-based editing method InstructPix2Pix [6] to update the scene with the Iterative Dataset Update (Iterative DU) technique. Besides iteratively updating the dataset [22, 47, 70], the iterative loss SDS [50] was adopted by some methods for text-based 3D editing, such as Vox-E [59], FocalDreamer [35], AvatarStudio [44] and DreamEditor [98]. DN2N [18] accelerates existing editing methods by training a generalizable NeRF so that each editing prompt will not require extra training steps like the above iterative methods. Our editing method employs a distinct pipeline, which obviates the necessity for iterative updates to edit 3D scenes, as the proposed regularization ensures the multiview consistency of the edited images. Notably, our method exhibits a marked efficiency compared to the aforementioned technique and often finishes within minutes. Concurrent with our work, Gaussian splatting representation is adopted to accelerate the radiance editing [10, 17], and we consider their methods orthogonal to us since we speed up the editing on the diffusion side.

3. Preliminaries

Diffusion Models. Diffusion models [15, 23, 65, 66] aim to approximate a given data distribution $q(\boldsymbol{x}_0)$ by learning a model distribution $p_\theta(\boldsymbol{x}_0)$ that is both a good approximation of $q(\boldsymbol{x}_0)$ and allows for efficient sampling. A significant contribution in this field is the development of Denoising Diffusion Probabilistic Models (DDPMs [23, 64, 67]), which belong to latent variable models described by the following form:

$$p_\theta(\boldsymbol{x}_0) = \int p_\theta(\boldsymbol{x}_{0:T}) \mathrm{d}\boldsymbol{x}_{1:T} \tag{1}$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are latent variables in the same sample space as $\boldsymbol{x}_0$ (denoted as $\mathcal{X}$), and $p_\theta(\boldsymbol{x}_{0:T}) := p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. In DDPM [23], we have $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I})$ and $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \sigma_t^2 \mathbf{I})$, where the trainable component $\mu_\theta(\boldsymbol{x}_t, t)$ has the time-dependent constant variance σ_t^2. The forward process is then defined to learn μ_θ as $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, where $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I})$ with β_t as predefined constants. Next, in DDPM we have

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \right), \tag{2}$$

where α_t and $\bar{\alpha}_t$ are constants derived from β_t and $\boldsymbol{\epsilon}_\theta$ is a noise predictor, usually parameterized by neural networks.

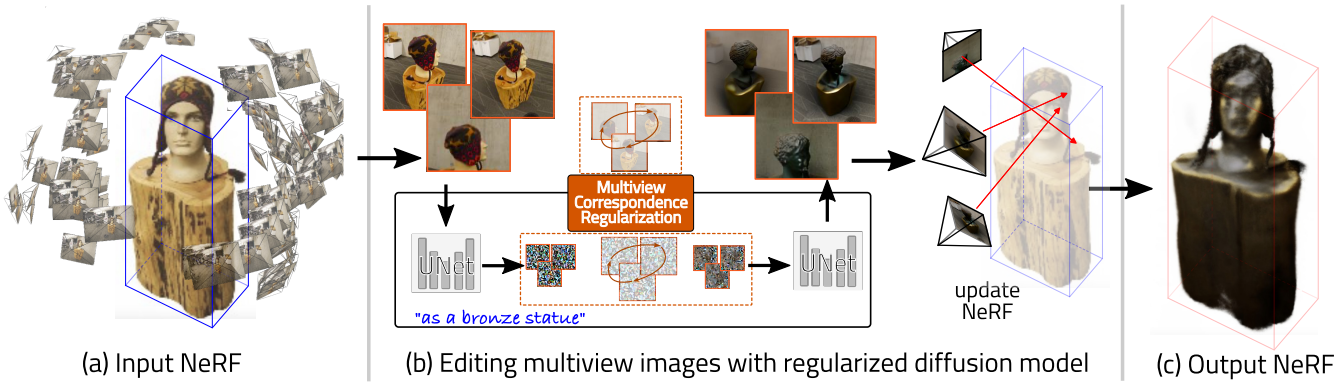


Figure 3. **The overall framework of multiview correspondence regularized diffusion.** We regularize that the output obtained during the denoising process of diffusion aligns with the input multiview images in terms of multiview correspondence

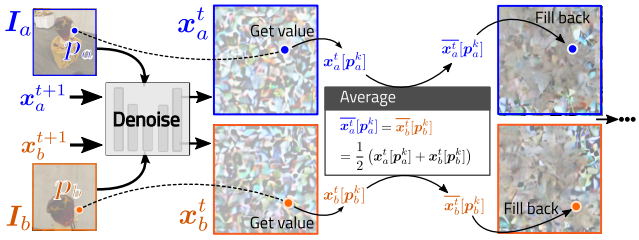


Figure 4. **Multiview correspondence regularization on the diffusion process.** $\boldsymbol{I}_a$ and $\boldsymbol{I}_b$ are input views with correspondence pair $(\boldsymbol{p}_a, \boldsymbol{p}_b)$. $\boldsymbol{x}_a^t$ and $\boldsymbol{x}_b^t$ are the samples at time t as defined in Eq. (1). "Denoise" step refers to Eq. (2).

In image generation tasks, the generation can be conditioned on inputs like text or reference images, and we have

$$\boldsymbol{\epsilon}^t = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}), \tag{3}$$

where $\boldsymbol{c}$ is the conditioning inputs.

Radiance Fields. 3D contents in radiance fields are represented by functions that take in spatial location (x, y, z) and viewing direction (θ, ϕ) and output the radiance and occupancy of that location. The radiance value and occupancy for all points along a camera ray are considered to render an image from the radiance fields. Volume rendering [46] facilitates the generation of radiance fields by accessing and analyzing all points present on the camera rays. Recent advancements, such as the utilization of 3D Gaussian points [30] to parameterize a radiance field, have led to more efficient rendering pipelines. we opt to generalize the rendering process by symbolically representing it as

$$\boldsymbol{I} = \pi(\mathcal{F}, \mathbf{P}), \tag{4}$$

where $\boldsymbol{I}$ is the rendered image, $\mathcal{F}$ is the radiance field, and $\mathbf{P}$ is the camera projection matrix (*i.e.*, intrinsics and extrinsics).

4. Our Method

The overall framework of our proposed method is straightforward. First, we edit multiview images by the provided textual prompt. Subsequently, we proceed to update the radiance field with these edited images. Our approach differs from Instruct-NeRF2NeRF in two key respects. Firstly, while Instruct-NeRF2NeRF performs image editing on one image at a time, we apply it to a batch of multiview images. Secondly, we employ a single-pass dataset update scheme, whereas Instruct-NeRF2NeRF requires iterative updates to the dataset.

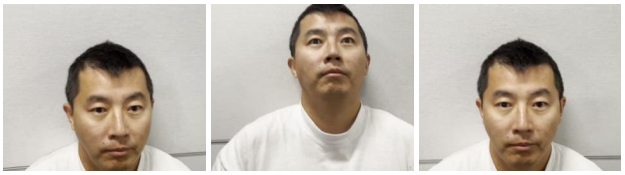
4.1. Multiview Correspondence Regularization

The goal of correspondence regularization is to maintain the correspondence among the multiview images after editing. Our method for pursuing this objective is quite clear: *we aim to ensure that the samples $\boldsymbol{x}_{0:T}$ during diffusion are aligned with the inputs from the multiple views during the denoising process.*

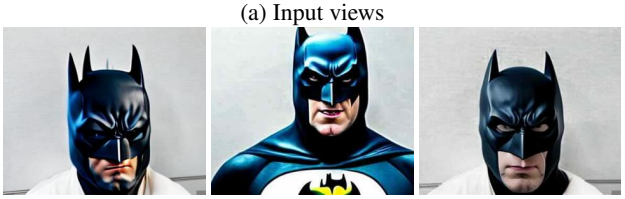
Without loss of generality, let us consider two views, denoted as $\boldsymbol{I}_a$ and $\boldsymbol{I}_b$, and suppose there are K corresponding points denoted as $\{(\boldsymbol{p}_a^k, \boldsymbol{p}_b^k)\}_{k=1}^K$. According to the definition of denoising in equation 3, now the conditioning for editing the two images becomes $\boldsymbol{c}_a = \{\boldsymbol{c}_{\text{prompt}}, \boldsymbol{I}_a\}$ and $\boldsymbol{c}_b = \{\boldsymbol{c}_{\text{prompt}}, \boldsymbol{I}_b\}$. For time t, we further have the estimated sample $\boldsymbol{x}_a^t$ and $\boldsymbol{x}_b^t$ based on the conditioning. Our regularization is based on a distance on the estimated noise based on the corresponding points in the input, namely correspondence distance, as defined by

$$d\left(\boldsymbol{x}_a, \boldsymbol{x}_a; \{(\boldsymbol{p}_a^k, \boldsymbol{p}_b^k)\}_{k=1}^K\right) = \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{x}_a[\boldsymbol{p}_a^k] - \boldsymbol{x}_b[\boldsymbol{p}_b^k]\right), \tag{5}$$

where $\boldsymbol{x}^t[\boldsymbol{p}^k]$ means the value of $\boldsymbol{x}^t$ on the coordinate $\boldsymbol{p}^k$. Our regularization can be then formulated as a loss for min-



(a) Input views



(b) View-by-view editing w/o regularization

(c) With regularization on all denoising steps (*i.e.* using Eq. (7))

(d) With our softened regularization (Sec. 4.2)

Figure 5. **Multiview image editing results** of (a) input views: (b) view-by-view editing baseline without any regularization; (c) enforcing multiview correspondence on all denoising steps; (c) our softened correspondence regularization. (Prompt: *"Turn him into Batman"*.)

imizing the distance, *i.e.*,

$$L_{\text{reg}}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \min_{\boldsymbol{x}_a, \boldsymbol{x}_b} d\left(\boldsymbol{x}_a, \boldsymbol{x}_a; \{(\boldsymbol{p}_a^k, \boldsymbol{p}_b^k)\}_{k=1}^K\right). \quad (6)$$

Incorporating the regularization loss in the retraining of the diffusion network can indeed be a viable approach. However, it is worth noting that this method may result in significant training time requirements, making it less desirable. Fortunately, there exists a closed-form solution for the regularization loss, that is,

$$\overline{\boldsymbol{x}_a^t}[\boldsymbol{p}_a^k] = \overline{\boldsymbol{x}_b^t}[\boldsymbol{p}_b^k] = \frac{1}{2}\left(\boldsymbol{x}_a^t[\boldsymbol{p}_a^k] + \boldsymbol{x}_b^t[\boldsymbol{p}_b^k]\right), \forall t \in [0, T] \quad (7)$$

where $\overline{\boldsymbol{x}_a^t}$ and $\overline{\boldsymbol{x}_b^t}$ are the results with the perfect multiview correspondence. Eq. (7) can be interpreted that we calculate the mean value of all correspondence pairs on the outputs from Eq. (3) and continue diffusion with Eq. (2). However, directly using Eq. (7) leads to undesired over-blurred image outputs, as shown in Fig. 5(c).

The occurrence of blurred outcomes can be comprehended. This is predominantly due to the presence of noise in the pairs of multiview correspondences (*i.e.*, $\{(\boldsymbol{p}_a^k, \boldsymbol{p}_b^k)\}_{k=1}^K$). This noise can arise from either imprecise estimation of correspondences or due to the diffusion backbone we have employed, which relies on downsampling in the latent space. Even if the pixel-level correspondence appears to be reliable, it does not guarantee accurate correspondence in the latent space. Furthermore, there may exist instances where strict correspondence is lacking in the latent space. In cases where the correspondence is noisy, the application of regularization will result in a blurred effect, as it produces an averaged outcome across different patches. We propose to soften the regularization in the following section to address the undesired blurry outputs.

4.2. Softened Regularization

The cause of the blurred generation can be attributed to an excessively strong regularization. Thus, it is natural to consider decreasing the strength of regularization in the diffusion process. Our insight is that this reduction in regularization should preserve the majority of the semantics without the necessity of retaining all image details. For example, if we want to edit human facial images while maintaining the multiview alignment of the facial structure is essential, inconsistencies in details such as wrinkles across different views may not be of significant concern.

Drawing on previous research [45], which suggests that diffusion models primarily focus on capturing the overall structure during the initial denoising stages and pay more attention to finer details in subsequent stages, we propose to apply Eq. (7) solely to the early steps of the diffusion process. Formally, we introduce a threshold, denoted as T_{end}, which replaces T in Eq. (7). For denoising steps beyond T_{end}, no regularization is employed, and the denoising steps are carried out as per the standard procedure outlined in Eq. (3).

In Figure 1, we present the visual representation to demonstrate the influence of various selections of T_{end}. The level of indistinctness grows with the increase in T_{end}. Empirically, we found that by setting T_{end} to 10, we achieve favorable multiview-consistent editings. Consequently, we have employed this particular value for all of our experiments using the InstructPix2Pix [6].

4.3. Updating Radiance Fields

Style Loss. By utilizing the multiview images obtained by regularized diffusion in the previous section, the process of updating the radiance fields becomes straightforward. First, we follow Instruct-NeRF2NeRF and sample a batch of N training cameras $\{\mathbf{P}_n\}_{n=1}^N$. A set of images is first rendered as in Eq. (4), then we apply multiview correspondence regularized 2D image editing on it. Next, with the same training cameras $\{\mathbf{P}_n\}_{n=1}^N$, we randomly render image patches with these cameras, and sample corresponding patches from the

Figure 6. **Impact of the regularization ending step T_{end}.** Moving the regularization to early steps can avoid unwanted blurriness in the generation. (Prompt: *"Turn him into Albert Einstein"*, *"Turn the tower into Big Ben"* and *"As a wooden building"*.)

generated images in the previous step. Here, we slightly abuse the notation between images and patches, and denote the rendered patches as $\boldsymbol{I}_n$ and the corresponding edited patches as $\boldsymbol{I}_n^{\text{gen}}$. However, direct training with $\boldsymbol{I}_n^{\text{gen}}$ leads to artifacts, which arise from the presence of inconsistent areas since the regularization improves consistency but is not perfect. Therefore, we propose to leverage a loss function that is less sensitive to multiview inconsistency, and through empirical analysis, we find that the style loss based on the Gram matrix [19] is an excellent candidate for this purpose. The total training loss is then

$$L = \frac{1}{N}\sum_{n=1}^{N}\left((\boldsymbol{I}_n - \boldsymbol{I}_n^{\text{gen}})^2 + \lambda(G(\boldsymbol{I}_n) - G(\boldsymbol{I}_n^{\text{gen}}))^2\right), \tag{8}$$

where $G(\cdot)$ is the Gram matrix [29] of the input, and λ is set to 0.1 in our experiments empirically.

Single-Pass Dataset Update. Instruct-NeRF2NeRF proposed a scheme for updating NeRF named iterative dataset update, whereby a randomly selected image from the dataset is edited and subsequently added to the training views. The updated training views are then used to update the NeRF model over multiple steps. In their experimental settings, the NeRF updating step is 10 steps, so the editing model (Instruct-Pix2Pix) is called every 10 steps with a randomly sampled image for updating the training dataset.

We adopt a different strategy for updating NeRF. Thanks to the multiview-consistently edited images, we can directly edit NeRF with those images without the need for iteratively updating the dataset. We denote the set for storing the generated image as $\mathcal{D}_{\text{gen}}$, which is empty at the beginning of the training. Denote the input views as $\mathcal{D}_{\text{input}} = \{\boldsymbol{I}_v\}$, we randomly sample a batch of images $\{\boldsymbol{I}_b\}_{b=1}^{B}$ from $\{\boldsymbol{I}_v\}$ and then apply the multiview correspondence regularized diffusion on the batch of images. Denote the generated images as $\{\boldsymbol{I}_b^{\text{gen}}\}_{b=1}^{B}$, we add them to the set of generated images, *i.e.*, $\mathcal{D}_{\text{gen}} := \mathcal{D}_{\text{gen}} \bigcup \{\boldsymbol{I}_b^{\text{gen}}\}_{b=1}^{B}$. After that, we edit NeRF with $\mathcal{D}_{\text{gen}}$ and training with Eq. (8) for 200 steps. A comparison of Single-Pass Dataset Update and Iterative Dataset Update can be found below.

Iterative Update	**Single-Pass Update**
Init NeRF $\mathcal{F}$ with input $\mathcal{D}_{\text{input}} = \{\boldsymbol{I}_v\}$	Init NeRF $\mathcal{F}$ with input $\mathcal{D}_{\text{input}} = \{\boldsymbol{I}_v\}$
Init $\mathcal{D}_{\text{gen}} = \mathcal{D}_{\text{input}}$	Init $\mathcal{D}_{\text{gen}} = \{\}$
repeat	**repeat**
Sample $\boldsymbol{I}_i$	Sample $\{\boldsymbol{I}_b\}_{b=1}^{B}$
$\boldsymbol{I}_i^{\text{gen}} = \text{Edit}(\boldsymbol{I}_i)$	$\{\boldsymbol{I}_b^{\text{gen}}\} = \text{RegEdit}(\{\boldsymbol{I}_b\})$
$\mathcal{D}_{\text{gen}}[i] = \boldsymbol{I}_i^{\text{gen}}$	$\mathcal{D}_{\text{gen}} := \mathcal{D}_{\text{gen}} \bigcup \{\boldsymbol{I}_b^{\text{gen}}\}$
for i in range(N_{IU}) **do**	**for** i in range(N_{SU}) **do**
Update $\mathcal{F}$ with $\mathcal{D}_{\text{gen}}$	Update $\mathcal{F}$ with $\mathcal{D}_{\text{gen}}$
end for	**end for**
until reach max steps	**until** reach max steps
Return $\mathcal{F}$	**Return** $\mathcal{F}$

N_{IU} and N_{SU} are tuneable hyper-parameters, with default values set at 10 (in [22]) and 200, respectively.

5. Experiments

Our experiments were conducted following the framework proposed in Instruct-NeRF2NeRF [22]. Specifically, we trained the input NeRF model using the Nerfacto model within the Nerfstudio [72]. To evaluate the performance of our method, we conducted experiments on the dataset provided by Instruct-NeRF2NeRF, with the exception that scenes containing identifiable human faces were replaced with our re-captured versions for privacy and legal considerations.

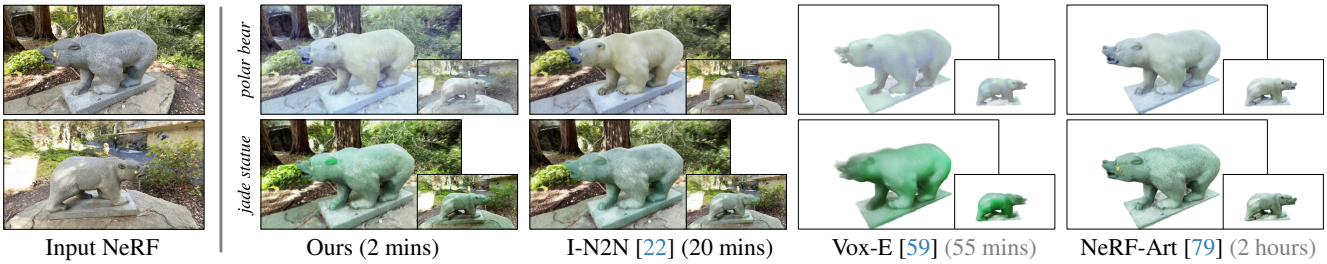


Figure 7. **Comparison with SoTA methods.** For ours and Instruct-NeRF2NeRF, we use instructional text furnished with the suffix *"Turn the bear into a ..."*. Background of input images is removed for Vox-E [59] and NeRF-Art [79].

Figure 8. **Comparison of the NeRF editing efficiency.** Our method can effectively edit NeRFs in just 30 seconds for scene styles.

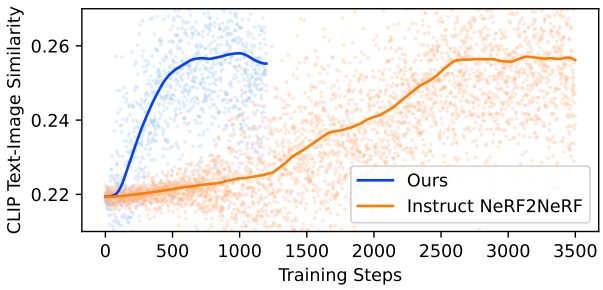


Figure 9. **CLIP Text-Image Similarity score during training.** We repeat the experiments 10 times as the variance of the CLIP Text-Image Similarity score is large. The exponential moving average is adopted to smooth the mean value curve.

Our approach significantly enhances the existing method, Instruct-NeRF2NeRF [22], concerning its efficacy in editing. However, illustrating the efficiency effects of editing at various optimization steps is not straightforward through images. Therefore, we earnestly recommend that readers refer to our supplementary video for a more comprehensive comparison of the editing efficiency. Time in this paper is all measured on a single A100. For our method, editing a batch of 4 multiview images with our regularization takes around 30 seconds, and 1 step of NeRF optimization takes around 0.15 seconds.

5.1. Comparison with State-of-The-Arts

We primarily compare our methodology against the state-of-the-art method Instruct-NeRF2NeRF [22]. We begin by showing the CLIP Text-Image Similarity scores of the edited NeRF along with training iterations, as depicted in Fig. 9. The CLIP Text-Image Similarity score represents the cosine similarity between the textual embedding and the image embedding derived from CLIP. We use the text about the target object within the given instructions. For instance, if the instruction specifies "Turn the bear into a grizzly bear," we utilize "a grizzly bear" as the text input for computing the similarity scores. Notably, we observe

Figure 10. **Our method can be combined with Instruct-NeRF2NeRF.** Initializing with our method can speed up the editing with on-par performance. (Prompt: *"Turn the bear into a panda."*)

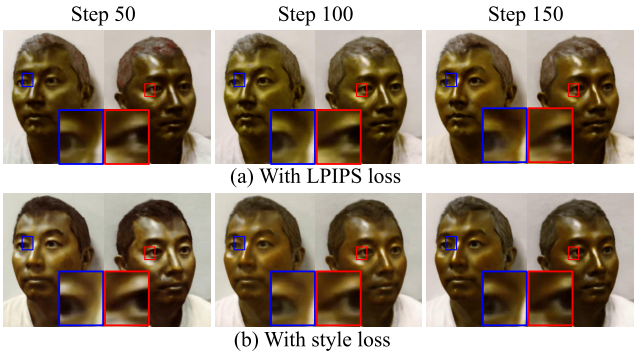


Figure 11. **Comparison of training with LPIPS loss and style loss.** LPIPS loss is adopted in Instruct-NeRF2NeRF. (Prompt: *"As a bronze bust."*)

that the scores exhibit a high variance, with significant fluctuations occurring after each iteration. Consequently, we repeat the experiment 10 times and present the smoothed average value for analysis. We can observe that our proposed approach exhibits significantly accelerated convergence in comparison to the Instruct-NeRF2NeRF method, all the while achieving comparable performance outcomes following optimization completion.

Furthermore, in Fig. 7, we present a visual comparison between our method and Instruct-NeRF2NeRF, Vox-E [59] and NeRF-Art [79]. While NeRF-Art [79] and Vox-E [59] have shown commendable performance, we have opted not to report their time metrics. This is due to the 3D representation adopted in their methodology, which makes it hard to offer a justifiable comparison. We showcase the efficacy of our approach in Fig. 8. Remarkably, we observe that our method excels in editing scene style, potentially owing to the ability of our regularization to attain superior multiview coherence within this particular context.

5.2. Ablation Studies

Compatibility to Instruct-NeRF2NeRF. We have demonstrated the effectiveness of our approach in the preceding experimental section. However, it should be noted that the final editing outcomes may not always match the level of quality achieved by Instruct-NeRF2NeRF. Although the similarity scores depicted in Fig. 9 indicate a close performance between the two methods, in practice, we have observed that our approach and Instruct-NeRF2NeRF exhibit different generation patterns. In certain instances, it becomes challenging to unequivocally assert that our method is consistently more visually appealing than Instruct-NeRF2NeRF.

To address this issue, we propose a solution that involves combining our method with Instruct-NeRF2NeRF. By initiating the optimization process with our approach and subsequently switching to Instruct-NeRF2NeRF, we can achieve improved results. The outcomes of this combined method are presented in Fig. 10. It is evident from the figure that following initialization with our method, Instruct-NeRF2NeRF requires only 800 additional steps to produce outcomes of comparable quality to those achieved with 3000 steps without initialization.

Style Loss. In Sec. 4.3, we claimed that incorporating style loss can effectively mitigate the influence of inconsistency in the input edited views. In Fig. 11, we present a comparative analysis between the utilization of style loss and the default LPIPS loss in Instruct-NeRF2NeRF. Remarkably, we observe that the images exhibit enhanced sharpness and a more consistent alteration in style throughout the training procedure. As an illustration, we note the gradual transition to a slightly green hue on the faces during the middle stages of training with LPIPS loss.

6. Conclusion

We introduce an efficient framework for editing NeRF. Our framework entails the editing of multiple views of images using the proposed multiview correspondence regularization. Subsequently, we perform optimization of NeRF using these manipulated images. Our approach demonstrates considerable efficiency when compared to recent SoTA methods such as Instruct-NeRF2NeRF. Unlike these methods,

which rely on single-view editing and iterative optimization of NeRF, our approach achieves multiview consistency in the edited images without requiring iterative processes.

Acknowledgement

We thank Ayaan Haque for reviewing and verifying the claims presented in this paper.

References

[1] Edward H Adelson, James R Bergen, et al. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1(2):3–20, 1991. 3

[2] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. 3

[3] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. 1, 2

[4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 3

[5] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems*, 2021. 3

[6] 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*, pages 18392–18402, 2023. 1, 3, 5

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, 2022. 3

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 3

[10] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting, 2023. 3

[11] Yiwen Chen, Chi Zhang, Xiaofeng Yang, Zhongang Cai, Gang Yu, Lei Yang, and Guosheng Lin. It3d: Improved text-to-3d generation with explicit view synthesis. *arXiv preprint arXiv:2308.11473*, 2023. 3

[12] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4182–4194, 2023. 3

[13] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023. 3

[14] 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. 3

[15] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[16] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. In *European Conference on Computer Vision*, pages 636–654. Springer, 2022. 3

[17] Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. *arXiv preprint arXiv:2311.16037*, 2023. 3

[18] Shuangkang Fang, Yufeng Wang, Yi Yang, Yi-Hsuan Tsai, Wenrui Ding, Ming-Hsuan Yang, and Shuchang Zhou. Text-

driven editing of 3d scenes without retraining. *arXiv preprint arXiv:2309.04917*, 2023. 3

[19] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2, 3, 6

[20] Bingchen Gong, Yuehao Wang, Xiaoguang Han, and Qi Dou. Recolornerf: Layer decomposed radiance field for efficient color editing of 3d scenes. *arXiv preprint arXiv:2301.07958*, 2023. 3

[21] Ori Gordon, Omri Avrahami, and Dani Lischinski. Blended-nerf: Zero-shot object generation and blending in existing neural radiance fields. *arXiv preprint arXiv:2306.12760*, 2023. 3

[22] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 2, 3, 6, 7

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[24] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13869–13878, 2021. 3

[25] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 3

[26] Yukun Huang, Jianan Wang, Yukai Shi, Xianbiao Qi, Zheng-Jun Zha, and Lei Zhang. Dreamtime: An improved optimization strategy for text-to-3d content creation. *arXiv preprint arXiv:2306.12422*, 2023. 3

[27] 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. 3

[28] 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

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 6

[30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 3, 4

[31] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 3

[32] 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. 3

[33] Jae-Hyeok Lee and Dae-Shik Kim. Ice-nerf: Interactive color editing of nerfs via decomposition-aware weight optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3491–3501, 2023. 3

[34] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 3

[35] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. *arXiv preprint arXiv:2308.10608*, 2023. 3

[36] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3227–3238, 2023. 3

[37] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3

[38] Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3d asset using case-aware diffusion priors, 2023. 3

[39] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 3

[40] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 3

[41] 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. 3

[42] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 3

[43] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8446–8455, 2023. 3

[44] Mohit Mendiratta, Xingang Pan, Mohamed Elgharib, Kartik Teotia, Mallikarjun B R, Ayush Tewari, Vladislav Golyanik, Adam Kortylewski, and Christian Theobalt. Avatarstudio: Text-driven editing of 3d dynamic human head avatars. *ACM Trans. Graph.*, 42(6), 2023. 3

[45] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 5

[46] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 3, 4

[47] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Watch your steps: Local image and scene editing by text instructions. *arXiv preprint arXiv:2308.08947*, 2023. 3

[48] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 3

[49] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 3

[50] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2022. 1, 2, 3

[51] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 3

[52] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C Lin. Dynamic mesh-aware radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 385–396, 2023. 3

[53] 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

[54] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023. 3

[55] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[56] 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. 1, 3

[57] 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. 2

[58] Sara Fridovich-Keil and 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*, 2022. 3

[59] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 430–440, 2023. 1, 3, 7, 8

[60] Hoigi Seo, Hayeon Kim, Gwanghyun Kim, and Se Young Chun. Ditto-nerf: Diffusion-based iterative text to omni-directional 3d model. *arXiv preprint arXiv:2304.02827*, 2023. 3

[61] 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. 3

[62] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Anything-3d: Towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*, 2023. 3

[63] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation, 2023. 3

[64] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 3

[65] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[66] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 3

[67] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3

[68] 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*, pages 7495–7504, 2021. 3

[69] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3

[70] Yuqi Sun, Reian He, Weimin Tan, and Bo Yan. Instruct-neuraltalker: Editing audio-driven talking radiance fields

with instructions. *arXiv preprint arXiv:2306.10813*, 2023. 3

[71] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion: (0-)image-conditioned 3d generative models from 2d data. *arXiv*, 2023. 3

[72] 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. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 6

[73] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3

[74] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *ICCV*, 2023. 3

[75] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 3

[76] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv preprint arXiv:2304.12439*, 2023. 3

[77] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, pages 443–453. IEEE, 2022. 3

[78] 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

[79] 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. 1, 3, 7, 8

[80] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 2

[81] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 3

[82] Daniel Watson, William Chan, Ricardo Martin Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *International Conference on Learning Representations*, 2022. 3

[83] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 3

[84] Jinbo Wu, Xiaobo Gao, Xing Liu, Zhengyang Shen, Chen Zhao, Haocheng Feng, Jingtuo Liu, and Errui Ding. Hd-fusion: Detailed text-to-3d generation leveraging multiple noise estimation. *arXiv preprint arXiv:2307.16183*, 2023. 3

[85] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 3

[86] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4479–4489, 2023. 2, 3

[87] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 3

[88] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 3

[89] Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models, 2023. 3

[90] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 3

[91] Chaohui Yu, Qiang Zhou, Jingliang Li, Zhe Zhang, Zhibin Wang, and Fan Wang. Points-to-3d: Bridging the gap between sparse points and shape-controllable text-to-3d generation. *arXiv preprint arXiv:2307.13908*, 2023. 3

[92] 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. 3

[93] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, Leif Kobbelt, and Lin Gao. Interactive nerf geometry editing with shape priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[94] 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 Trans. Graph.*, 40(4):149:1–149:18, 2021. 3

[95] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 3

[96] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):1–12, 2018. 3

[97] Joseph Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023. 3

[98] 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

Supplementary
Efficient-NeRF2NeRF: Streamlining Text-Driven 3D Editing with Multiview Correspondence-Enhanced Diffusion Models

Anonymous CVPR submission

Paper ID 1026

1. Implementation details

Our method is implemented on Nerfstudio [**?**]. To get the correspondence inputs across different views, we start from the first image in the batch and directly calculate the correspondence points on other frames by projects. This is because we have the NeRF input and the depth maps are available for multiview projection. However, we found that using the optical flow-based dense correspondence estimator, such as PIPS [**?**] achieves a similar performance. For 360-degree scenes, we empirically found that PIPS is slightly more robust since there are heavy occlusions in the scene, and removing occluded areas can leave large areas of a view not regularized. An example is the front view and the back view of the bear scene. The bear image can be not regularized in this case if we use depth-based correspondence, but it can be regularized if we use optical-flow-based correspondence due to the similar patches in the images.

A potential problem of the multiview correspondence regularization is that the regularization is designed to be applied to batches. However, this can only improve the multiview consistency for this specific batch, and there are no improvements across different batches. A straightforward way to solve the problem is that since every module is doing inference during diffusion, we can just cache the denoising outputs on disk and then apply the regularization on all views. Apparently, caching intermediate denoising values takes a large amount of time, which is unwanted in our Efficient-NeRF2NeRF framework. Our solution to this problem is to use the same random noise across edits in different batches.

2. More quantitative results

Here, we provide additional quantitative results following the evaluation in Instruct-NeRF2NeRF. The rationale behind including these results in the supplementary section lies in the observation that the metrics exhibit a considerable amount of variance. Specifically, the CLIP-based similarity scores exhibit challenges in terms of reproducibility, despite employing fixed random seeds and conducting repeated runs. As shown in Fig. 1, slightly rotating the camera results in a big change in the similarity score. Furthermore, it is worth noting that the alignment between the similarity scores and human preference often proves to be inadequate. Nevertheless, we report more detailed quantitative results in Tab. 1. In the table, the method "Per-frame IP2P" and "One-time DU" are following the baseline setup in Instruct-NeRF2NeRF. "Per-frame IP2P" means directly

Similarity score: 0.2383 Similarity score: 0.2467

Figure 1. The metric, CLIP text-image similarity, exhibits a considerable amount of variance. For this example, the text for calculating similarity is "grizzly bear". We slightly rotate the camera a little bit, but the score changes a lot.

Method	CLIP Text-Image Similarity	CLIP Direction Consistency
Per-frame IP2P	0.2383	0.9124
One-time DU	0.1332	0.9482
Ours w/o reg	0.1820	0.9341
IN2N (2 mins)	0.1865	0.9978
IN2N (20 mins)	0.2156	0.9763
Ours (2 mins)	0.2127	0.9625

Table 1. Quantitative evaluation results on the tested scenes.

apply frame-by-frame editing on the images. "One-time DU" means directly editing all images at the beginning of optimization. "Ours w/o reg" means using our method but without the regularization, and results with 2 minutes' optimization are reported. The variance of CLIP Text-Image Similarity is around 0.06 for all methods. The variance of CLIP Direction Consistency is around 0.03 for all methods except IN2N with 2 minutes, since IN2N after 2 minutes generates images roughly similar to the original NeRF. For testing the performance, we use the scenes released by Instruct-NeRF2NeRF and remove the "face" and "person-small" scenes due to privacy and legal concerns. For "farm-small" and "campsite-small", we use six prompts as in Instruct-NeRF2NeRF: "Make it autumn", "Make it midnight", "Make it look like it just snowed", "Make it stormy", "Make it sunset" and "Make it look like the Namibian desert". For the "bear" scene, we use 3 prompts: "Turn the bear into a grizzly bear", "Turn the bear into a panda" and "Turn the bear into a polar bear".