ConsistDreamer: 3D-Consistent 2D Diffusion for High-Fidelity Scene Editing

Jun-Kun Chen[1†] Samuel Rota Bulò[2] Norman Müller[2] Lorenzo Porzi[2]
Peter Kontschieder[2] Yu-Xiong Wang[1]
[1]University of Illinois Urbana-Champaign [2]Meta
{junkun3,yxw}@illinois.edu
{rotabulo,normanm,porzi,pkontschieder}@meta.com

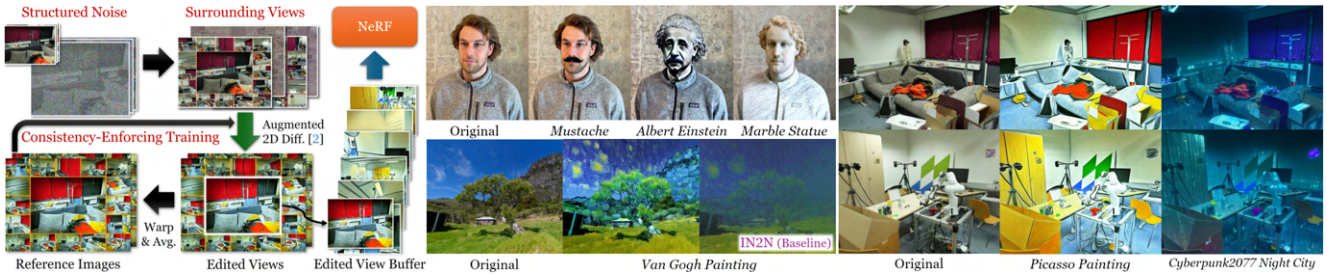


Figure 1. **Our ConsistDreamer lifts 2D diffusion with 3D awareness and consistency**, achieving high-fidelity instruction-guided scene editing with superior sharpness and detailed textures. **Left:** The three synergistic components within ConsistDreamer that enable 3D consistency. **Right:** State-of-the-art performance of ConsistDreamer across various editing tasks and scenes, especially when prior work (*e.g.*, IN2N [8]) fails and in challenging large-scale indoor scenes from ScanNet++ [43]. More results are on our project page.

Abstract

This paper proposes ConsistDreamer – a novel framework that lifts 2D diffusion models with 3D awareness and 3D consistency, thus enabling high-fidelity instruction-guided scene editing. To overcome the fundamental limitation of missing 3D consistency in 2D diffusion models, our key insight is to introduce three synergistic strategies that augment the input of the 2D diffusion model to become 3D-aware and to explicitly enforce 3D consistency during the training process. Specifically, we design surrounding views as context-rich input for the 2D diffusion model, and generate 3D-consistent structured noise instead of image-independent noise. Moreover, we introduce self-supervised consistency-enforcing training within the per-scene editing procedure. Extensive evaluation shows that our ConsistDreamer achieves state-of-the-art performance for instruction-guided scene editing across various scenes and editing instructions, particularly in complicated large-scale indoor scenes from ScanNet++, with significantly improved sharpness and fine-grained textures. Notably, ConsistDreamer stands as the first work capable of successfully editing complex (e.g., plaid/checkered) patterns. Our project page is at immortalco.github.io/ConsistDreamer.

1. Introduction

With the emergence of instruction-guided 2D generative models as in [2], it has never been easier to generate or edit images. Extending this success to 3D, *i.e.*, instruction-guided 3D scene editing, becomes highly desirable for artists, designers, and the movie and game industries. Nevertheless, editing 3D scenes or objects is inherently challenging. The absence of large-scale, general 3D datasets makes it difficult to create a counterpart generative model similar to [2] that can support arbitrary 3D scenes. Therefore, state-of-the-art solutions [8, 25] circumvent this challenge by resorting to generalizable 2D diffusion models. This approach, known as *2D diffusion distillation*, renders the scene into multi-view images, applies an instruction-conditioned diffusion model in 2D, and then distills the editing signal back to 3D, such as through a neural radiance field (NeRF) [6, 8, 15].

However, a fundamental limitation of this solution is *the lack of 3D consistency*: a 2D diffusion model, acting independently across views, is likely to produce inconsistent edits, both in color and shape. For example, a person in one view might be edited to be wearing a red shirt, while appearing in a green shirt in another view. Using these images to train a NeRF can still produce reasonable edits, but the model will naturally converge towards an "averaged" representation of the inconsistent 2D supervision, and lose

[†]Work started during an internship at Meta Reality Labs Zurich.

most of its details and sharpness. A commonly observed failure mode is that of regular (*e.g.*, checkered) patterns, which completely disappear once distilled to 3D due to misalignments across views. Generating consistent multi-view images thus becomes crucial for achieving high-fidelity 3D scene editing.

While largely overlooked in prior work, our investigation reveals that the source of inconsistency is *multi-faceted*, and primarily originates from the *input*. **(1)** As the 2D diffusion model can only observe a single view at a time, it *lacks sufficient context* to understand the entire scene and apply consistent editing. **(2)** The editing process for each image starts from *independently generated Gaussian noise*, which brings challenges to consistent image generation. Intuitively, it is difficult to generate consistent multi-view images by denoising inconsistent noise, and even for a single view, it may not always yield the same edited result. **(3)** The input to the 2D diffusion model contains *no 3D information*, making it much harder for the model to reason about 3D geometry and to share information across different views of the scene, even when made available to it.

Motivated by these observations, we propose ConsistDreamer – a novel framework to achieve 3D consistency in 2D diffusion distillation. ConsistDreamer introduces three *synergistic* strategies that augment the input of the 2D diffusion model to be 3D-aware and enforce 3D consistency in a self-supervised manner during the training process.

To address the limited context issue within a single view, our framework involves incorporating context from other views. We capitalize on the observation that 2D diffusion models inherently support "composed images," where multiple sub-images are tiled to form a larger image. Given the capability of the self-attention modules in the UNet of the 2D diffusion model to establish connections between the same objects across different sub-images, each image can be edited with the context derived from other images. Therefore, we leverage the composed images to construct a *surrounding view* (Fig. 1), where one large, central main view is surrounded by several small reference views. This approach allows us to edit the main view with the context from reference views, and vice versa. Doing so not only enriches the context of the scene in the input, but also enables the simultaneous editing of multiple views.

Regarding the noise, we introduce *3D-consistent structured noise* (Fig. 1), with the key insight of generating consistent noise for each view once at the beginning. Specifically, we generate and fix Gaussian noise on the surface of the scene objects, and then render each view to obtain the 2D noise used for the image at that view in all subsequent diffusion generations. This approach aligns with existing 3D diffusion work [22] which also generates noise in 3D at the beginning of a generation. Ensuring that the denoising procedure starts with consistent noise substantially facilitates the process of achieving consistent images by the end.

The combination of surrounding views and structured noise provides the 2D diffusion model with 3D consistent input, yet it is insufficient. An explicit enforcement of 3D consistency is also required during the learning process. To this end, we propose *self-supervised consistency-enforcing training* within the per-scene editing procedure (Fig. 1). We augment the 2D diffusion model by a ControlNet [47] that introduces 3D positional embedding to make it 3D-aware. Inspired by [37, 42], we perform warping and averaging for all sub-views in the edited surrounding view image. This process yields a surrounding view of 3D consistent sub-views used as the self-supervision target. To further achieve "cross-batch consistency" – consistency between different batches in different generations – we perform multiple generations in parallel, and construct consistent target images from all sub-views in all generated surrounding view images, so as to supervise all generations collectively. After consistency-enforcing training, the 2D diffusion model is able to generate consistent multi-view images. Consequently, a trained NeRF will not have to smooth out inconsistencies, but ultimately converge to sharp results preserving fine-grained details.

Empowered by such a 3D-consistent 2D diffusion model, our ConsistDreamer achieves high-fidelity and diverse instruction-guided 3D scene editing *without* any mesh exportation and refinements or a better scene representation like Gaussian Splatting [14], as shown in Fig. 1. Compared with previous work, the editing results of ConsistDreamer exhibit significantly improved sharpness and detail, while preserving the diversity in the original 2D diffusion model's [2] editing results. Notably, ConsistDreamer stands as the *first* work capable of successfully editing complex (*e.g.*, checkered) patterns. Moreover, ConsistDreamer demonstrates superior performance in complicated, high-resolution ScanNet++ [43] scenes – an accomplishment where state-of-the-art methods faced challenges in achieving satisfactory edits.

Our contributions are three-fold. **(1)** We introduce ConsistDreamer, a simple yet effective framework that enables 3D-consistent instruction-guided scene editing based on distillation from 2D diffusion models. **(2)** We propose three novel, synergistic components – structured noise, surrounding views, and consistency-enforcing training – that lift 2D diffusion models to generate 3D-consistent images across all generated batches. Notably, our work is the *first* that explores cross-batch consistency and denoising consistency in 2D diffusion distillation and attains these through manipulating noise. **(3)** We evaluate a range of scenes and editing instructions, achieving state-of-the-art performance in both, scenes considered by previous work and more complicated, large-scale indoor scenes from ScanNet++.

2. Related Work

NeRF-Based Scene Editing. Neural radiance field (NeRF) [21] and its variants [1, 3, 7, 19, 33, 35, 38, 44] are widely-used approaches to representing scenes. NeRF leverages neural networks or other learnable architectures to learn to reconstruct the 3D geometry of a scene only from multi-view images and their camera parameters, and support novel view synthesis. With the development of NeRF, editing a NeRF-represented scene is also deeply studied, covering different types of editing objectives and editing operation indicators, *a.k.a.*, "user interfaces." Some methods [16, 20, 41] support editing the position, color, and/or shape of a specific object indicated by users through a pixel, a text description, or a segment, *etc*. Another line of work [4, 24, 39, 46] studies human-guided shape editing, which allows users to indicate a shape editing operation with a cage or point cloud provided by the model. The task we investigate is instruction-guided scene editing, which allows users to indicate the editing operation through instructions in natural language. The first work in this direction is NeRF-Art [34], which mainly focuses on style transfer instructions, and uses pre-trained CLIP [26] as the stylization loss for the instruction-indicated style. More recent work [8, 13, 15, 45, 49] leverages diffusion models [10] instead of CLIP to benefit from powerful diffusion models and support more general instructions.

Distillation-Based 3D Scene Generation. Lacking 3D datasets to train powerful 3D diffusion models, current solutions distill the generation signal from a 2D diffusion model to exploit its ability in 3D generation. DreamFusion [25] is the first work in this direction, which proposes score distillation sampling (SDS) to distill the gradient update direction ("score") from 2D diffusion models, and supports instruction-guided scene generation by distilling a pre-trained diffusion model [28]. HiFA [48] proposes an annealing technique and rephrases the distillation formula to improve the generation result. Magic3D [18] improves the generation results by introducing a coarse-to-fine strategy and a mesh exportation and refinement method. ProlificDreamer [36] further improves the generation results by introducing an improved version of SDS, namely variational score distillation (VSD), to augment and fine-tune a pre-trained diffusion model and use it for generation.

Diffusion Distillation-Based 3D Scene Editing. Similar to [28] for instruction-guided generation tasks, another diffusion model [2] was proposed for instruction-guided image editing, by generating the edited image conditioned on both the original image and the instruction, which is therefore compatible with SDS [25]. Instruction 3D-to-3D [13] uses SDS with [2] to support instruction-guided style transfer on 3D scenes. Instruct-NeRF2NeRF (IN2N) [8] adopts another way to operate the 2D diffusion model, similar to the rephrased version of SDS in HiFA [48], which iteratively generates edited images to update the NeRF dataset for NeRF fitting, and supports more general editing instructions such as object-specific editing. ViCA-NeRF [6] proposes a different pipeline to first edit key views and then blend key views and apply refinement. Edit-DiffNeRF [45] augments the diffusion model and fine-tunes it with a CLIP loss to improve the success rate of editing. DreamEditor [49] utilizes a fine-tuned variant of [29] instead of [2] and focuses on object-specific editing.

Consistency in Distillation-Based Pipelines. When distilling from 2D diffusion to perform 3D generation or editing, 3D awareness and 3D consistency of the generated images are crucial, as 3D-inconsistent multi-view images are not a valid descriptor of a scene. However, achieving 3D consistency in 2D diffusion is challenging. Early work [8, 13, 18, 25, 48] does not alter the diffusion model and relies on consistency derived from NeRF, by directly training NeRF with the inconsistent multi-view images. The NeRF will then converge to an averaged or smoothed version of the scene, according to its model capability, which results in blurred results with few textures and even fails to generate regular patterns like a plaid or checkered pattern. Follow-up work begins to improve the consistency of the diffusion and/or the pipeline. ViCA-NeRF [6] achieves consistency by proposing a different pipeline based on key views. ProlificDreamer [36] makes the diffusion 3D-aware by inputting the camera parameter to the diffusion model and applying per-scene fine-tuning. CSD [15], IVID [37], and ConsistNet [42] propose a joint distillation procedure for multiple views, aiming to generate or edit multiple images in one batch consistently, through either attention, depth-based warping, or Kullback–Leibler divergence.

However, these methods all share two major constraints: (1) the noise used for generation is not controlled, therefore a single view may lead to different and inconsistent generation results with different noises; (2) these methods only study and enforce the consistency between images within a single batch. Nevertheless, the full generation or editing procedure for the scene is across multiple batches, and there might be inconsistencies in different batches. Our ConsistDreamer resolves these limitations by proposing novel structured noise and consistency-enforcing training.

3. ConsistDreamer: Methodology

Our ConsistDreamer is a novel IN2N-like [8] framework applied upon a diffusion-based 2D image editing model [2]. As illustrated in Fig. 2, our pipeline maintains a buffer of edited views for the NeRF to fit, and uses [2] to generate new edited images for random views according to the instruction, the original appearance, and the current NeRF rendering results. Noticing that the NeRF fitting procedure and diffusion generation procedure are relatively independent, we equivalently execute them in parallel. Within

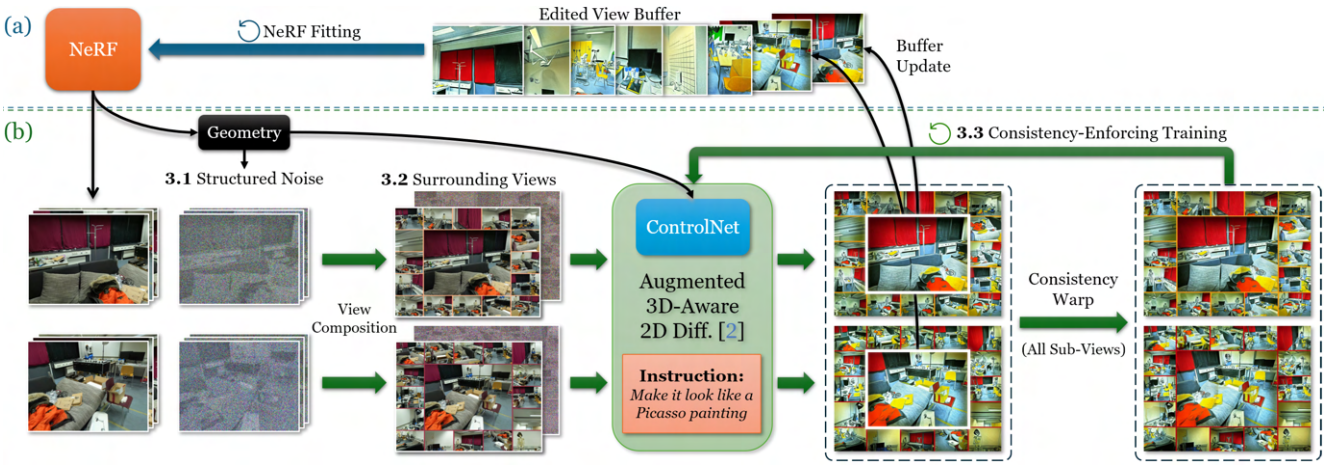


Figure 2. **ConsistDreamer framework** is an IN2N-like [8] pipeline containing two major procedures. **(a)** In the NeRF fitting procedure, we continuously train NeRF with a buffer of edited views. **(b)** In diffusion generation and training, we add our 3D-consistent structured noise to rendered multi-view images, and compose surrounding views with them, as input to the augmented 3D-aware 2D diffusion [2]. We then add the edited images to the buffer for (a), and apply self-supervised consistency-enforcing training using the consistency-warped images. Note: images of structured noise are only for illustration – they are actually visually indistinguishable from Gaussian noise images.

this framework, we propose (1) *structured noise* to enable a 3D-consistent denoising step, starting from 3D-consistent noise at the beginning and ending with 3D-consistent images; (2) *surrounding views* to construct context-rich composed images as input to the 2D diffusion instead of a single view; and (3) a self-supervised *consistency-enforcing training* method via consistent warping in surrounding views, to achieve cross-view and cross-batch consistency.

3.1. Structured Noise

2D diffusion models generate a new image from a noisy image, which is either pure Gaussian noise or a mixture of noise and the original image. Prior works like DreamFusion [25] and IN2N typically sample different Gaussian noise in each iteration. However, varying noise leads to highly different generation results (as shown in the supplementary material). In other words, previous methods cannot even produce consistent (*i.e.*, identical) images for the same view in different generations, fundamentally limiting their ability to generate consistent results.

This observation motivates us to control and manipulate the noise, by introducing 3D-consistent structured noise. Intuitively, while it is difficult to generate, denoise, or restore 3D-consistent images from inconsistent random noise, the task becomes more manageable when generating consistent images from noise that is itself consistent. Therefore, instead of using independently generated noise in each iteration, we generate noise on the surface of the scene *only once* during initialization, and render the noise at each view to obtain the noise used in generating the image for that view. Our strategy aligns with 3D diffusion models like DiffRF [22], which directly generate noise in 3D space. The difference lies in the denoising step: while such work directly denoises in 3D, we distill the "3D denoising process" from pre-trained 2D diffusion models.

As a latent diffusion model, [2] actually requires noise in latent space, which is $(H/8, W/8, 4)$ instead of the image shape $(H, W, 3)$. Each element in this noise latent should be independently generated from $N(0, 1)$. Constructing such 3D-consistent structured noise remains non-trivial: we need to place noise in 3D, project noise into 2D pixels at multiple scales, and ensure correspondence between different views. Additionally, the distribution of each image's noise should be Gaussian, as noise in an incorrect or dependent distribution may lead to abnormal generation results (as shown in the supplementary material).

To overcome these challenges, we construct a dense point cloud of the scene by unprojecting all the pixels in all the views to points, with the depth predicted by NeRF. For each point p, we randomly sample a weighted noise $c(p) = (x, w)$, where $x \sim N(0, 1)$ is an independently generated Gaussian noise, and $w \sim U(0, 1)$ is its weight. To generate the noise at one view, we identify the sub-point cloud that is front-most in this view, and project it onto the plane. For multiple points projected to the same pixel, we aggregate them by selecting the weighted noise (x, w) with the maximum w, and form a noise image I of shape (H, W) consisting of values in x. As each x is independently generated and selected (according to w), we have $I \sim N(0, 1)^{H \times W}$, *i.e.*, making I valid 2D Gaussian noise.

Given that each pixel in the latent space is only roughly related to its corresponding 8×8 region in the image, we can generate noise in the latent space by operating at the downsampled resolution of $(H/8) \times (W/8)$. We thus generate different weighted noise $\{c_i(p)\}$ for each of the four channels of the latent space, and stack the individually rendered noise I_i to construct a Gaussian noise image of $(H/8, W/8, 4)$, which is then used as the noise by the

diffusion model.

The structured noise serves as the foundation for 3D-consistent generation. In Sec. 3.3, we introduce a training method to ensure a consistent denoising procedure from the beginning to the end, so that the denoised images of different views at every denoising step is also 3D consistent.

3.2. Surrounding Views

Using the original view as input is a standard practice, when employing 2D diffusion models. This method works well in simple 360° or forward-facing scenes used by IN2N [8], as a single view covers most objects in the scene. However, in more complicated scenes like the cluttered rooms in ScanNet++ [43], a view may only contain a corner or a bare wall in the room. This hinders the diffusion model to generate plausible results, due to the limited context in a single view.

Intriguingly, our investigation reveals that [2] performs well on composed images, generating an image composed of style-consistent edited sub-images with the same structure (as shown in the supplementary). This observation inspires us to exploit a novel input format for diffusion models – surrounding views with a composition of one main view and many reference views, so that all views collectively provide contextual information. As illustrated in Fig. 2, the key principles in the design of a surrounding view are: (1) the main view that we focus on in this generation should occupy a large proportion; and (2) it should include as many reference views as possible at a reasonable size to provide context. In practice, we construct a surrounding view w.r.t. a specific main view, by surrounding a large image of this view with $4(k-1)$ small reference images of other views, leaving a margin of arbitrary color. This ensures that the main image is roughly $(k-2)$ times larger than the small images. Here k is a hyperparameter. The reference images are randomly selected from all the views or nearby views from the main view, providing both a global picture of the scene and much overlapped content to benefit training.

We use such surrounding views as input images to [2], by constructing the surrounding views of the current NeRF's rendering results, structured noise, and original views. Though not directly trained with this image format, [2] still supports generating edited images in the same format, with each image corresponding to the edited result of the image in the same position. The attention modules in its UNet implicitly connect the same regions in different views, enabling the small views to provide extra context to the main view. This results in consistently edited styles among all the sub-images in the surrounding view image.

The surrounding views not only provide a context-rich input format for 2D diffusion models, but also allow it to generate edited results for $(4k-3)$ views in one batch, benefiting our consistency-enforcing training in Sec. 3.3.

3.3. Consistency-Enforcing Training

We design consistent per-scene training based on structured noise and surrounding views, enforcing 2D diffusion to generate 3D consistent images through a consistent denoising procedure.

Multi-GPU Parallelization Paradigm. Our pipeline involves training both NeRF and 2D diffusion. Observing that training and inferring a diffusion model is considerably more time-consuming than training the NeRF, while there are very few dependencies between them, we propose a multi-GPU parallelization paradigm. With $(n+1)$ GPUs, we dedicate GPU0 to continuously and asynchronously train a NeRF on the buffer of edited images. The remaining n GPUs are utilized to train the diffusion model and generate new edited images added to the buffer for NeRF training. At each diffusion training iteration, we allocate a view to each of the n GPUs and train diffusion on them synchronously. This parallelization eliminates the need to explicitly trade off between NeRF and diffusion training, leading to a **10**× speed-up in training. With multiple diffusion generations running synchronously, we can also enforce *cross-generation consistency*.

Augmenting 2D Diffusion with 3D-Informing ControlNet. Intuitively, a 3D-consistent model needs to be 3D-aware; otherwise, it lacks the necessary information and may solely adapt to the input structured noise, potentially leading to overfitting. Therefore, we incorporate an additional ControlNet [47] adaptor into our 2D diffusion, which injects 3D information as a new condition. The 3D information is obtained by using NeRF to infer the depth and 3D point for each pixel in the view. We then query its feature in a learnable 3D embedding (implemented as a hash table in [23]) to acquire a pixel-wise 3D-aware feature image, which serves as the condition for ControlNet. These components make the augmented diffusion to be aware of, learn from, and generate results based on 3D information. Additionally, we apply LoRA [11] to further enhance the capability of diffusion.

Self-Supervised Consistency Loss. Lacking ground truth for consistently edited images, we introduce a self-supervised method to enforce 3D consistency. For a set of generated multi-view images, we construct a corresponding reference set of 3D consistent multi-view images to serve as a self-supervision target. Inspired by [37, 42], we employ depth-based warping with NeRF-rendered depth to establish pixel correspondence across views. We design a weighted averaging process to aggregate these pixels to the final image, ensuring multi-view consistency (detail in supplementary).

Specifically, we edit n surrounding views synchronously on n GPUs, with each surrounding view containing $(4k-3)$ views, resulting in a total of $V=(4k-3)n$ views. For each view v, we warp the edited results of the remaining $V-1$

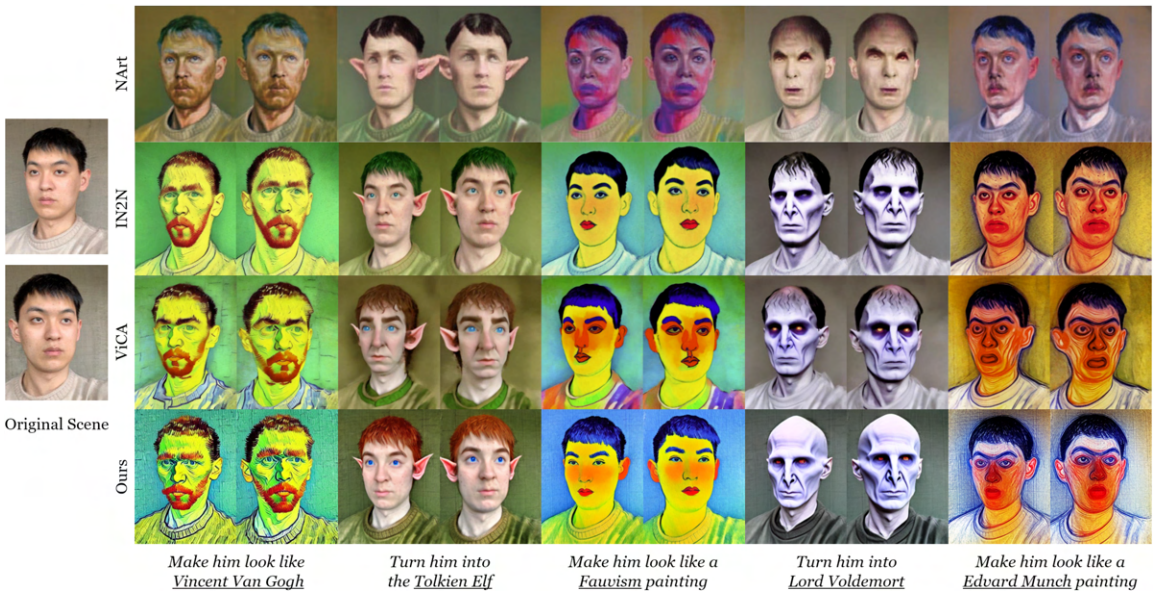


Figure 3. **Comparison in the Fangzhou scene** shows that our ConsistDreamer produces significantly sharper editing results with more fine-grained textures and higher consistency with the instruction, *e.g.*, Lord Voldemort with no hair on which all baselines fail. The instructions are the bottom texts, except for NArt [34], which uses the underlined texts. The images of baselines are taken from their paper.

views to it, and compute their weighted average to obtain the reference view v'. We then re-aggregate reference views $\{v'\}$ back into surrounding views in the original structure for each GPU. These re-assembled surrounding views are then used as the target images to supervise 2D diffusion.

To guide 2D diffusion in preserving the original style and avoiding smoothing out, we define our *consistency loss* as the sum of the VGG-based perceptual and stylization loss [12], instead of a pixel-wise loss, between diffusion's output and the target image. In addition to this primary loss, we propose several regularization losses to prevent mode collapse and promote 3D awareness (detail in supplementary). With the consistency loss, ConsistDreamer effectively enforces not only cross-view consistency among all views in each surrounding view, but also cross-generation or cross-batch consistency for views edited by different GPUs.

Consistent Denoising Procedure. With our structured noise, the denoising in 2D diffusion initiates with consistent noise. This leads to a further goal to make the *entire denoising procedure* 3D consistent and thus end with consistent images. We achieve this by enforcing all the views in the intermediate denoising images to be also 3D consistent at each denoising step. Therefore, unlike the conventional diffusion training with single-step denoising, our training involves a full multi-step denoising procedure with passing through gradients. As it is impossible to fit the entire computational graph into the GPU memory, we use checkpointing [27, 40] to trade space with time. Doing so enables constructing the reference set of images with warping for each intermediate denoising step, which is then used to supervise the intermediate denoising image. This provides more direct signals of 3D consistency in the training of diffusion, facilitating the generation of 3D consistent results.

Shape Editing. Some instructions, *e.g.*, *Make him smile*, change the shape or geometry of the scene during editing, while our structured noise and consistency-enforcing training rely on the geometry. To be compatible with shape editing, we design a *coarse-to-fine* strategy: we first edit the scene using ConsistDreamer with only the surrounding view and disabling the other two components, *i.e.*, using image-independent noise and the original implementation of [2]. This allows the scene to converge to a coarse edited shape according to the instruction. We then activate structured noise and consistency-enforcing training to refine the editing. We periodically adjust the structured noise with changes in geometry, while preserving the noise values. With this strategy, ConsistDreamer also achieves high-fidelity shape editing.

4. Experiments

Editing Tasks. In our setting, each editing task is a pair of `(scene, instruction)`, indicating which instruction-guided editing operation should be applied on which scene. The output of the task is another scene, being the edited scene under the instruction. The scenes we use for evaluation contain two parts: (1) *IN2N*. Scenes used by IN2N [21], including scenes of human faces or bodies, outdoor scenes, and statues; and (2) *SN++*. Scenes in ScanNet++ [43], which are complicated indoor scenes with free-formed structures and camera trajectories. We also use two types of editing instructions: (1) *style transfer* which transfers the style of the scene into the described style, and (2) *object-specific editing* which edits a specific object of the scene.

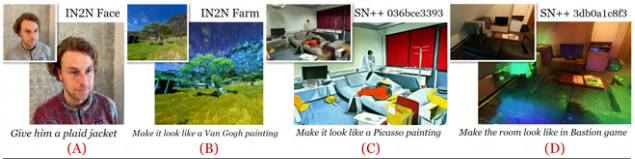
We use these tasks to compare our approach with baselines, and conduct ablation study on representative tasks.

NeRF Backbone and Diffusion Model. For a fair comparison with previous works [6, 8], we use the Nerfacto model in NeRFStudio [32] as our NeRF backbone, and the pre-trained diffusion model [2] from Hugging Face as our initial checkpoint. The NeRF representation for the scene is trained with NeRFStudio in advance, and then used in our pipeline.

ConsistDreamer Variants. We investigate the following variants for our ablation study (where −SN, −SV, and −T denote removing structured noise, surrounding views, and consistency-enforcing training, respectively): (1) Full ConsistDreamer. (2) No structured noise (−SN): use independently generated noise for each view instead of structured noise, but still use surrounding views and perform consistency-enforcing training. (3) No training (−T): use surrounding views and structured noise, but do not augment and train [2] and keep using the original checkpoint. (4) Only surrounding views (−SN −T): only use surrounding views, and do not use structured noise or train [2]. (5) "IN2N" (−SN −SV −T): ours with all the proposed components removed, which can be regarded as an alternative version of IN2N. Note that consistency-enforcing training requires surrounding views to produce sufficient edited views in one generation; we cannot remove surrounding views but still apply consistency-enforcing training on [2].

Baselines. We mainly compare our method with two baselines: Instruct-NeRF2NeRF (IN2N) [8] and ViCA-NeRF (ViCA) [6], as they are most closely related to our task. We also compare with NeRF-Art (NArt) [34] as an early work. Other methods, however, lack publicly available or working code and/or only use a few scenes supported by NerfStudio. Therefore, we could only compare with CSD [15], DreamEditor [49], GE [5], EN2N [31], and PDS [17] under a few tasks in supplementary, and are unable to compare with Edit-DiffNeRF [45] and Instruct 3D-to-3D [13]. Note that ConsistDreamer solves instruction-guided scene editing instead of scene generation, so we do not compare with models for the generation task [25, 37, 42, 48].

Evaluation Metrics. Observing that our ConsistDreamer generates significantly sharper editing results, consistent with previous work [6, 8], we compare ConsistDreamer with baselines mainly through qualitative evaluation. For the ablation study, the appearance of the scenes edited by our different variants may be visually similar and unable to be fairly compared using qualitative results. Therefore, we propose *distillation fidelity score* (DFS) to evaluate how faithful the editing is distilled and applied on NeRF compared with the diffusion's output [2], rooted in the basic setting that we distill from [2] to edit 3D scenes. In this situation, our editing ability is bounded by [2]'s. Consistent with the training objective of DreamFusion [25], we aim to minimize the distance between two distributions: the distribution of a rendered image at a random view from the edited NeRF, and the distribution of the diffusion editing result of an image at a random view in the original scene. Following this, we define the fidelity metric as the Fréchet inception distance (FID) [9, 30] between two sets – the set of images rendered by the edited NeRF at all training views, and the set of edited images generated by the original [2] for all training views, corresponding to these two distributions. A lower FID means a higher fidelity that the editing is applied to the scene.



Variant	Components	A	B	C	D
Full	All	**27.4**$^{\pm 0.4}$	**159.8**$^{\pm 2.1}$	**62.7**$^{\pm 1.9}$	**132.8**$^{\pm 1.2}$
No Str. Noise	−SN	35.0$^{\pm 0.5}$	234.8$^{\pm 2.4}$	83.1$^{\pm 2.2}$	217.8$^{\pm 2.0}$
No Training	−T	34.6$^{\pm 1.1}$	221.8$^{\pm 2.0}$	80.4$^{\pm 2.0}$	201.8$^{\pm 1.9}$
Only Sur. Views	−SN −T	34.0$^{\pm 0.3}$	262.0$^{\pm 2.7}$	83.9$^{\pm 2.2}$	214.4$^{\pm 2.0}$
"IN2N"	−SN −SV −T	91.0$^{\pm 1.2}$	255.8$^{\pm 2.0}$	90.9$^{\pm 1.5}$	222.4$^{\pm 2.0}$

Table 1. **Ablation study** on the distillation fidelity score (↓) *quantitatively* validates the effectiveness and complementarity of each of our components. Our full ConsistDreamer significantly outperforms all variants across various scenes and types of instructions.

Qualitative Results. The qualitative comparison in the Fangzhou scene from the IN2N dataset is shown in Fig. 3. Distilling from the same diffusion model [2], IN2N [8], ViCA [6], and our ConsistDreamer produce results in a similar style. As especially shown in the "Vincent Van Gogh" and "Edvard Munch" editing, our ConsistDreamer generates results containing fine-grained representative textures of Van Gogh and Munch, while the baseline results are blurred and only contain simple or coarse textures. This validates that with our proposed components, ConsistDreamer is able to generate consistent images from [2] with detailed textures, and does not rely on consistency derived from NeRF, which, unfortunately, smooths out the results. Notably, in the "Lord Voldemort" case, our ConsistDreamer is the *only one* that successfully edits the image to resemble the well-known, distinctive appearance of Lord Voldemort, featuring no hair and a peculiar nose. Among all the editing tasks, our ConsistDreamer consistently produces editing results with the most detailed ears and hair/head, and does not contain unnatural color blocks.

Additional qualitative results are shown in Fig. 4, and more results and the comparison with baselines on these tasks are provided in the supplementary and on our project page. Overall, our ConsistDreamer generates sharp, bright editing results in all tasks across various scenes, including human, indoor, and outdoor scenes. (1) In the Face scene, our ConsistDreamer successfully applies the plaid (checkered) jacket editing, a common failure case in most previous

Figure 4. ConsistDreamer *consistently* generates high-quality and high-fidelity editing results, featuring detailed, fine-grained textures across various scenes and instructions. Notably, ConsistDreamer also maintains the high diversity from [2], as exemplified by the highly diversified results (a)(b). **Additional results and comparisons are provided in the supplementary and on our project page**.

methods, including IN2N. Also, our ConsistDreamer is able to assign fine-grained marble texture in the marble statue editing, a clear mustache in Mustache editing, and clear wrinkle and hair in Einstein editing, while IN2N produces blurred and over-smooth results with poor details. Notably, our ConsistDreamer minimizes the side effects of the editing, while IN2N unexpectedly and significantly changes the skin color in the Mustache editing and the wall color in the Einstein editing. (2) The Tolkien Elf and Fauvism editing tasks in the Fangzhou scene show that our ConsistDreamer could preserve most diversity from the original [2], due to the use of structured noise sampled for the whole editing. With the structured noise, we can focus on the consistency of generation for the given noise, without suffering from averaging results generated from different noises, which may lose diversity by converging to an average style for all noises. (3) Our ConsistDreamer works well in outdoor scenes, as all the details on the floor, mountain, plants, and camps are preserved in the edited results. (4) In complicated indoor scenes from the ScanNet++ dataset, our ConsistDreamer generates editing results that are easy to recognize as the given style, with fine-grained textures (Van Gogh), regular patterns (Picasso), or special lighting conditions (Bastion and Transistor). All these results validate that our ConsistDreamer generates high-quality editing results.

Ablation Study. As shown in Table 1, we conduct the ablation study on four representative tasks (A)-(D), covering instructions of object-specific editing, artistic style transfer, and other style transfer, and scenes of human, indoor, and outdoor scenes. The results show that our full ConsistDreamer outperforms all the variants with significant gains in all tasks under DFS, which mainly comes from our consistent denoising procedure in Sec. 3.3 that requires all three major components to achieve. Training towards a consistent denoising procedure produces considerable extra supervision signals to the augmented [2], making it converge better towards consistent generation results. We can also observe that the consistency-enforcing training and the use of surrounding views improve the fidelity in most of the tasks, especially in the complicated large-scale indoor scenes (C)(D), showing that these components indeed improve the consistency in generation.

5. Conclusion

This paper proposes ConsistDreamer, an instruction-guided scene editing framework that generates 3D consistently edited images from 2D diffusion models. Empirical evaluation shows that ConsistDreamer produces editing results of significantly higher quality, exhibiting sharper, brighter appearance with fine-grained textures, across various scenes including forward-facing human scenes, outdoor scenes, and even large-scale indoor scenes in ScanNet++, where it succeeds in common failure cases of previous methods. We hope that our work can serve as a source of inspiration for distillation-based 3D/4D editing and generation tasks.

Acknowledgement. Jun-Kun and Yu-Xiong were supported in part by NSF Grant 2106825 and NIFA Award 2020-67021-32799, using NVIDIA GPUs at NCSA Delta through allocations CIS220014 and CIS230012 from the ACCESS program.

References

[1] 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 *ICCV*, 2021. 3

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Learning to follow image editing instructions. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 3

[4] Jun-Kun Chen, Jipeng Lyu, and Yu-Xiong Wang. NeuralEditor: Editing neural radiance fields via manipulating point clouds. In *CVPR*, 2023. 3

[5] 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. 7

[6] Jiahua Dong and Yu-Xiong Wang. ViCA-NeRF: View-consistency-aware 3D editing of neural radiance fields. In *NeurIPS*, 2023. 1, 3, 7

[7] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. NeRFReN: Neural radiance fields with reflections. In *CVPR*, 2022. 3

[8] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3D scenes with instructions. In *ICCV*, 2023. 1, 3, 4, 5, 7

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 7

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 5

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 6

[13] Hiromichi Kamata, Yuiko Sakuma, Akio Hayakawa, Masato Ishii, and Takuya Narihira. Instruct 3D-to-3D: Text instruction guided 3D-to-3D conversion. *arXiv preprint arXiv:2303.15780*, 2023. 3, 7

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 2

[15] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn2, and Jinwoo Shin1. Collaborative score distillation for consistent visual editing. In *NeurIPS*, 2023. 1, 3, 7

[16] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for editing via feature field distillation. In *NeurIPS*, 2022. 3

[17] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling, 2023. 7

[18] 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 *CVPR*, 2023. 3

[19] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 3

[20] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Junyan Zhu, and Bryan C. Russell. Editing conditional radiance fields. In *ICCV*, 2021. 3

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 6

[22] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. DiffRF: Rendering-guided 3D radiance field diffusion. In *CVPR*, 2023. 2, 4

[23] 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. 5

[24] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. CageNeRF: Cage-based neural radiance field for generalized 3D deformation and animation. In *NeurIPS*, 2022. 3

[25] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 1, 3, 4, 7

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[27] Elvis Rojas, Albert Njoroge Kahira, Esteban Meneses, Leonardo Bautista-Gomez, and Rosa M. Badia. A study of checkpointing in large scale training of deep neural networks. *arXiv preprint arXiv:2012.00825*, 2020. 6

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[29] 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 *CVPR*, 2023. 3

[30] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 7

[31] Liangchen Song, Liangliang Cao, Jiatao Gu, Yifan Jiang, Junsong Yuan, and Hao Tang. Efficient-NeRF2NeRF: Streamlining text-driven 3d editing with multiview correspondence-enhanced diffusion models, 2023. 7

[32] 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 *SIGGRAPH*, 2023. 7

[33] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 3

[34] 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*, pages 1–15, 2023. 3, 6, 7

[35] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *CVPR*, 2021. 3

[36] 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. In *NeurIPS*, 2023. 3

[37] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3D-aware image generation using 2D diffusion models. In *ICCV*, 2023. 2, 3, 5, 7

[38] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *CVPR*, 2021. 3

[39] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *ECCV*, 2022. 3

[40] Xiangzhe Xu, Hongyu Liu, Guanhong Tao, Zhou Xuan, and Xiangyu Zhang. Checkpointing and deterministic training for deep learning. In *2022 IEEE/ACM 1st International Conference on AI Engineering – Software Engineering for AI (CAIN)*, 2022. 6

[41] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *ICCV*, 2021. 3

[42] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. ConsistNet: Enforcing 3D consistency for multi-view images diffusion. *arXiv preprint arXiv:2310.10343*, 2023. 2, 3, 5, 7

[43] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023. 1, 2, 5, 6

[44] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

[45] Lu Yu, Wei Xiang, and Kang Han. Edit-DiffNeRF: Editing 3D neural radiance fields using 2D diffusion model. *arXiv preprint arXiv:2306.09551*, 2023. 3, 7

[46] 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 *CVPR*, 2022. 3

[47] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 5

[48] Joseph Zhu and Peiye Zhuang. HiFA: High-fidelity text-to-3D with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023. 3, 7

[49] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. DreamEditor: Text-driven 3D scene editing with neural fields. In *SIGGRAPH Asia*, 2023. 3, 7