

Inpaint4DNeRF: Promptable Spatio-Temporal NeRF Inpainting with Generative Diffusion Models

Han Jiang^{*}
HKUST

hjiangav@connect.ust.hk

Chi-Keung Tang
HKUST

cktang@cs.ust.hk

Haosen Sun^{*}
HKUST

hsunas@connect.ust.hk

RuoXuan Li^{*}
HKUST

rliba@connect.ust.hk

Yu-Wing Tai
Dartmouth College

yu-wing.tai@dartmouth.edu

Abstract

Current Neural Radiance Fields (NeRF) can generate photorealistic novel views. For editing 3D scenes represented by NeRF, with the advent of generative models, this paper proposes Inpaint4DNeRF to capitalize on state-of-the-art stable diffusion models (e.g., ControlNet [39]) for **direct** generation of the underlying completed background content, regardless of static or dynamic. The key advantages of this generative approach for NeRF inpainting are twofold. First, after rough mask propagation, to complete or fill in previously occluded content, we can individually generate a small subset of completed images with plausible content, called seed images, from which simple 3D geometry proxies can be derived. Second and the remaining problem is thus 3D multiview consistency among all completed images, now guided by the seed images and their 3D proxies. Without other bells and whistles, our generative Inpaint4DNeRF baseline framework is general which can be readily extended to 4D dynamic NeRFs, where temporal consistency can be naturally handled in a similar way as our multiview consistency.

1. Introduction

Recent development of Neural Radiance Fields (NeRF) [18] and its dynamic variants including [20, 25, 28] have shown their great potential in modeling scenes in 3D and 4D. This representation is suitable for scene editing from straightforward user inputs such as text prompts, eliminating the need for modeling and animating in detail. One important task in scene editing is *generative inpainting*, which refers to generating plausible content that is consistent with the background

^{*}Equal contribution. The order of authorship was determined alphabetically.

scene. Generative inpainting has a wide range of potential applications, including digital art creation, and VR/AR.

Although several recent works have addressed the inpainting and text-guided content generation problem on NeRFs, they have various limitations in directly generating consistent novel content seamlessly with the background based on the text input. Specifically, [4, 24, 41] allow users to edit the appearance of an existing object in the NeRF based on the text prompt, but their generations are limited by the original target object, thus they cannot handle substantial geometry changes. In [19, 31] a given NeRF is inpainted by removing the target object and inferring the background, but their inpainting task is not generative as they do not match the inferred background with any other user input. Recent generative works [12, 21, 26, 32] produce new 3D or 4D content from the text input, but the generation is not conditioned on the existing background. While [27] can generate contents around the target object without removing it, generative inpainting on NeRF should allow users to remove target objects while filling the exposed region with plausible 3D content, which was previously totally or partially occluded in the given scene.

While applying recent diffusion models [22] on generative 2D inpainting may largely solve the problem, simply extending diffusion models to higher dimensions to static or dynamic NeRFs introduces extensive challenges, including new and more complex network structures and sufficient higher-dimensional training data. On the other hand, NeRFs are continuous multiview estimations on its training images which provide a natural connection between 2D images and NeRF, enabling us to propagate the 2D inpainting results by diffusion models to the underlying scene.

Thus, in this paper, we propose Inpaint4DNeRF, the first work on text-guided generative NeRF inpainting with diffusion models, which can be naturally extended to inpaint 4D dynamic NeRFs. Given a text prompt, and a target fore-

ground region specified by the user by text or on the single individual image(s), our method first utilizes stable diffusion to inpaint a few independent training views. Next, we regard these inpainted views as seed images and infer a coarse geometry proxy from them. With the strong guidance by seed images and their geometry proxies, the other views being inpainted are constrained to be consistent with the seed images when refined with stable diffusion, which is followed by NeRF finetuning with progressive updates on the training views to obtain final multiview convergence. For dynamic cases, once generative inpainting is achieved on a static single frame, it can be regarded as the seed frame where the edited information can be propagated to other frames to naturally inpaint the scene in 4D without bells and whistles. Our generative Inpaint4DNeRF proposes a text-guided approach for inpainting the relevant background after removing static foreground objects specified by a user-supplied prompt, where multiview consistency is meticulously maintained.

To summarize, Inpaint4DNeRF presents the following contributions. First, harnessing the power of recent diffusion models on image inpainting, we can directly generate text-guided content with new geometry while being consistent with the context given by the unmasked background. Second, our approach infers and refines other views from initially inpainted seed images to achieve multiview consistency across all the given views, where temporal consistency is naturally enforced and achieved.

2. Related Work

2.1. NeRF Editing

NeRF Editing, as pioneered by [5, 14, 38], capitalizes on the capabilities of Neural Radiance Fields (NeRFs) to enable intricate manipulations and edits within 3D scenes and objects, including object removal, geometry transformations, and appearance editing. Enabling editing within the implicit density and radiance field has enormous promise across many fields, including virtual reality, augmented reality, content development, and beyond. Nevertheless, these prior works either edit the geometry of already existing contents in NeRF, or edit only the appearance of the scene. None of these works targets at *generating* novel contents consistent with the underlying NeRF given by the input images to be inpainted. For this task, [41] generates new texture on an existing object, but does not generate new geometry; [27] generates promptable geometrical content around a target object, but the generation is limited near the object and cannot replace the existing object. In our work, we focus on a subtask of NeRF editing, *generative inpainting* in NeRF.

2.2. Inpainting Techniques

The volume-renderable representation and the training mode of NeRF introduce a natural bridge between the represented

3D or 4D scene and its multiview observations. This enables image editing techniques to be applied on NeRF inpainting.

Recent advancements in 2D image inpainting have primarily relied on generative models [29], particularly Generative Adversarial Networks (GANs) [36, 37] and Stable Diffusion (SD) models [17, 23]. These models have demonstrated remarkable capabilities in generating visually plausible predictions for missing pixels. Specifically, the capacity to simulate complex data distributions has drawn notable attention to Stable Diffusion [23], an extension of the denoising diffusion probabilistic model that supports inpainting guided by text prompts. This method produces inpainted pixels and samples of superior quality by using controlled diffusion processes. Diffusion models and noise-conditioned score networks are vital to stable diffusion as they interact to optimize the generation process. Researchers have leveraged Stable Diffusion inpainting for various applications, including semantic inpainting, texture synthesis, and realistic object removal. These applications involve understanding contextual information in the image and generating inpaintings that seamlessly integrate with the existing scene.

However, most of these works have been limited to synthesis in the image domain and have yet to explicitly consider maintaining fidelity to the 3D structure across various viewpoints. Several prior works [19, 31] addressed the pure NeRF inpainting problem, which removes target objects and infers the background without generating prompt-guided contents. Notably, our study not only goes beyond traditional 2D image inpainting, addressing the challenge of inpainting within 3D and 4D scenes while maintaining view consistency even under perspective changes, but is also promptable in the sense that the inpainted content is matched with the text description while consistent with the underlying 3D or 4D spatiotemporal scene to be inpainted.

2.3. Text-Guided Visual Content Generation

The advent of generative models has led to extensive research on guiding the generation results using natural language. For example, the latent diffusion model, as exemplified by [23], has made significant strides in text-guided image generation. Various image modification techniques, such as [7, 9, 10, 33], have emerged as a result of these improvements.

Based on the above text-to-image achievements, text-to-3D generation has been introduced, as shown in [2, 12, 21, 32]. These approaches aim to bridge the gap in 3D content generation, leveraging the Score Distillation Sampling (SDS) technique and its variants for multiview convergence. Moreover, attempts have been made to generate 4D dynamic content from text [26], with several techniques including a temporal consistency regularizer to extend DreamFusion [21] to dynamic NeRFs. Despite the complexity of SDS sampling, they have achieved impressive results in terms of 3D consistency. On the other hand, our proposed method conditions

on the *seed* view to control the generation of other views to force multiview convergence. This approach restricts the ill-posed text-guided generation problem to a well-posed problem with strong priors, thus making the problem easier to tackle. 3D generation conditioned on one generated view has been presented in some most recent works, including Zero123 [13] and SyncDreamer [16]. Given an image of an object and multiple camera poses, they can infer feasible observation of the object from other views.

However, existing implementations are all limited to a single object without conditioning on the background, and struggle with manipulating objects within large scenes. In other words, they all contribute to the pure generation task which is different from our inpainting task. Our method distinguishes itself by enabling the removal, addition, and manipulation of specific objects within a given background NeRF, while maintaining consistency with the unmasked background and partially masked foreground objects. In addition, the 3D inpainted results can be extended to 4D while maintaining temporal consistency.

3. Method

Given a pre-trained NeRF, a set of masks on its training images denoting the target object to be replaced (or removed), and a text prompt, we propose *generative promptable inpainting*, which can be decomposed into three objectives: 1) 3D and 4D visual content generation, where the resulting fine-tuned NeRF should contain a new object that is multiview and temporally consistent; 2) text-prompt guided generation, where the semantics of the generated object should match the input text prompt; 3) the generated inpainted content should be consistent with the existing NeRF background.

Our proposed framework consists of three main stages, as shown in Figure 1. First, we employ stable diffusion [22] to inpaint one view as the first seed image, and generate a coarse set of seed images conditioned on the first seed image. The other views are then inferred from the seed images and refined by stable diffusion. This stage pre-processes the training images aiming to make convergence easier and faster later. Next, we fine-tune the NeRF by performing a stable diffusion version of iterative dataset update [4] to enforce 3D multiview consistency. A converged 3D NeRF is obtained in this stage. If we target at inpainting 4D NeRF, we propagate the 3D inpainted result along the time dimension in the final stage. In the following, we will describe the three stages in detail, respectively in each subsection.

3.1. Training View Pre-processing

Text-guided visual content generation is inherently a highly underdetermined problem: for a given text prompt, there are infinitely many object appearances that are feasible matches. In our method, we generate content in NeRF by first performing inpainting on its training images and then backward

propagating the modified pixels of the images into NeRF.

If we inpaint each view independently, the inpainted content will not be consistent across multiple views. Some prior 3D generation works, using techniques including score distillation sampling [21], have demonstrated the possibility of multiview convergence based on independently modified training views. Constraining the generation problem by enforcing the training views to be strongly related to each other will simplify convergence to a large extent. Therefore, we first inpaint a small number of *seed images* associated with a coarse set of cameras that covers sufficiently wide viewing angles. For the other views, the inpainted content are strongly conditioned on these seed images.

Progressive seed image generation. Given a set of training viewpoints $\{\phi^i\}$, we manually select a subset $\{\phi_{seed}^i\} \subseteq \{\phi^i\}$ that has comparable view coverage with $\{\phi^i\}$, and perform inpainting on their associated training images $\{C_{seed}^i\}$ to obtain $\{\hat{C}_{seed}^i\}$. As the seed images $\{C_{seed}^i\}$ themselves may not be multiview-consistent, we first choose one view ϕ_{seed}^0 and inpaint the first seed image \hat{C}_{seed}^0 , and regard it as the seed of other additional seed images. This first inpainted image \hat{C}_{seed}^0 is conditioned only on C_{seed}^0 but nothing else. For other seed views ϕ^i where $i > 0$, we use the stable diffusion model to inpaint \hat{C}_{seed}^i , with the previously inpainted \hat{C}_{seed}^{i-1} as a strong prior. Details on how to utilize the previous inpainted image from a different camera as a prior will be introduced later in this subsection.

Generating other views. Once we have inpainted all the seed images $\{\hat{C}_{seed}^i\}$, we use them to interpolate all the other views to get the full set of pre-processed inpainted images $\{\hat{C}^i\}$. For a particular view ϕ^j , we choose its closest n cameras (viewpoints) from $\{\phi^i\}$ to form $\{\phi^1, \phi^2, \dots, \phi^n\}$, associated with a set of weights $\{w^1, w^2, \dots, w^n\}$ where the weights are proportional to the inverse of the camera distances and sum up to 1:

$$w^k = \frac{1}{(\mathbf{x}(\phi^k) - \mathbf{x}(\phi^j)) \cdot \sum_{a=1}^n w^a}$$

where \mathbf{x} denotes the world space location of the viewpoints. These weights approximately reflect how much each neighboring view is related to the target view ϕ^j . According to these weights, we combine the visual content from the neighborhood cameras of ϕ^j and generate an inpainted image for it. We use a projection-correction method to implement the aforementioned process as described below.

Inpainted depth as geometry proxy. To ‘copy’ the inpainted content from ϕ^i to ϕ^j , a simple approach is to cast rays from ϕ^i into 3D space and then project them to ϕ^j , and the rgb values associated with the rays are rasterized onto ϕ^j . For this approach, in addition to the two camera poses, we need a depth map to determine which 3D locations the rays will cast to. The depth map is required to be plausible itself to make the projected content on ϕ^j plausible. Nonetheless,

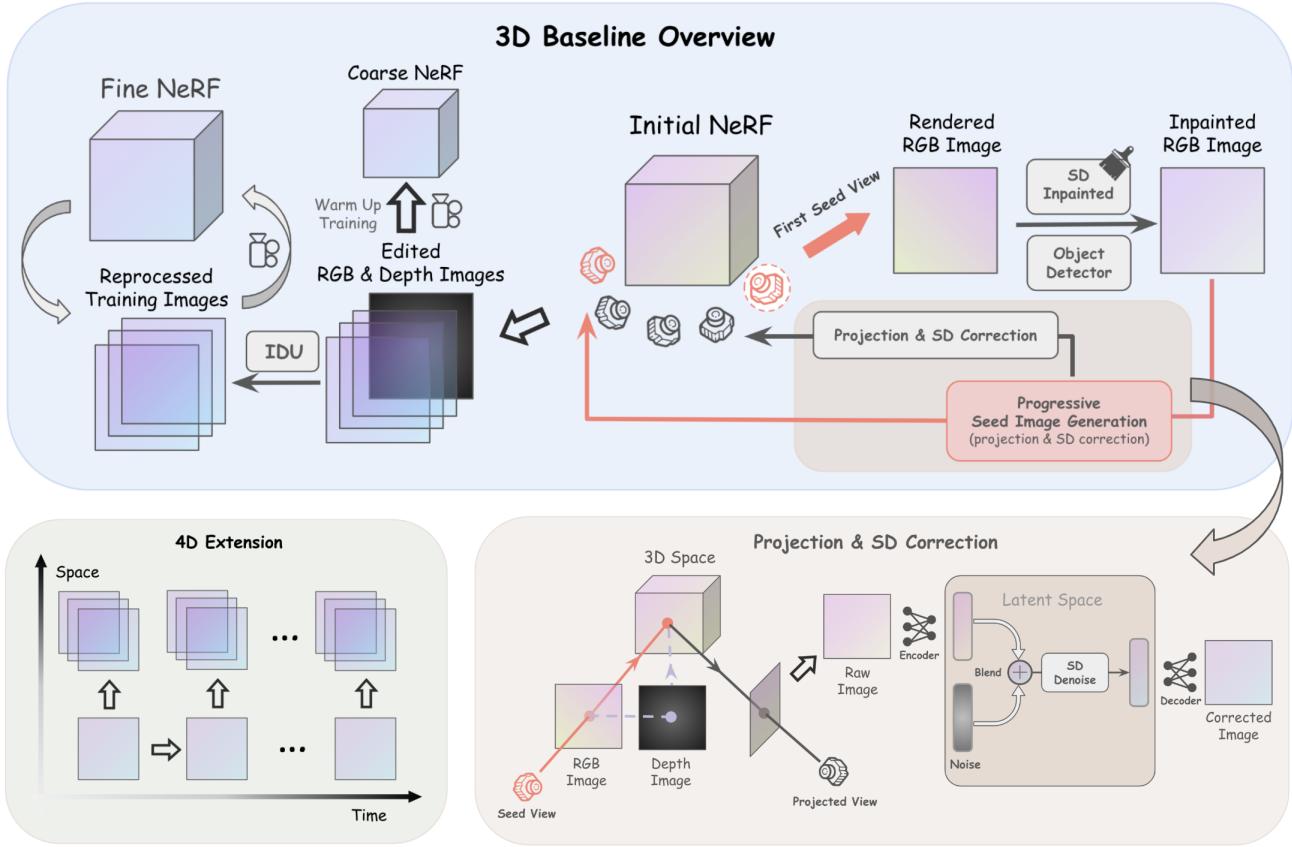


Figure 1. Baseline Overview. Our generative NeRF inpainting is based on the inpainted image of one training view. The other seed images and training images are obtained by using stable diffusion to hallucinate the corrupted details of the unprojected raw image. These images are then used to finetune the NeRF, with warmup training to get geometric and coarse appearance convergence, followed by iterative training image update to get fine convergence. For 4D extension, we first obtain a temporally consistent inpainted seed video based on the first seed image. Then for each frame, we infer inpainted images on other views by projection and correction, as in our 3D baseline.

it does not need to be perfectly accurate, since the purpose of the projection step is to give a rough prior on ϕ^j . Therefore, we assume a planar depth on ϕ^i . Specifically, on the depth map D^i of ϕ^i , we first perform pure inpainting using Lama [29] within the user-provided mask to erase the original depth, so that the depth within the mask is consistent with the background. Then, we feed the user's text prompt into LangSAM [8, 15] to generate an accurate mask of the inpainted object on \hat{C}^i . Within this mask, the depth value is modified to be an appropriate constant for approximately accurate projections. Since we are substituting an object by another inpainted object, the original object's location is a good indicator of where the new object should locate. In this way, the inpainted object on \hat{C}^i is projected onto \hat{C}^j as if its geometry is planar, with the planar artifact to be auto-corrected later. In practice, to avoid projection artifacts on the background, we only keep the projection result within the mask of ϕ^j . Pixels outside the mask will remain the same.

For seed image generation from the first seed image, we

only use 1 view for the projection. For generating non-seed views, we perform the projection n times for the n neighboring cameras respectively, which gives n projected results. Multiplying these results with their associated weights $\{w^1, w^2, \dots, w^n\}$, they are blended together as the final projection result. In practice, directly blending in image space may cause unnatural overlapping of different projected results. To reduce this artifact, we use the VAE of stable diffusion to encode the projections into latent tensors and perform blending in the latent space. Then the blended latent is decoded to be the final projection result. In this way, the stable diffusion correction step (introduced as follows) will have a higher chance to generate good results.

Stable diffusion correction. Since our planar depth estimation is not always accurate, while the raw projected results are plausible in general, the many small artifacts make the relevant images unqualified for training. To make them look more natural, we propose to cover the projection artifacts with stable diffusion hallucinated details. First, we blend the

raw projection with random noise in stable diffusion’s latent space for t timesteps, where t is relatively small to the total number of stable diffusion timesteps. Then, starting from the last t steps, we denoise with stable diffusion to generate the hallucinated image. The resulting image is then regarded as the initial training image. With this step, the training image pre-processing stage is completed.

3.2. Progressive Training

Warmup Training. Our training image pre-processing stage provides a good initialization for rough convergence. Before fine-tuning on these images to get fine convergence, notice that the 3D object in NeRF is still the original object intact, which is quite different from what is depicted in the pre-processed training images above in geometry and appearance. Therefore, for the first stage of NeRF training, without any finetuning, we directly train the NeRF on the pre-processed images for coarse convergence. Note that when we perform fine-tuning later, the original object will have no effect because warmup training has already erased all its appearance information.

Iterative Dataset Update. Given a NeRF in which the editing target has fixed overall geometry, Iterative Dataset Update (IDU), first proposed and shown effective in InstructNerf2Nerf [4], is a useful training image fine-tuning strategy which can edit the appearance and fine geometry of the target object in NeRF. In our task, warmup training has already provided a converged coarse geometry, leaving fine geometry and appearance to be determined. In their work, they use InstructPix2Pix [1] as a diffusion model strongly conditioned on the original training image. In our task, we use stable diffusion in a similar way to achieve similar objectives. The fine-tuned image should be conditioned on the pre-processed image, as well as the depth map. In detail, for pre-processed image conditioning, we adopt the same approach as in the previously introduced projected image correction. The preprocessed image and the current NeRF rendering are blended together, either in image space or in latent space. Then a small amount of noise is injected into the blended image, followed by stable diffusion denoising from an intermediate timestep. For depth map conditioning, we input the depth map of the current NeRF into ControlNet to guide the editing process.

Along with training image updates and NeRF training, the underlying NeRF will gradually converge to an object with fine details, making rendering from NeRF more reliable. Meanwhile, stable diffusion correction with less noise injected into the training images introduces less multiview inconsistency to fine details. Therefore, we gradually increase the intermediate timestep of our stable diffusion correction when training NeRF, until the intermediate timestep becomes very close to the final denoise timestep. Consequently, the coarse NeRF converges to a fine one.

Regularizers. To supervise NeRF training, we use the L1 photometric loss between NeRF rendering and the training images. However, sole supervision of RGB cannot avoid inaccuracies and noise in geometry. To achieve clean converged geometry, we add the following two regularizers into supervision. First, we also render a depth value along a ray, and compute the depth loss as defined in [3] between the rendered depth and our pre-processed 2-layer depth i.e., initial inpainted geometry proxies. Supervision with our planar 2-layer depth is not accurate, but it helps to avoid incorrect converged geometry, such as merging of the inpainted geometry with the background. Second, to deal with noisy RGB and depth renderings and “floaters” (artifacts hovering in the underlying NeRF volume), we use LPIPS loss [40] on RGB as a regularizing term. In practice, we found LPIPS to be effective in reducing noise and floaters. Since LPIPS is based on patches, rays selected in NeRF training are not allowed to be random, but are required to form rectangular patches. Thus we render small image patches, corresponding to rectangular areas in ground truth images, during NeRF optimization. The effect of the regularizers is also previously demonstrated in [31, 34].

3.3. 4D Extension

Our method extends naturally to 4D by adopting the idea in 3D, that is, to have a raw inference from the seed image, and correct its details with stable diffusion. To infer across frames, we apply multiple existing video editing methods to the input video. We use point-tracking to record the original movement. This movement will be applied to animate the novel object generated through our progressive training. Meanwhile, the foreground of the original video will be removed to adjust the new scene. The video with only the background will be combined with the animated novel object to build up a raw inpainted video that is temporally consistent. For each frame, we extract the seed image from this seed video and project it to other views. Finally, training images from all views and frames are refined by stable diffusion before being used in training.

To determine the motion for the new object, since there exists no stable method for generating a background-consistent movement for a generated foreground object, we utilize the original movement of the replaced object. To achieve this goal, it is required to first extract the original movement, and then apply it back to the novel object. We use CoTracker [6] to track the movements of multiple key points on the target. CoTracker enables us to track self-defined key points on the original video precisely. Making use of this functionality, the trajectory followed by the key points can be obtained and further projected to the generated object with consistent background information. In this way, we are able to animate the generated object.

4. Experiments

For our experiments, we select from dynamic scenes in the Nvidia Dynamic Scenes Dataset [35]. Scenes in this dataset are captured using a sparse set of 12 stationary cameras located in two rows, producing images of resolution 1015×1920 . The static scenes we use are taken from one frame from the dynamic scenes. For the backbone NeRF, we use static and dynamic versions of K-Planes [25] implemented in nerfstudio [30]. For each scene, we conduct inpainting by replacing a foreground object with another text-prompted object with a different geometry. We will demonstrate the effectiveness of our method by showing the qualitative intermediate and final results. In addition, we will explain different parts of our design by ablations and comparisons on our baseline.

4.1. Qualitative results

3D Examples. We show several 3D inpainting examples in figure 2. For each individual inpainting task, we show 2 renderings of the final NeRF from different views to demonstrate the multiview consistency. Additionally, we show the first seed image, another pre-processed image, as well as the RGB and depth map in the three stages: before training, after warmup training, and after convergence. These before-and-after images demonstrate the efficacy of each stage in our method. As shown in Figure 2, a roughly consistent pre-processed image can optimize a coarse inpainted NeRF after warmup training, and the geometry (represented by depth map) converges during warmup training. Then, fine convergence across views is achieved after the final training stage. All 3D inpainting tasks are trained on a single Nvidia RTX 4090 GPU. Warmup training takes approximately 0.5–1 hour, and the main training stage with IDU takes approximately 1–2 hours.

4D Example. We show a 4D inpainting example in figure 3 to demonstrate that our method has the potential to generalize to dynamic NeRFs. In this example, we remove the foreground object in the video of the seed view using E2FGVI [11], a flow-based method with optimization by feature propagation and content hallucination. For transferring motion to the generated object, after key point tracking, we estimate a rigid transformation between the key points, and propagate the pixels along the transformation. This dynamic scene consists of 16 frames, in which the first frame includes the first seed image. As shown in the figures, we successfully obtained an overall convergence on the generated object with correct motion for all the illustrated frames.

4.2. Ablation and comparison

To demonstrate the efficacy of various elements in our baseline design, we compare our baseline to the following 3 variants. As we are the first, to the best of our knowledge,

to aim at generative NeRF inpainting, there does not exist baselines for direct comparison. Therefore, we conduct ablation by modifying some parts in our baseline, and conduct comparisons by replacing a subprocess by an existing baseline.

View independent inpainting. In our baseline, all training images are pre-processed to be consistent with the first seed image, providing a guarantee for coarse convergence. To show the necessity of our pre-processing strategy, in this baseline, we inpaint all training images independently. Warmup training and IDU are kept unchanged. The results are shown in figure 4. It is clear that independent inpainting results in divergence across views, and IDU fails to produce the initial highly inconsistent image convergence. This explains that the pre-processing step to help convergence is indispensable.

Instruct-Nerf2Nerf after warmup. The second stage of our NeRF training is largely based on iterative dataset update, which is the core of instruct-nerf2nerf [4]. This brings out the question of whether instruct-nerf2nerf can accomplish our task. Since instruct-nerf2nerf is mainly for appearance editing without large geometry changes, it is inappropriate to directly run it on our generative inpainting task. Therefore, before running their baseline, we first run our baseline until the end of warmup training, so that we have a coarse geometry as the prerequisite. By this design, the main difference between the two methods is that instruct-nerf2nerf uses instruct-pix2pix, a non-latent diffusion model, to do training image correction. Figure 5 shows its results compared with ours. We can see that the final result maintains the blurriness from the warmup NeRF, and does not achieve a higher level of consistency.

Warmup without depth supervision. The goal of warmup training is to provide an initial convergence, especially in geometry, which requires depth loss as supervision. To demonstrate this, we provide an example of warmup training without any depth supervision, and show the rendered depth map in figure 6. Compared with the depth map in our standard setting, this depth map has clear errors within and around the target object. From inconsistencies in geometry, the later fine training stage may suffer from “floater” and inconsistency artifacts. This can be eliminated by our simple planar depth map as guidance, even if it is not accurate.

5. Conclusion

We introduce Inpaint4DNeRF, a unified framework that can directly generate text-guided, background-appropriate, and multi-view consistent content within an existing NeRF. To ensure convergence from the original object to a completely different object, we propose a training image pre-processing method that projects from initially inpainted seed images to other views, with details refined by stable diffusion. A

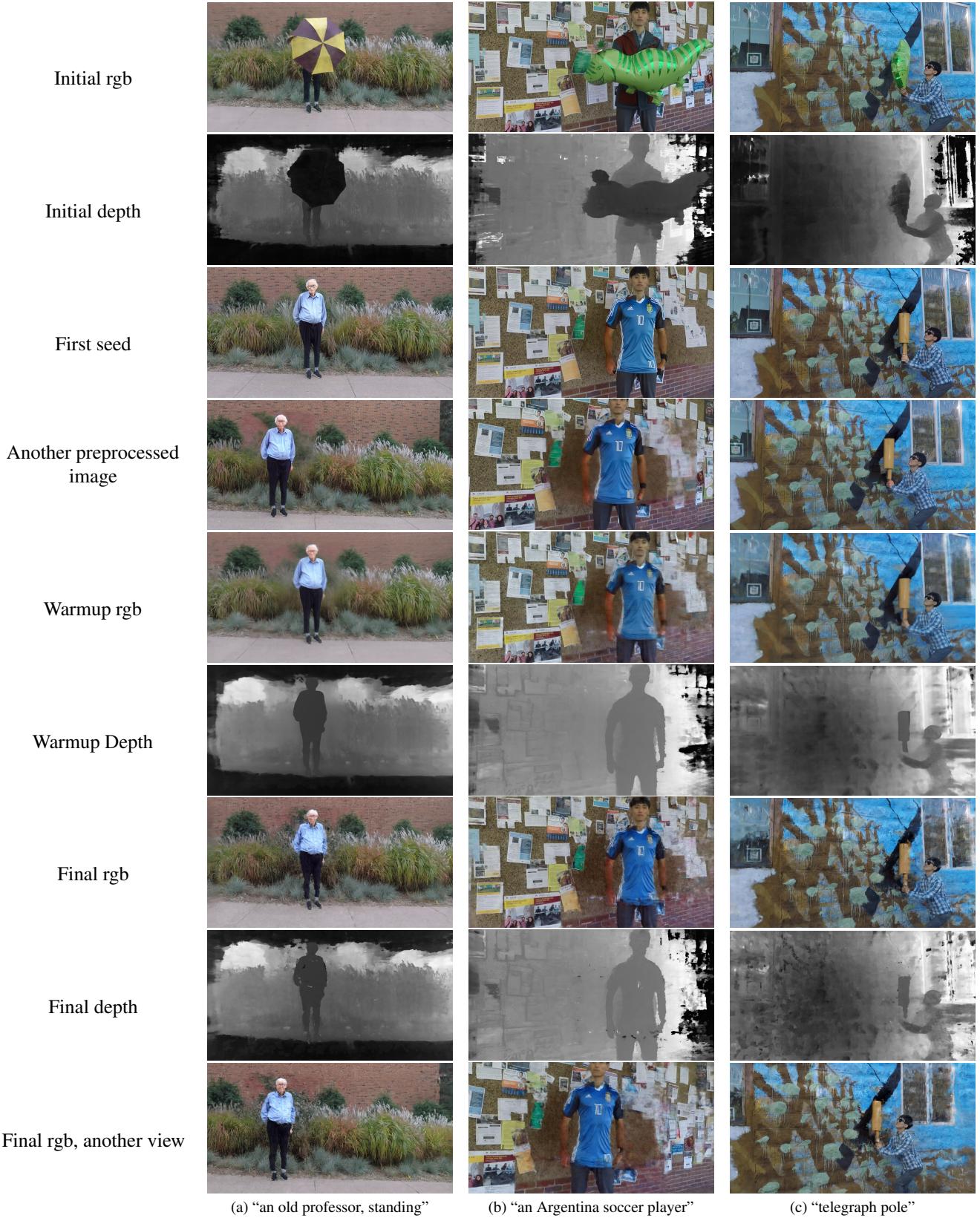


Figure 2. Our qualitative results in 3D. Each column illustrates an inpainting example. We show final renderings from 2 views to demonstrate the multiview consistency. We also show depth maps and rgb images of different training stages to show their roles.

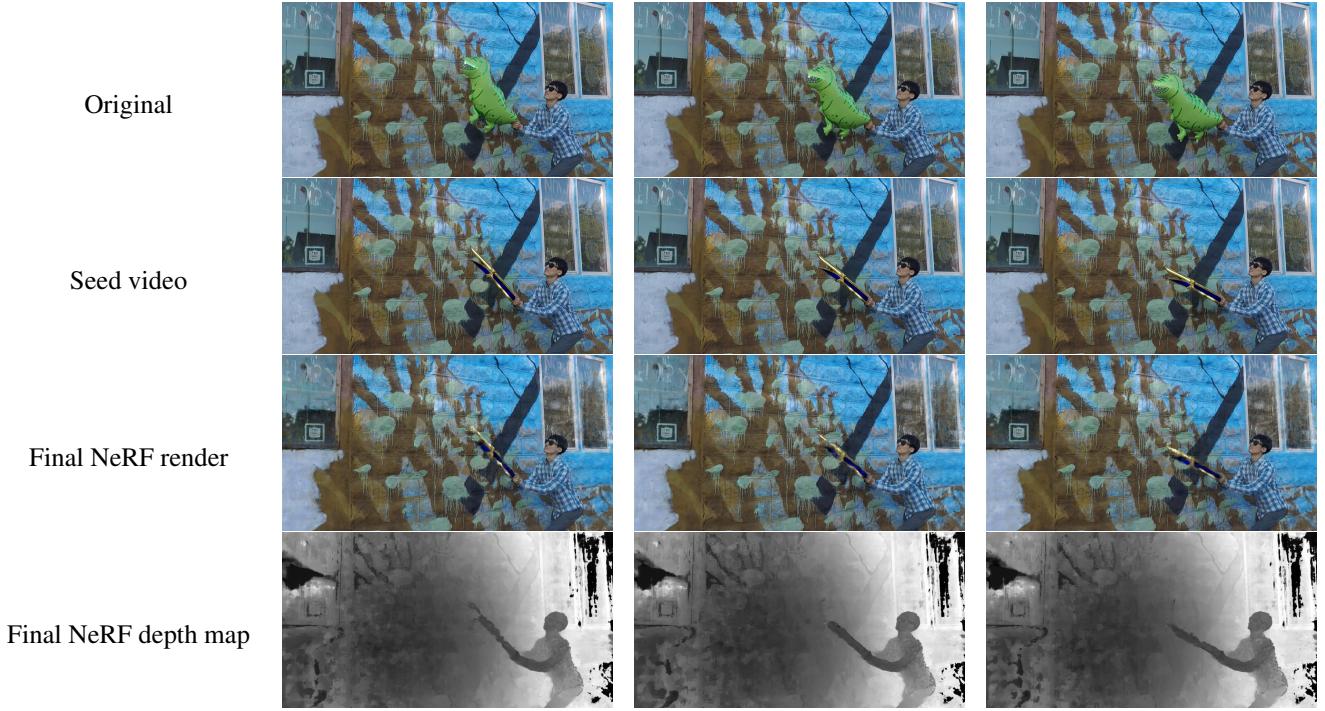


Figure 3. 4D NeRF Inpainting example. Text prompt: “a golden sword, side view”. The first column corresponds to the first frame which includes the first seed image, and the other columns correspond to 2 later frames. Inpaint4DNeRF can generate a moving object that is overall consistent.



Figure 4. Training results with view independent inpainting. Left: rgb render. Right: noisy and incorrect depth map.

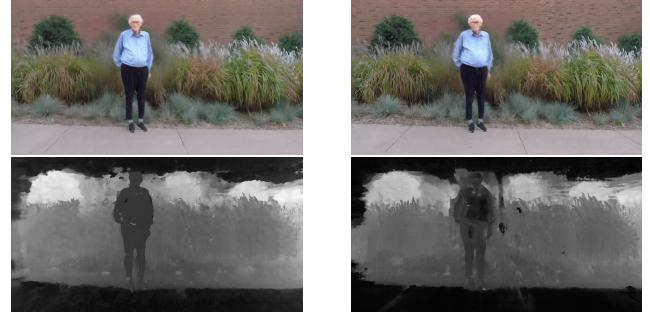


Figure 6. Depth maps and rgb renderings from warmup training with and without depth supervision. Left: with depth supervision (ours). Right: without depth supervision.



Figure 5. Training results with instruct-nerf2nerf. Left: Result from our baseline. Right: result from warmup training followed by instruct-nerf2nerf.

roughly multiview consistent set of training images, combined with depth regularization, guarantees coarse convergence on geometry and appearance. Finally, the coarse NeRF is fine-tuned by iterative dataset update with stable diffusion. Our baseline can be readily extended to dynamic NeRF inpainting by generalizing the seed-image-to-other strategy from the spatial domain to the temporal domain. We provide

3D and 4D examples to demonstrate the effectiveness of our method. We also investigate the role of various elements in our baseline by ablation and comparison.

The proposed framework expands the possibilities for realistic and coherent scene editing in 3D and 4D settings. However, our current baseline still has some limitations, providing room for further improvement. Specifically, it is challenging for our method to handle complex geometry generation with a camera set covering wide angles. The consistency of the final NeRF can still be improved. In addition, to extend our method fully into 4D, certain techniques are

required to further improve temporal consistency and maintain better multiview consistency across frames. We hope that our proposed baseline can inspire these future research directions for text-guided generative NeRF inpainting.

References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. [5](#)
- [2] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [2](#)
- [3] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [5](#)
- [4] 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. [1, 3, 5, 6](#)
- [5] Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. Editable free-viewpoint video using a layered neural representation. In *ACM SIGGRAPH*, 2021. [2](#)
- [6] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. [5](#)
- [7] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation, 2022. [2](#)
- [8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. [4](#)
- [9] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. Manigan: Text-guided image manipulation, 2020. [2](#)
- [10] Bowen Li, Xiaojuan Qi, Philip Torr, and Thomas Lukasiewicz. Lightweight generative adversarial networks for text-guided image manipulation. *Advances in Neural Information Processing Systems*, 33:22020–22031, 2020. [2](#)
- [11] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17562–17571, 2022. [6](#)
- [12] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [1, 2](#)
- [13] 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, 2023. [3](#)
- [14] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [15] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. [4](#)
- [16] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. [3](#)
- [17] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. pages 11451–11461, 2022. [2](#)
- [18] 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. [1](#)
- [19] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023. [1, 2](#)
- [20] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), 2021. [1](#)
- [21] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. [1, 2, 3](#)
- [22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. [1, 3](#)
- [23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. [2](#)
- [24] Cheng Peng Zerong Zheng Boyao Zhou Hongwen Zhang Yebin Liu. Ruizhi Shao, Jingxiang Sun. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor, 2023. [1](#)
- [25] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. [1, 6](#)
- [26] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. *arXiv:2301.11280*, 2023. [1, 2](#)
- [27] Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. Blending-nerf: Text-driven localized editing in neural radiance fields, 2023. [1, 2](#)

- [28] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 1
- [29] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lemitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 2, 4
- [30] Matthew Tancik, Ethan Weber, Evonne Ng, Rui long Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 6
- [31] 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. 1, 2, 5
- [32] 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. 1, 2
- [33] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2256–2265, 2021. 2
- [34] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9416–9426, 2020. 5
- [35] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera, 2020. 6
- [36] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2
- [37] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018. 2
- [38] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2
- [39] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1
- [40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5
- [41] Shangzhan Zhang, Sida Peng, Yinji ShenTu, Qing Shuai, Tianrun Chen, Kaicheng Yu, Hujun Bao, and Xiaowei Zhou. Dyn-e: Local appearance editing of dynamic neural radiance fields, 2023. 1, 2