

Language-driven Object Fusion into Neural Radiance Fields with Pose-Conditioned Dataset Updates

Ka Chun Shum¹ Jaeyeon Kim¹ Binh-Son Hua^{2,4} Duc Thanh Nguyen³ Sai-Kit Yeung¹

¹Hong Kong University of Science and Technology ²VinAI ³Deakin University ⁴Trinity College Dublin

Abstract

Neural radiance field (NeRF) is an emerging technique for 3D scene reconstruction and modeling. However, current NeRF-based methods are limited in the capabilities of adding or removing objects. This paper fills the aforementioned gap by proposing a new language-driven method for object manipulation in NeRFs through dataset updates. Specifically, to insert an object represented by a set of multi-view images into a background NeRF, we use a text-to-image diffusion model to blend the object into the given background across views. The generated images are then used to update the NeRF so that we can render view-consistent images of the object within the background. To ensure view consistency, we propose a dataset update strategy that prioritizes the radiance field training based on camera poses in a pose-ordered manner. We validate our method in two case studies: object insertion and object removal. Experimental results show that our method can generate photo-realistic results and achieves state-of-the-art performance in NeRF editing.

1. Introduction

Editing of 3D scenes by insertion or removal of objects has been a fundamental task in computer graphics and computer vision. The task has often been performed using traditional 3D scene authoring tools [5, 9]. For example, to insert an object into a 3D scene, traditional approach requires user to manually select the object and position it into the scene. This manual pipeline has been adopted in a wide range of applications, such as furniture arrangement in interior design [15] and asset creation in game development [10, 13].

Recent advances in deep learning have opened new directions to scene modeling. Neural radiance field (NeRF) [40] is a pioneer to render view-consistent photo-realistic images using neural networks. Generative models such as generative adversarial networks (GANs) [12] and diffusion models [17] learn to output photo-realistic images from un-

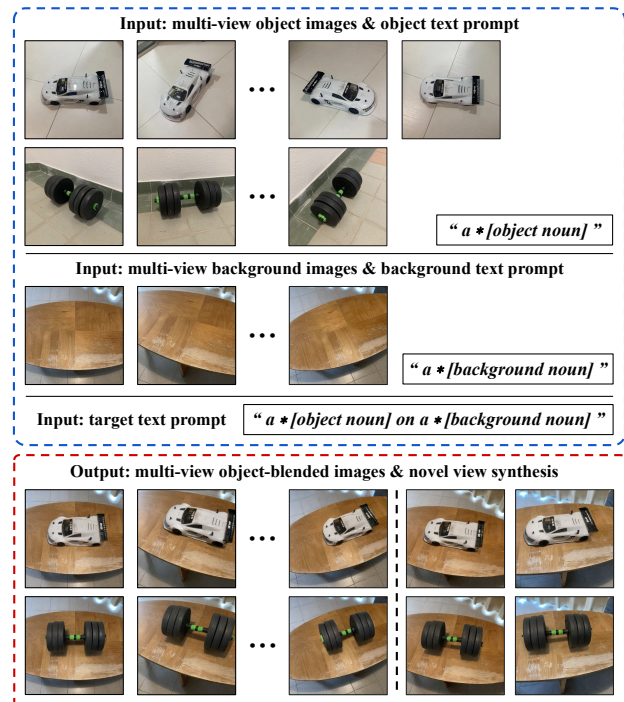


Figure 1. **Object insertion.** We propose a language-driven method for view-consistent 3D object insertion into a background NeRF scene. Given an object defined by a set of multi-view images, our method generates plausible text-guided insertion results that require geometry manipulation of the original background NeRF.

constrained image collections. Recent text-guided diffusion models [50, 51, 53] have shown great promise in generating high-quality and diverse images from a single text prompt.

Such successes inspire us to revisit 3D scene editing via language-driven image synthesis techniques. Particularly, for object insertion (see Fig. 1), our method performs view-consistent edits by taking as input a set of multi-view object images, multi-view background images, and a target text prompt. We utilize a text-to-image diffusion model to con-

tinuously synthesize multi-view images containing both a target object and a background during NeRF training. These synthesized images can be iteratively used to refine the NeRF of the background to learn geometric and appearance features of the object. This approach to refining a radiance field is also known as dataset update [14]. However, a particular challenge is that the refinement process may result in unstable NeRF training, degrading its rendering quality due to inconsistent views synthesized. To address this issue, we propose a *pose-conditioned* dataset update strategy that gradually engages the target object into the background, beginning at a randomly selected pose (view), then views close to the already-used views before propagating to views further away. We observe that this pose-conditioned strategy significantly improves the NeRF learning, reducing rendering artifacts and maintaining view-consistent rendering. In summary, we make the following contributions in our work.

- We propose a new framework for object manipulation in NeRFs via text-to-image diffusion. Our work can create high-quality 3D scenes from simple inputs (text prompts and multi-view images). It, therefore, has the potential for conveniently building a variant of 3D object libraries that is applicable for the scene editing task.
- We propose a pose-conditioned dataset update strategy that stabilizes the fusion of objects into a background NeRF, enabling view-consistent rendering. Our method requires no priors on the geometry or texture of the objects, unlike existing works relying on accurate meshes [7, 20, 31, 81], depth maps [42, 67], fine masks [58, 63], semantics [1], or lighting assumption [69, 78].
- We showcase our method in two case studies: object insertion and object removal via a user-friendly manner. Specifically, we use a 3D bounding box to define the location for object insertion/removal. Box-based location requires only a rough orientation, which is easy to be adjusted and visualized using real-time NeRF GUIs [43, 59].
- We conduct extensive experiments to validate key techniques of our method and demonstrate its state-of-the-art performance in NeRF editing.

2. Related work

Image synthesis. Early attempts in deep learning-based image synthesis have utilized generative models such as GANs [12]. Several methods combine supervised losses with adversarial losses to learn class-conditional GANs [41] and image-to-image translations [21, 65, 80]. StyleGAN [22] and its variants [23, 24] perform image editing directly on latent representations, but such editings are not intuitive and thus difficult to control the output.

Recently, diffusion models [17] have made significant progress in image synthesis with high-quality data samples constructed from sophisticated forward and denoising diffusion steps. In addition, vision-language models such as

CLIP [49] have made text prompts an intuitive condition to generate and edit images. These developments have led to text-guided diffusion models [50, 51, 53] which offer excellent synthesis quality. Downstream applications then can be built by fine-tuning these text-guided models to fit with the application domains [4, 25, 36, 52, 70, 77]. Our method follows this streamline where we use a text-to-image diffusion model [52] to guide the fusion of objects into a background.

Neural 3D modeling. Image-based 3D modeling methods use convolutional neural networks (CNNs) [29] to learn 3D structures from multiple images [11, 18, 33, 60, 79]. However, CNNs often struggle to deal with complex shapes, texture, and lighting captured in the images. Follow-up works integrate differentiable rendering and represent 3D structures as neural surfaces [39] or shapes [8]. 3D-aware GANs integrates volume rendering into their generators to synthesize novel views from a single image [6, 45, 56, 57]. NeRFs [40] apply the same neural rendering approach, but are optimized on a ray rendering loss on multi-view input images. There are methods addressing the limitations of NeRFs in various aspects such as visual artifacts [2], data complexity [37], camera poses [74], and computational efficiency [43, 76].

NeRF editing. NeRF editing has often been carried out by parameter tuning [35], layer feature fusion [61], or deformable rays [75]. An alternative to editing a NeRF is to directly amend the multi-view images used to learn it. For example, NeRF stylization methods [19, 44, 46, 62] freeze the geometry branch and optimize the color branch in a NeRF to stylize multi-view images. Several methods attempt to decompose an existing NeRF, which in turn holds color information and separates voxels considering multi-view masks [30] or semantics [27, 28]. NeRF inpainting fills simple unseen background geometry and colors with help of depth priors [42] or filtering inpainted multi-views [67].

Creation of complex geometry and vivid colors for a NeRF is challenging due to higher-level requirements of consistency. DiscoScene [72] fuses background and object NeRFs, thus is unable to condition customized contents. FocalDreamer [31] and DreamEditor [81] rely on fine meshes to function, disregarding the advantage of multi-view representation of NeRFs. Our method is perhaps most similar to Instruct-NeRF2NeRF [14] which shares a related data updating schema. However, Instruct-NeRF2NeRF [14] always generates geometry aligned with the original geometry (e.g., transfer a sneaker to a sneaker-shape apple), thus fails in most cases of object insertion or removals. Besides, it requires heavy retraining of a diffusion model on large-scale self-constructed datasets.

Recently, text-to-image diffusion has been applied to generate 3D contents. DreamFusion [47] introduced a score distillation sampling (SDS) loss that progressively consolidates

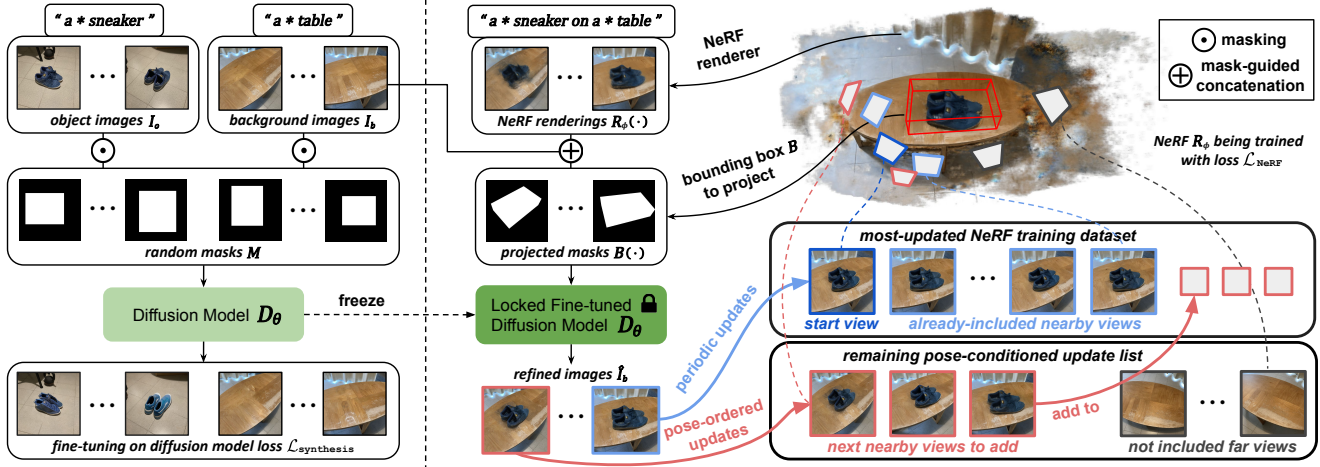


Figure 2. **Overview of our pipeline.** We customize and fine-tune a text-to-image diffusion model for view synthesis in an inpainting manner (left). We then apply the model to progressively fuse an object into background views to update a background NeRF (right). The process of view synthesis and NeRF updating is performed repeatedly. Views generated by the diffusion model are added to an on-going dataset to strengthen the NeRF. In return, the NeRF renders color hints for the diffusion model to create new views.

view information from a diffusion model into NeRFs. This loss has been applied with additional treatments to image resolution [32], conditional images [38], photo-realism [66], or scene geometry [71]. We also adopt the SDS loss but develop a novel training schedule to address challenges in multi-view object and background fusion for NeRF editing.

3. Method

3.1. Overview

For the ease of presentation, we first describe our method for object insertion. We then extend it to object removal. An overview of our method for object insertion is illustrated in Fig. 2. Here we aim to insert an object into a background NeRF in two steps. In the first step, we synthesize training views for the NeRF with the target object embedded in the background. In the second step, the NeRF is updated with the synthesized views to learn the geometry and appearance of the object. There are two key challenges in this approach. First, background preserving is required in the synthesized views for the NeRF updating. Second, image synthesis may generate view-inconsistent images, causing artifacts in the resulting NeRF.

To address the first challenge (i.e., object-blended image synthesis with background preserving), we leverage a state-of-the-art diffusion model to image synthesis, and opt to customize the model with text prompts for object blending (see Sec. 3.2). We formulate this task as image inpainting where we place a binary mask on each background image to indicate the location where the object is inserted in, and adjust the text prompts with both the object and background.

To achieve view-consistent renderings, we propose a new

strategy, namely *pose-conditioned* dataset updates, to schedule the data used in the NeRF updating (see Sec. 3.3). Our strategy is inspired by an important observation about the nature of NeRF: a view rendered by a NeRF maintains an extent of pose-aware color information from nearby already-used views, the nearer the more noticeable. Therefore, if we pass nearby renderings to the diffusion model with properly controlled noise, view-consistent results can be generated based on the learned color hints. From such an observation, we design a novel data scheduler for our NeRF updating. We initially train the NeRF using regular method on a dataset of multi-view background images. We then progressively fuse the object into the NeRF by iteratively updating the dataset with object-blended background images in a pose-ordered manner, i.e., views are sorted in such a way that new views are acquired nearby already-used views. Our method can also be adapted to implement object removal (see Sec. 3.4).

3.2. Object-blended image synthesis

We present our target object and background by a set of multi-view object images \mathbf{I}_o and background images \mathbf{I}_b . Our goal is to build an image synthesis model that can blend the object (from \mathbf{I}_o) into the background (from \mathbf{I}_b) at custom locations specified by binary masks M .

Let D_θ (with parameters θ) be such an image synthesis model. Here we adopt a pre-trained Stable Diffusion [51] to implement D_θ . The model D_θ includes a denoising model ϵ_θ that learns the noise component $\epsilon_\alpha \sim \mathcal{N}(\mathbf{0}, 1)$ added in the diffusion process, where $\alpha \in [0, 1]$ is a denoising strength. We customize D_θ with our background and object images in an inpainting fashion. Specifically, for each image $I \in \mathbf{I}_b \cup \mathbf{I}_o$, we make a masked-out (background-preserved)

version $I \odot M$ where $M \sim \mathcal{U}$ is a random square binary mask whose coordinates are sampled from a uniform distribution \mathcal{U} and \odot is a pixel-wise product. The image I is associated with a text prompt p where we use “*” to indicate our object/background identifiers [52], e.g., “* sneaker” means that our target object is a sneaker and “* table” is our target background. We fine-tune D_θ with the following loss:

$$\mathcal{L}_{\text{synthesis}}(\theta) = \mathbb{E}_{I, M \sim \mathcal{U}, p, \epsilon_\alpha \sim \mathcal{N}(\mathbf{0}, \mathbf{1})} [\|\epsilon_\alpha - \epsilon_\theta(I)\|_2^2] \quad (1)$$

Eq. 1 represents a general form for the fine-tuning process. Intermediate steps, e.g., input formation, latent encoding-decoding, are presented in our supplementary material.

We fine-tune the diffusion model D_θ with all the images in \mathbf{I}_b and \mathbf{I}_o for n_{bg} and n_{obj} times. After fine-tuning, we can use D_θ to synthesize images \hat{I}_b that blend the object presented in \mathbf{I}_o into the background images $I_b \in \mathbf{I}_b$ via a combined text prompt \tilde{p} , e.g., “a * sneaker on a * table”,

$$D_\theta(I_b \odot M, \tilde{p}, \alpha) \rightarrow \hat{I}_b \quad (2)$$

where α is set manually to control how much object content in the masked area to be inpainted by D_θ (we demonstrate how we set α for different purposes in Sec. 3.3).

3.3. Pose-conditioned dataset updates

Let R_ϕ (with parameters ϕ) be our NeRF, which is initially trained with the multi-view background images in \mathbf{I}_b using an existing approach [43]. We also retrieve a set of camera poses $\{\pi(I_b)\}$ (4×4 matrices) for the images $I_b \in \mathbf{I}_b$ using the pose estimation method COLMAP [54, 55].

We propose to update the NeRF R_ϕ progressively with object-blended background images generated by the fine-tuned diffusion model D_θ . Our idea is to build a progressive multi-view and pose-conditioned object-blended image dataset $\hat{\mathbf{I}}_b$ from \mathbf{I}_b to be used to update R_ϕ . We initialize $\hat{\mathbf{I}}_b^{(0)} = \emptyset$ and $\mathbf{I}_b^{(0)} = \mathbf{I}_b$. We start with a random background image $I_b \in \mathbf{I}_b$. Let B be a 3D bounding box, and $B(I_b)$ be the projection mask of B onto I_b using the camera pose $\pi(I_b)$. We generate an object-blended image \hat{I}_b as,

$$D_\theta(I_b \odot B(I_b), \tilde{p}, \alpha) \rightarrow \hat{I}_b \quad (3)$$

where we set $\alpha = 1$ (maximum strength) to blend the object into the masked area by using only the knowledge that D_θ learns from \tilde{p} (during the fine-tuning), and colors from the background image I_b as there is no existing color clue in the masked area $I_b \odot B(I_b)$.

We update $\hat{\mathbf{I}}_b^{(1)} = \hat{\mathbf{I}}_b^{(0)} \cup \{\hat{I}_b\}$ and $\mathbf{I}_b^{(1)} = \mathbf{I}_b^{(0)} \setminus \{I_b\}$. We then update R_ϕ using $\hat{\mathbf{I}}_b^{(1)}$, and keep doing so by adding images into $\hat{\mathbf{I}}_b$ sequentially. In particular, let $\hat{\mathbf{I}}_b^{(n-1)}$ be the dataset at the $(n-1)$ -th step. The next image $I_b^{(n)} \in \mathbf{I}_b$ for

step n is selected so as it is closest (in terms of the camera poses) to already-used images,

$$I_b^{(n)} = \arg \min_{I_{b,k} \in \mathbf{I}_b^{(n)}} \min_{I_{b,j} \in \mathbf{I}_b^{(n-1)}} \|\pi_T(I_{b,k}) - \pi_T(I_{b,j})\|^2 \quad (4)$$

where π_T is the translation component of the pose π . This view selection reflects the standard multi-view reconstruction pipeline where the multi-view data are collected with a smoothly connected, inward-surrounding camera trajectory. This requirement is also fulfilled by most NeRF datasets.

Given a background image $I_b^{(n)}$ in Eq. 4, to utilize the nearby color hints learned by R_ϕ , we create a background-preserved foreground-rendered image $\tilde{I}_b^{(n)}$ as,

$$\tilde{I}_b^{(n)} = (I_b^{(n)} \odot B(I_b^{(n)})) \oplus (R_\phi(\pi(I_b^{(n)})) \odot \bar{B}(I_b^{(n)})) \quad (5)$$

where $R_\phi(\pi(I_b^{(n)}))$ is the rendering result of R_ϕ in the pose $\pi(I_b^{(n)})$ (for the background image $I_b^{(n)}$), \bar{B} is the complement of B (as we want to maintain the foreground rendered by R_ϕ), and \oplus is a pixel-wise addition.

We then generate a view-consistent image $\hat{I}_b^{(n)}$ by applying D_θ to $\tilde{I}_b^{(n)}$ defined in Eq. 5 as,

$$D_\theta(\tilde{I}_b^{(n)} \odot B(I_b^{(n)}), \tilde{p}, \alpha) \rightarrow \hat{I}_b^{(n)} \quad (6)$$

where we set α to a low value to allow D_θ to utilize color hints from previous nearby views, provided from the rendering results $R_\phi(\pi(I_b^{(n)}))$. We empirically found that $\alpha \in [0.3, 0.4]$ gives best view-consistent rendering.

Again, we update $\hat{\mathbf{I}}_b^{(n)}$ with $\hat{I}_b^{(n)}$, and update R_ϕ accordingly by minimizing the loss:

$$\mathcal{L}_{\text{NeRF}}(\phi) = \mathbb{E}_{\hat{I}_b \in \hat{\mathbf{I}}_b^{(n)}} [\|R_\phi(\pi(\hat{I}_b)) - \hat{I}_b\|^2]. \quad (7)$$

During the updating process, we include n_{near} views into the ongoing dataset for every n_{new} NeRF updating steps. In addition, we periodically replace each old view by a new one using Eqs. 5 and 6 for every n_{old} NeRF updating steps. The updating procedure is completed once all the background images in \mathbf{I}_b have been processed, i.e., $\mathbf{I}_b^{(n)} = \emptyset$.

3.4. Adapting to object removal

Our framework can be adapted to object removal. In particular, we also fine-tune the diffusion model D_θ as in Sec. 3.2. However, we do not fine-tune D_θ with object images as we want to remove objects. Instead, we first individually inpaint all background images on projection masks using background prompts without identifier (e.g., “a table”). These inpainted images are then treated as *pseudo ground-truth* and used to customize D_θ with the background text prompts containing identifiers (e.g., “a * table”). The pseudo ground-truth backgrounds are visually pleasing but still remain cross-view inconsistencies. To circumvent this issue, we perform

NeRF updating using dataset updates as in Sec. 3.3. However, only background prompts are used during the NeRF updating. We found that this joint 2D-3D interaction gradually transforms disruptive inpainted image regions into view-consistent background images. We show the necessity of the pseudo ground-truth and the NeRF updating in making view-consistent object removal in our experiments.

4. Experiments

4.1. Datasets

For object insertion, we propose a dataset comprising multi-view images of 8 backgrounds and 10 objects. The number of images for each background and object ranges from 60 to 100 and 40 to 80, respectively. We collected the images using an iPhone camera and resized the images to 512×512 resolution for the ease of training. Except for our self-captured data, we also test on synthetically rendered multi-view images [40, 47].

For object removal, we run experiments on the commonly used inpainting datasets from Mip-NeRF-360 [3] and IBR-Net [64]. We centrally cropped images in those datasets to make 512×512 images to fit with our pipeline.

4.2. Baselines

For comparisons, we select SOTA baselines from different branches of work that can perform object insertion. To ensure the fairness, we barely modify the baselines, only when necessary to fit them with our settings.

Traditional 3D editing pipeline. This pipeline creates a 3D model from multi-view images. Scene editing is then performed directly on the 3D model. To simulate this traditional pipeline, we use COLMAP [54, 55] to reconstruct the mesh for the background and object, and then manually crop and place the object mesh into the background mesh.

Image inpainting. We adopt the inpainting variant of Stable Diffusion [51] as a baseline. However, since image inpainting treats each view independently, for a fair comparison, we perform single-view inpainting on each background image.

Single-image-to-3D NeRF. NeRF-based view synthesis partially fulfills scene editing. Here we select Zero123 [34], a SOTA that uses a distillation prior [47] from a 2D diffusion model to synthesize novel views for comparison.

Instruct-NeRF2NeRF [14]. This work also applies dataset updates for NeRF training. However, it fails to add/remove objects with noticeable non-uniform appearance. To test this method, we re-format text prompts, e.g., we replace “a * sneaker on a * table” by “add/make a sneaker on a table”.

4.3. Implementation details

We fine-tune the diffusion model D_θ with object images ($n_{obj} = 5,000$ times) more than background images ($n_{bg} = 500$ times) as we found the model needs extra training to

learn objects with complex geometry and texture. For NeRF training, we empirically found $n_{near} = 3$, $n_{new} = 500$, and $n_{old} = 10$ balance well the quality and efficiency.

We run all experiments on a Nvidia RTX 3090 GPU. Diffusion fine-tuning takes around 30 minutes. NeRF updating speed relies on the number of background images while each backpropagation takes 0.5 seconds. Inference speed of the diffusion model is 8 seconds/image. Our NeRF training costs around 2.7 times more than the standard NeRF training.

4.4. Qualitative results

Object insertion. We present qualitative results of object insertion in Fig. 3. As shown, our method (Fig. 3-f) can generate plausible contents with view consistency. Moreover, the outputs of our method also match well the text prompts and the generated objects are precisely located.

In contrast, the traditional 3D pipeline (Fig. 3-b) suffers from seamed object-background boundaries and unrealistic lighting. The geometry in obscure regions, e.g., corners, is not accurately reconstructed. The image inpainting baseline (Fig. 3-c) produces reasonable results on individual views but generates view-inconsistency. Single-image-to-3D NeRF (Fig. 3-d) fails on complex scenes. A larger camera pose shift can result in background mismatch or even content collapse. Instruct-NeRF2NeRF (Fig. 3-e) is known to be strong at stylizing existing geometry but weak at generating new geometry. Under our setting, it performs random view-consistent editing, but fails to accomplish object insertion.

Object removal. We illustrate several results of object removal in Fig. 4, which shows that our method can generate view-consistent backgrounds. We also observed that without using pseudo ground-truth background, the removal can cause gradual background collapse, which we discuss further in our ablation study in Sec. 4.6.

4.5. Quantitative results

Since there is no real-world ground-truth for scene editing, direct quantitative evaluations are not possible. Instead, we adopt the CLIP Score [16], a well-known metric to measure how well an image correlates to a target prompt in the CLIP space. Let $E(I)$ and $E(p)$ be the CLIP embeddings of an image I and a text prompt p . The CLIP Score is defined as,

$$\text{CLIPScore}(I, p) = \cos^+(E(I), E(p)) \quad (8)$$

where $\cos^+(a, b) = \max(0, \cos(a, b))$. We average the $\text{CLIPScore}(\hat{I}_b, \hat{p})$ for all the edited images $\hat{I}_b \in \hat{\mathbf{I}}_b$ paired with their target prompts \hat{p} , and use this average value as a performance metric for scene editing; higher score means better performance.

We also use the CLIP Directional Consistency (CLIPDC) in [14] to measure the editing quality and consistency across views in the CLIP space. This metric relies on an assumption that the difference in the CLIP space between a background

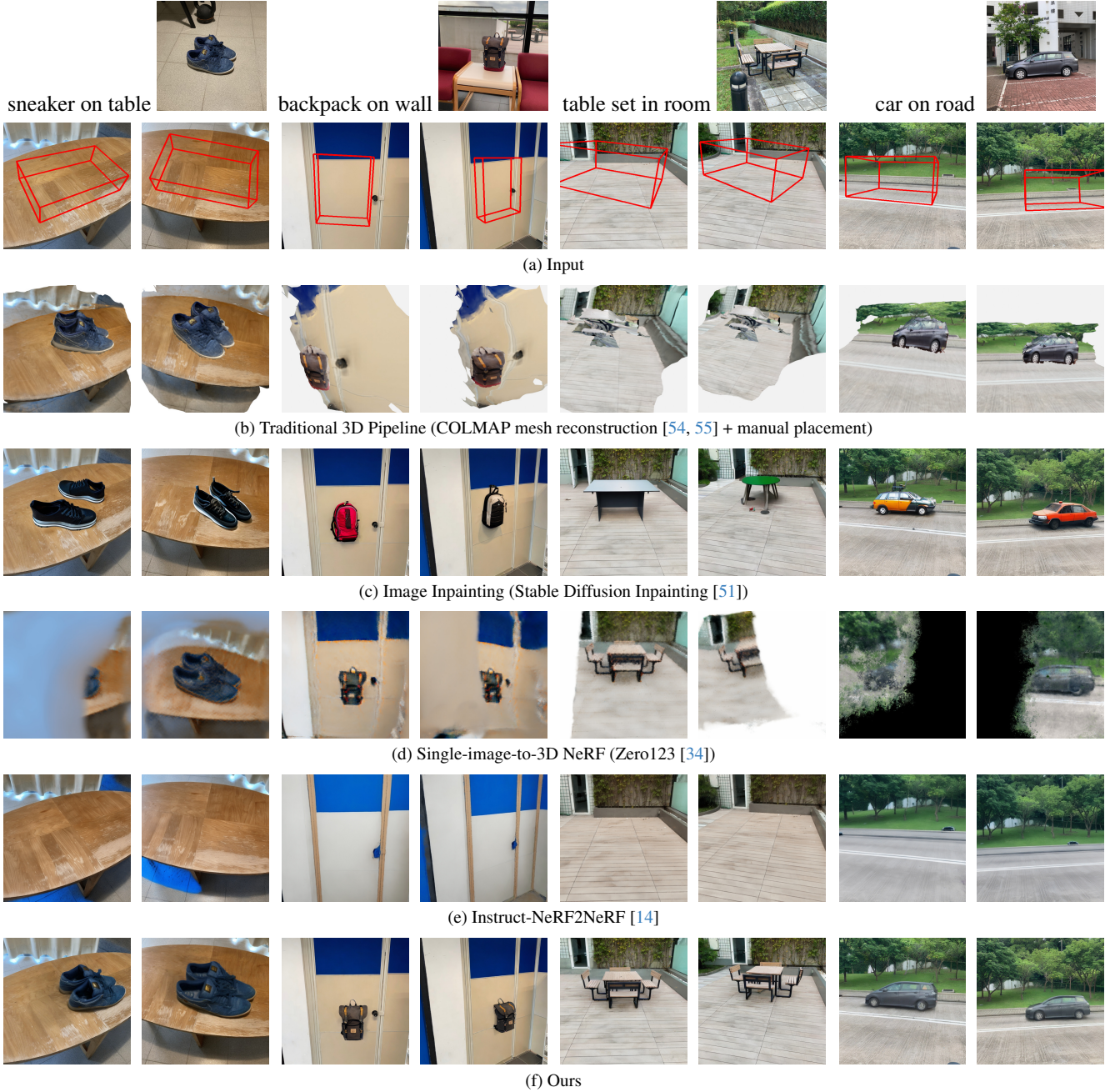


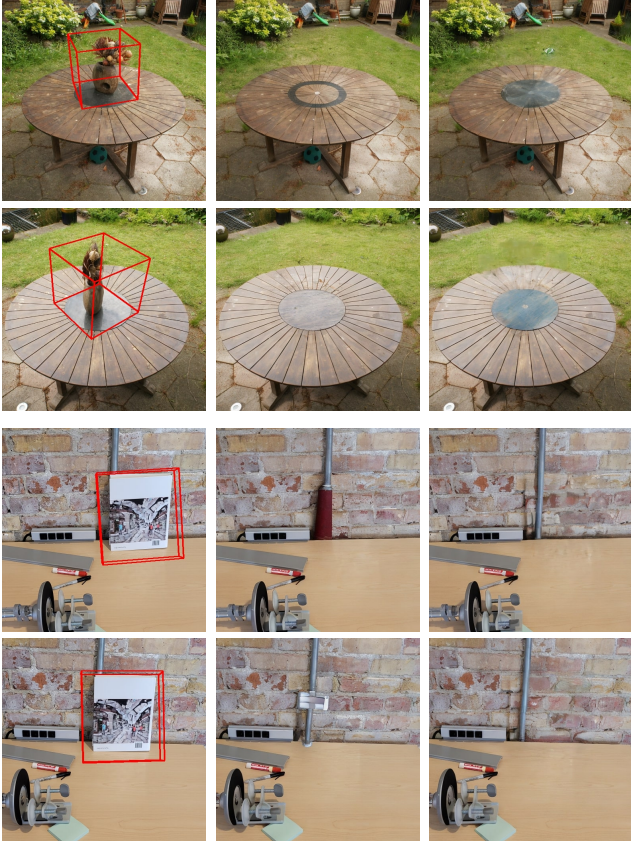
Figure 3. **Qualitative results of object insertion.** Inputs include multi-view object/background images, a 3D bounding box where the object is inserted in, and a text prompt. Note that some baselines use parts of the inputs due to the nature of their techniques.

image I_b and its edited version \hat{I}_b should match with the difference between the original prompt p (paired with I_b) and the customized prompt \tilde{p} (paired with \hat{I}_b). Moreover, a good editing should make consistent differences between every pair of I_b and \hat{I}_b , especially for adjacent views. We follow [14] to calculate the CLIPDC between two adjacent

edited views $\hat{I}_b^{(i)}$ and $\hat{I}_b^{(i+1)}$ as,

$$\begin{aligned} \text{CLIPDC}(\hat{I}_b^{(i)}, \hat{I}_b^{(i+1)}) = & \\ & \cos^+ \left(E(\tilde{p}) - E(p), E(\hat{I}_b^{(i)}) - E(I_b^{(i)}) \right) \times \quad (9) \\ & \cos^+ \left(E(\hat{I}_b^{(i)}) - E(I_b^{(i)}), E(\hat{I}_b^{(i+1)}) - E(I_b^{(i+1)}) \right) \end{aligned}$$

We measure the CLIPDC of an edited scene by averaging



(a) Original BG with object bounding box (b) Pseudo ground-truth BG (c) View-consistent editing result

Figure 4. **Qualitative results of object removal** where we show the importance of pseudo ground-truth background (BG) in generating view-consistent editing results.

Method	CLIPScore \uparrow	CLIPDC \uparrow
Traditional 3D [54, 55]	0.2648	0.1169
Inpainting [51]	0.2683	0.1325
Zero123 [34]	0.2197	0.0361
Instruct-NeRF2NeRF [14]	0.2347	0.0263
Ours	0.2743	0.1642

Table 1. **Quantitative results.** Higher CLIP metrics scores indicate higher image editing quality or consistency.

the $\text{CLIPDC}(\hat{I}_b^{(i)}, \hat{I}_b^{(i+1)})$ for all the adjacent image pairs $(\hat{I}_b^{(i)}, \hat{I}_b^{(i+1)})$ in the scene. We report both the CLIPScore and CLIPDC of our method and other baselines on 35 unique edits (cross editing on the more generalizable 5 scenes \times 7 objects) in Tab. 1. Experimental show that our method outperforms all the baselines on both the CLIPScore and CLIPDC metrics.

4.6. Ablation Study

In this ablation study, we validate the effectiveness of technical components in our method. We suggest readers observe the results in Fig. 5, which shows two views of an edited scene. We provide detailed analyses on these results below.

Diffusion model fine-tuning. Recall that we fine-tune the diffusion model D_θ on both object and background images (in Sec. 3.2). Here we prove that such a fine-tuning is necessary. Fine-tuning on object images makes the diffusion model aware of the same object during the dataset updates (in Sec. 3.3). We verify this in Fig. 5-a, where we skip the fine-tuning of D_θ on object images. As shown, without using object images, the diffusion model cannot generate the same object across views.

Likewise, fine-tuning the diffusion model with background images helps to preserve the background at inpainted borders (areas in between mask boundary and inner-mask object boundary). As shown in Fig. 5-b, without fine-tuning on background images, the model cannot inpaint the background within the mask region properly. The inpainted background gets darker at latter NeRF updating steps, and finally ends up with a noticeable error. We hypothesize this collapse that the model, without being fine-tuned on relevant background images, applies the bias from pre-learned knowledge which does not align with the surrounding background.

Pose-conditioned dataset updates. To prove the effectiveness of the pose-conditioned dataset update strategy, we experiment with another scheme in which the dataset used to refine the NeRF is updated with random views. We present the results of this experiment in Fig. 5-c. We observe that, with view-random updates, objects in different views can converge into inconsistent poses (e.g., the hats in the left and right view are generated in different poses).

The above inconsistency commonly happens with many text-to-3D NeRF generation or editing methods. We hypothesize this phenomenon as follows. Initially the diffusion model can generate the object in inconsistent poses across views due to little 3D-aware hint available. This inconsistency is continually introduced to the NeRF updating and then, in return, passed partially as input to the diffusion model (as in Eq. 6), making further pose divergence eventually. Without considering nearby-views, the diffusion model cannot fix this inconsistency as objects generated on individual views match well their given text prompts.

Our proposed pose-conditioned dataset update strategy simulates the nature of NeRF construction, in which a view rendered closed to already-used views should contain similar but slightly blurry and distorted object content. Pose-conditioned view arrangement thus gives the diffusion model enough hints to generate objects with view-consistent appearance and poses, specified in nearby views. View-consistency is thus achieved progressively in the training dataset and is integrated into the constructed NeRF (see Fig. 5-e).

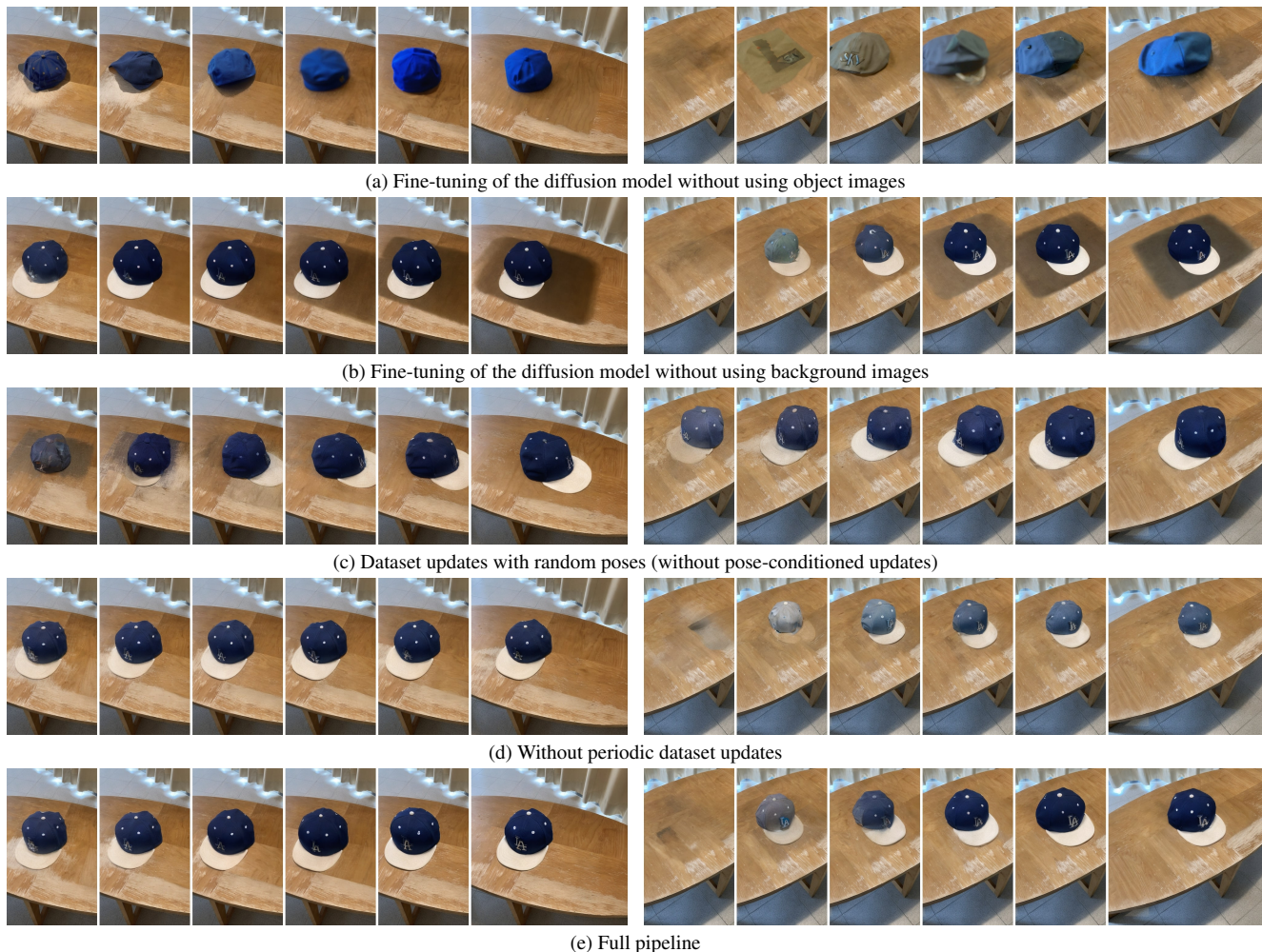


Figure 5. **Ablation study results.** Each row shows the results of a variant of our pipeline. The left and right columns include images of two different views of an edited scene. For each column, from left to right are the results of increasing training steps, where the most left image in each column is an early-stage result and the most right is the final output. The left column is a view near the starting view, which converges faster than the right column from a farther view (except for the variant in (c) where both views converge equally fast).

Periodic dataset updates. We observe that periodically updating of training views during the NeRF updating is important to achieve high-quality renderings. We verify this by keeping all the training views from the pose-conditioned dataset updates fixed during the NeRF updating. We found that although objects are rendered at fairly accurate orientations, their texture and geometry still suffer from inconsistencies (see Fig. 5-d). Having periodically updates on the training data fixes these minor defects and facilitates a more consistent convergence of the NeRF updating (see Fig. 5-e).

5. Conclusion

We propose a new language-driven method for object manipulation in NeRFs. Our method is built upon a novel idea of joint 2D-3D interaction, keeping both 2D image synthesis

and 3D NeRF reconstruction in a loop. This idea is enabled by an advanced text-to-image diffusion technique that generates object-blended background images, and a novel pose-conditioned dataset update strategy that learns a NeRF from the multi-view images in a progressive manner.

Our method is not without limitations. Since our 2D views are synthesized by a diffusion model, we may share the flickering problem with diffusion-based video editing methods [26, 48, 68]. We leave this for future work, which can potentially be addressed by robust video translation methods [73]. It is also of great interest to postulate a theoretical foundation for pose-conditioned dataset update to better understand the convergence of NeRF training in scene editing.

Acknowledgment. This work was partially supported by a grant from the RGC of HKSAR, China (Project No. HKUST 16202323) and an internal grant from HKUST (R9429).

References

- [1] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven nerf editing with prior-guided editing field. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20919–20929, 2023. [2](#)
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5855–5864, 2021. [2](#)
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5470–5479, 2022. [5](#)
- [4] 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. [2](#)
- [5] Ronald A Castellino. Computer aided detection (cad): an overview. Cancer Imaging, 5(1):17, 2005. [1](#)
- [6] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16123–16133, 2022. [2](#)
- [7] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. arXiv preprint arXiv:2303.11396, 2023. [2](#)
- [8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5939–5948, 2019. [2](#)
- [9] Blender Online Community. Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [1](#)
- [10] Epic Games. Unreal engine. [1](#)
- [11] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. Science, 360(6394):1204–1210, 2018. [2](#)
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014. [1](#), [2](#)
- [13] John K Haas. A history of the unity game engine. 2014. [1](#)
- [14] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. arXiv preprint arXiv:2303.12789, 2023. [2](#), [5](#), [6](#), [7](#)
- [15] Jeffrey Harper. Mastering Autodesk 3ds Max 2013. John Wiley & Sons, 2012. [1](#)
- [16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718, 2021. [5](#)
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. [1](#), [2](#)
- [18] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2821–2830, 2018. [2](#)
- [19] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18342–18352, 2022. [2](#)
- [20] Yi-Hua Huang, Yan-Pei Cao, Yu-Kun Lai, Ying Shan, and Lin Gao. Nerf-texture: Texture synthesis with neural radiance fields. In ACM SIGGRAPH 2023 Conference Proceedings, pages 1–10, 2023. [2](#)
- [21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017. [2](#)
- [22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. [2](#)
- [23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. Advances in neural information processing systems, 33:12104–12114, 2020. [2](#)
- [24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8110–8119, 2020. [2](#)
- [25] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6007–6017, 2023. [2](#)
- [26] Gyeongman Kim, Hajin Shim, Hyunsu Kim, Yunjey Choi, Junho Kim, and Eunho Yang. Diffusion video autoencoders: Toward temporally consistent face video editing via disentangled video encoding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6091–6100, 2023. [8](#)
- [27] Hyunsu Kim, Gayoung Lee, Yunjey Choi, Jin-Hwa Kim, and Jun-Yan Zhu. 3d-aware blending with generative nerfs. arXiv preprint arXiv:2302.06608, 2023. [2](#)
- [28] 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. 2
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2
- [30] Gang Li, Heliang Zheng, Chaoyue Wang, Chang Li, Changwen Zheng, and Dacheng Tao. 3ddesigner: Towards photo-realistic 3d object generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2211.14108*, 2022. 2
- [31] 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. 2
- [32] 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
- [33] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4616–4624, 2018. 2
- [34] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023. 5, 6, 7
- [35] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5773–5783, 2021. 2
- [36] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2
- [37] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2
- [38] 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
- [39] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019. 2
- [40] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 5
- [41] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2
- [42] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinstein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 2
- [43] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 4
- [44] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *arXiv preprint arXiv:2207.02363*, 2022. 2
- [45] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 2
- [46] Hong-Wing Pang, Binh-Son Hua, and Sai-Kit Yeung. Locally stylized neural radiance fields. In *ICCV*, 2023. 2
- [47] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 5
- [48] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535*, 2023. 8
- [49] 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. 2
- [50] 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. 1, 2
- [51] 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, 2, 3, 5, 6, 7
- [52] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2, 4
- [53] 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. 1, 2
- [54] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5, 6, 7
- [55] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 4, 5, 6, 7
- [56] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *Advances in Neural Information Processing Systems*, 35:33999–34011, 2022. 2
- [57] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *Advances in Neural Information Processing Systems*, 35:24487–24501, 2022. 2
- [58] Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. Blending-nerf: Text-driven localized editing in neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14383–14393, 2023. 2
- [59] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2
- [60] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2897–2905, 2018. 2
- [61] 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. 2
- [62] 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. 2
- [63] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. Inpaintnerf360: Text-guided 3d inpainting on unbounded neural radiance fields. *arXiv preprint arXiv:2305.15094*, 2023. 2
- [64] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 5
- [65] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2
- [66] 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
- [67] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023. 2
- [68] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. 8
- [69] Tong Wu, Jia-Mu Sun, Yu-Kun Lai, and Lin Gao. De-nerf: Decoupled neural radiance fields for view-consistent appearance editing and high-frequency environmental relighting. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2
- [70] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 2
- [71] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 3
- [72] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4402–4412, 2023. 2
- [73] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. *arXiv preprint arXiv:2306.07954*, 2023. 8
- [74] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2
- [75] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yueshen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2
- [76] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2
- [77] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

- [78] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. ACM Transactions on Graphics (ToG), 40(6):1–18, 2021. [2](#)
- [79] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pages 286–301. Springer, 2016. [2](#)
- [80] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017. [2](#)
- [81] 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. [2](#)