This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

CTRL-D: Controllable Dynamic 3D Scene Editing with Personalized 2D Diffusion

Kai He Chin-Hsuan Wu Igor Gilitschenski
University of Toronto Vector Institute
{hekai, chwu, gilitschenski}@cs.toronto.edu



Figure 1. We present CTRL-D, a dynamic 3D scene editing framework that enables controllable, high-quality, consistent scene edits by editing only a single image using any 2D editing approach. Our framework is also compatible with both monocular and multi-camera scenes. Please see our project page for more dynamic visualizations: *https://IHe-KaiI.github.io/CTRL-D/*.

Abstract

Achieving controllable and consistent editing in dynamic 3D scenes remains a significant challenge. Previous work is largely constrained by its editing backbones, resulting in inconsistent edits and limited controllability. We propose to address this challenge using personalized diffusion models. In our work, we introduce a novel framework that first fine-tunes the InstructPix2Pix model, followed by a two-stage optimization of the scene based on deformable 3D Gaussians. The fine-tuning enables the model to "learn" the editing ability from a single edited reference image. This reduces the more complex task of dynamic scene editing to the more simple 2D image editing problem. By directly learning editing regions and styles from the reference, our approach enables consistent and precise local edits without the need for tracking desired editing regions, effectively addressing key challenges in dynamic scene editing. Then, our two-stage optimization progressively edits the trained dynamic scene, using a designed edited image buffer to accelerate convergence and improve temporal consistency. Compared to state-of-the-art methods, our approach offers more flexible and controllable local scene editing, achieving high-quality and consistent results.

1. Introduction

The advent of efficient 3D representations, such as Neural Radiance Fields (NeRF) [40] and 3D Gaussian Splatting (3DGS) [27], has revolutionized the real-world scene capture, enabling photorealistic novel-view synthesis. With a growing interest in dynamic scenes, several works [5, 59, 65, 70, 71] have extended NeRF and 3DGS to this setting, significantly broadening their scope of applications. Building on these developments, there is an increasing demand for editing existing scenes—a capability essential for various applications such as virtual and augmented reality (VR/AR), content creation, and simulation. While recent advances in static 3D scene editing have achieved promising results, the controllable and consistent editing of dynamic 3D scenes remains underexplored.

Recent progress [43, 57] in dynamic 3D scene editing has leveraged pre-trained diffusion-based image editing methods, such as InstructPix2Pix (IP2P) [4] and ControlNet [72], for iteratively editing the scene. However, these approaches often face limitations due to the constrained editing capacity of their backbones, such as the inability to perform precise local editing or restrictions imposed by the training domain of the pre-trained diffusion models. This leads to inconsistent editing results across frames or limited controllability. Overcoming these limitations for dynamic scene editing is an open research challenge. Inspired by recent strides in 2D editing [4, 39, 72], it has become increasingly robust and versatile. We investigate how a single edited image can serve as a reference for dynamic scene editing. This approach offers greater flexibility and controllability. It allows users to employ any 2D editing tool to modify the image, and then guides edits in dynamic scenes.

In this paper, we propose CTRL-D, a novel pipeline that first fine-tunes the IP2P model using only one edited image, enabling personalized dynamic scene editing with expanded editing capabilities. This personalization enables IP2P to "learn" the editing ability from the given reference image. Controllable local editing in dynamic scenes is challenging, as tracking the desired editing region is more difficult than in static 3D scenes. Previous work [42] relies on noise differences to determine editing regions. In contrast, our personalized IP2P variant directly learns these editing regions from the reference image. Furthermore, we incorporate a prior preservation loss inspired by [54]. It enables maintaining paired-edit consistency and stability.

After fine-tuning, the personalized IP2P can be deployed for dynamic scene editing tasks. We then introduce a two-stage optimization framework based on deformable 3D Gaussians [65, 70], designed to improve efficiency in dynamic scene editing. In the first stage, only the canonical space of the scene is optimized, with Gaussian densification performed based on a single keyframe. In the subsequent stage, the parameters of the deformation field and 3D Gaussians are optimized. We use a specially designed edited image buffer to accelerate convergence. This framework significantly improves efficiency and enhances temporal consistency, outperforming existing state-of-the-art methods for dynamic scene editing [43].

In summary, our contributions are threefold:

- We introduce a novel scene editing framework that fine-tunes IP2P using an edited reference image, expanding editing capabilities while preserving the consistency of the original model's editing style.
- We present a two-stage dynamic scene optimization based on deformable 3D Gaussians, enabling efficient and consistent dynamic scene editing.
- Our approach transforms the complex dynamic scene editing task into a simple 2D image editing, making dynamic scene editing more accessible and controllable, enabling consistent local editing, and showing superior results than previous work.

2. Related Work

Diffusion Models for 2D Editing. Diffusion models [24, 58, 60] have emerged as the leading image synthesis approach owing to their remarkable ability to generate high-quality images. By training these models on large-scale datasets, many works have demonstrated impressive results in Text-to-Image (T2I) generation [44, 52, 53, 55]. Building on this success, pre-trained T2I diffusion models have also been adapted for image editing via text prompts [2, 12, 15, 23, 26, 37, 39, 54]. Notably, SDEdit [39] modifies the input image by adding and removing noise based on the target prompt. Textual Inversion [15] and DreamBooth [54] enable personalized generation by learning the concept of a specific object from a small collection of images. Imagic [26] refines a single image by inverting and fine-tuning it, then regenerates the image with a new text prompt. Another line of work performs local inpainting, guided by a caption and an additional region constraint, which can be either model-generated [12, 23] or user-defined [2, 37]. InstructPix2Pix (IP2P) [4] further leverages Prompt-to-Prompt [23] to collect image editing examples for training. This achieves superior quality and speed compared to previous methods. ControlNet [72], by contrast, is not specifically focused on editing but rather extends T2I models with multiple input conditions, effectively enhancing editing capabilities. Recently, video editing has become an area of increased focus. Early works, like Tune-A-Video [66] and Video-P2P [35], enable localized edits by fine-tuning T2I models to capture continuous motion. While many follow-up methods [7, 10, 20, 31, 73] achieve plausible results, they still face challenges in view and temporal consistency, and quality generation.

NeRF and 3DGS Editing. Neural Radiance Field (NeRF) [40] and 3D Gaussian Splatting (3DGS) [27] have revolutionized 3D modeling with their high-fidelity rendering quality. This success has naturally led to the development of tools for editing both NeRF and 3DGS. Beginning with EditNeRF [34], which optimizes latent codes to refine both the appearance and shape of objects, subsequent works [3, 18, 63] have expanded editing capabilities by incorporating text prompts, guided by CLIP [51] for optimization. Other approaches enable localized editing by applying inpainting techniques from 2D manipulation to gather target datasets and update the learned fields [29, 41]. Additionally, some methods utilize predefined template models (e.g., meshes [11, 49, 68], skeletons [45], or point clouds [8]) to articulate and deform shapes within the radiance field.

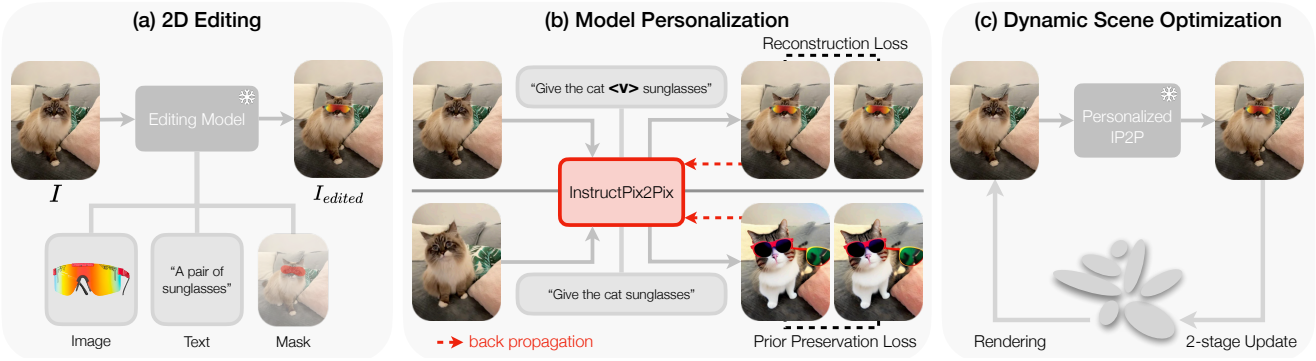


Figure 2. Our pipeline for controllable dynamic scene editing. Given a dynamic 3D scene, our method (a) first edits one frame as a reference with any 2D editing model, we then (b) fine-tune the InstructPix2Pix [4] with the edited reference image, along with sampled images from the original models to preserve the model priors, and then (c) we optimize the dynamic 3D scenes with deformable gaussian representation, using the designed 2-stage method.

Recently, with the rapid advancement of T2I diffusion and video generation models, leveraging these priors for 3D [14, 21, 22, 42, 56, 74] and 4D [43, 57] editing has gained substantial attention. While Instruct 4D-to-4D [43], Control4D [57] share our objective of achieving effective dynamic editing, they often produce inconsistent edits across frames and offer limited control over the scene.

Dynamic View Synthesis. Building on the success of NeRF, numerous works have extended it to model dynamic scenes. Some approaches [16, 17, 32, 62, 67] incorporate time as an additional dimension in the radiance field to capture scene deformations. However, these methods often struggle to model complex geometry, leading to unrealistic, inconsistent structures in occluded regions. Another class of methods [46, 47, 50, 61] introduces a deformation field to disentangle time from the radiance field, where deformations, conditioned on time [50] or a temporal latent code [46, 47, 61], are parameterized as rigid body motion. While effective for longer videos, these approaches generally perform best on object-centric scenes with minimal motion or simple camera paths. Recently, the introduction of 3DGS has advanced this field by providing a lightweight, explicit deformation model to handle Gaussian motions and shape changes [38, 70, 71], Despite the promising quality of dynamic GS models, methods for editing such representations are still largely unexplored.

3. Method

Our method takes as input a trained dynamic 3D Gaussian scene, along with the original source data: a set of multi-view images and their registered camera poses. The framework consists of three key modules: First, we select a single image from the input set and apply any off-the-shelf image editing technique to produce an edited version of this image (Sec. 3.4). Using the pair of source and edited images, we then fine-tune the InstructPix2Pix (IP2P) model with a specifically designed text prompt (Sec. 3.2). Finally, we introduce a two-stage optimization process for the dynamic 3D Gaussian model, which iteratively refines the dynamic scene using the personalized image editor (Sec. 3.3). An overview of the pipeline is shown in Fig. 2.

3.1. Background

Dynamic 3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [27] utilizes a set of 3D Gaussians $G = (\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{s}, \sigma)$ defined by a center position $\boldsymbol{x} \in \mathbb{R}^3$, opacity $\sigma \in \mathbb{R}_{\geq 0}$, and 3D covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ obtained from unit quaternion $\boldsymbol{r} \in \mathbb{H}$ and scaling $\boldsymbol{s}$. The color of the pixel on the image plane, denoted by $\mathbf{p}$, is rendered using a differentiable splatting rendering process as follows:

$$C(\mathbf{p}) = \sum_{i\in N} \alpha_i c_i \prod_{j=1}^{i-1}(1-\alpha_j), \qquad \alpha_i = \sigma_i \mathrm{e}^{-\frac{1}{2}(\mathbf{p}-\boldsymbol{\mu}_i)^T \Sigma'(\mathbf{p}-\boldsymbol{\mu}_i)}, \tag{1}$$

where c_i represents the color of each Gaussian along the ray, Σ' is the corresponding 2D covariance matrix, and $\boldsymbol{\mu}_i$ represents the uv coordinates of the 3D Gaussians projected onto the 2D image plane.

To model dynamic scenes, some works [65, 70] extend 3DGS with a deformation field, enabling the learning of 3D Gaussians in canonical space, and subsequently puts the deformed 3D Gaussians $G = (\boldsymbol{x}+\delta\boldsymbol{x}, \boldsymbol{r}+\delta\boldsymbol{x}, \boldsymbol{s}+\delta\boldsymbol{x}, \sigma)$ into the rasterization pipeline. Given the time t and center position $\boldsymbol{x}$ of 3D Gaussians as inputs, the deformation MLP predicts offsets:

$$(\delta\boldsymbol{x}, \delta\boldsymbol{r}, \delta\boldsymbol{s}) = \mathcal{F}_\theta(\gamma(\mathrm{sg}(\boldsymbol{x})), \gamma(t)), \tag{2}$$

where $\mathrm{sg}(\cdot)$ indicates a stop-gradient operation, and γ represents the positional encoding.

InstructPix2Pix. InstructPix2Pix (IP2P) [4] is a diffusion-based method for 2D image editing. Given an image I and a text instruction C_T, IP2P aims to generate an edited image z_0 based on I following the instruction guidance of C_T. The diffusion model predicts noise in the input image (or pure noise) z_t at timestep t, using a denoising U-Net ϵ_θ. IP2P is trained on a dataset in which, for each pair I and C_T, an example edited image I_{edited} is provided. The method builds on latent diffusion [53], incorporating a variational autoencoder [28] (VAE) with an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$. During training, Gaussian noise $\epsilon \sim \mathcal{N}(0,1)$ is added to $z = \mathcal{E}(I_{\text{edited}})$ to obtain a noisy latent representation z_t, with $t \in T$ as a randomly selected timestep. The denoiser ϵ_θ is initialized with weights from stable diffusion [53] and fine-tuned to minimize the diffusion objective:

$$\mathbb{E}_{I_{\text{edited}},I,C_T,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t,t,I,C_T)\|_2^2\right]. \tag{3}$$

To achieve classifier-free guidance for both image conditioning and text conditioning, IP2P randomly sets $I = \varnothing_I$ for 5% of examples, $C_T = \varnothing_T$ for 5% of examples, and both $I = \varnothing_I$ and $C_T = \varnothing_T$ for 5% of examples during training. Thus, we can control the strength of the edit by the image guidance scale s_I and the text guidance scale s_T. The modified score estimate is then computed as

$$\begin{aligned}\tilde{e_\theta}\left(z_t,t,I,C_T\right) &= e_\theta\left(z_t,t,\varnothing_I,\varnothing_T\right)\\ &+ s_I\left(e_\theta\left(z_t,t,I,\varnothing_T\right) - e_\theta\left(z_t,t,\varnothing_I,\varnothing_T\right)\right)\\ &+ s_T\left(e_\theta\left(z_t,t,I,C_T\right) - e_\theta\left(z_t,t,I,\varnothing_T\right)\right).\end{aligned} \tag{4}$$

3.2. Personalization of InstructPix2Pix

Assuming we have obtained the edited image I_{edited} and the source image I from the previous stage, we proceed with fine-tuning the U-Net ϵ_θ of InstructPix2Pix (IP2P) to create a personalized editing model. This fine-tuning process allows the model to specialize in our desired edits while preserving its prior capabilities. To achieve this, we require a text instruction that accurately describes the editing action. Initially, we manually generate or utilize GPT-4V [1] to generate a preliminary text instruction $C_T^\star$ to describe the edit. To enhance specificity, we follow the approach in [74], incorporating a specialized token <V> before the last noun or adjective describing the editing target. This process results in a specific text prompt C_T, tailored to describe our editing process.

To retain the generalization capability of IP2P, we integrate a prior preservation loss inspired by DreamBooth [54]. This approach helps maintain the model's adaptability while focusing on specific edits. The input for prior preservation training includes the initial text instruction $C_T^\star$, a randomly selected frame I_d from the dynamic scene dataset, and a corresponding edited image $I^\star_{\text{edited}}$ generated by the original IP2P with text condition $C_T^\star$ and image condition I_d. This configuration enables the model to learn new edits while preserving its prior knowledge.

The fine-tuning loss function for our method consists of two terms. The primary term minimizes the error between the predicted noise and the actual noise in the target-edited image, while the prior preservation term ensures that the model does not lose generalization over the original editing capabilities. The overall finetuning loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{finetune}} = &\mathbb{E}_{I_{\text{edited}},I,C_T,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t,t,I,C_T)\|_2^2\right]\\ &+ \lambda\mathbb{E}_{I^\star_{\text{edited}},I_d,C_T^\star,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t^\star,t,I_d,C_T^\star)\|_2^2\right],\end{aligned} \tag{5}$$

where z_t denotes the noisy latent representation at timestep t extracted from the edited input images, and $z_t^\star$ denotes the corresponding variables used in prior preservation training. λ is a hyperparameter that controls the relative weight of the prior-preservation term.

Data augmentation. In DreamBooth, fine-tuning typically requires 3–5 images, which allows for more robust model adaptation. However, in our setting, obtaining multiple paired images that achieve the same editing effect is challenging and often impractical. With only a single image pair available for fine-tuning, one-shot fine-tuning can cause overfitting and potential model collapse. Although various methods [13, 36] have been proposed to address these issues, they are not suitable for our editing framework. To overcome this limitation, we employ a straightforward and effective data augmentation technique. By applying affine transformations—including rotation, translation, and shear—to both the source and edited images, we find that it significantly reduces overfitting and enhances model stability during fine-tuning.

3.3. Optimization of Dynamic 3D Gaussians

In our pipeline, we propose a two-stage optimization approach for dynamic 3D Gaussians to progressively edit the dynamic scene. First, we use the edited image from the single image editing stage as a keyframe to optimize the canonical space of the scene, while simultaneously performing Gaussian densification. Next, we optimize both the deformation field and the 3D Gaussians using a carefully designed edited image buffer, which allows for efficient and consistent editing results across frames.

Stage 1: Keyframe-guided Gaussian densification. With the edited 2D image as a reference keyframe, we begin by optimizing the canonical space of the dynamic scene. In this initial stage, we refine only the 3D Gaussians, keeping the deformation field frozen. Unlike NeRF-based methods, 3DGS-based methods use point-based rendering. This discrete scene representation can lead to empty areas in the geometry of our edited regions, where Gaussians are sparse

or absent, limiting the scene representation capability. The original 3DGS introduces adaptive control to densify the Gaussians for regions under reconstruction, while also subdividing areas where Gaussians are large and exhibit significant overlap. In our pipeline, we employ a similar adaptive scheme, using positional gradients to identify areas for densification. By optimizing the 3D Gaussians with only the edited keyframe and increasing density based on gradient statistics, we enhance the 3D Gaussians' capacity to express complex appearances and geometries for editing. This process also accelerates the overall optimization, providing a coarse scene for the next stage.

Stage 2: Optimization on both deformation fields and 3D Gaussians. In this stage, we fix the number of Gaussians and optimize both the deformation field and the 3D Gaussians. Following the idea of Instruct-NeRF2NeRF [21], we apply a similar process with the Iterative Dataset Update (Iterative DU) which iteratively replaces the dataset image with its edited counterpart. Benefiting from the relative consistency of our editing results across frames, we introduce an edited image buffer to accelerate the editing process. In each iteration, we randomly select a frame that will not be edited and then use our personalized IP2P to generate the corresponding edited image, which is then added to the edited image buffer. Meanwhile, the 3D Gaussians and deformation field are trained solely on the images within this edited image buffer. We refer to this as the warm-up phase. Once every image has been edited at least once, resulting in an edited image buffer that contains all images, our strategy aligns with the original Iterative DU approach. Initially, only the keyframe is included in the edited image buffer.

After the warm-up phase, the preliminary results demonstrate reasonable temporal consistency. To further enhance this temporal consistency, we introduce a temporal loss $\mathcal{L}_{\text{temp}}$ inspired by [64], which minimizes the differences between adjacent frames. The overall loss function is then expressed as:

$$\mathcal{L} = (1 - \lambda_d)\mathcal{L}_1 + \lambda_d \mathcal{L}_{\text{D-SSIM}} + \lambda_t \mathcal{L}_{\text{temp}}, \quad (6)$$

where the first two terms are adapted from 3DGS, and λ_t represents the weight associated with our temporal regularization term.

3.4. Implementation Details

Single image editing. For text-driven editing, we use a text prompt along with an optional mask to guide the edit. We use SDEdit [39] or InstructPix2Pix [4] to perform 2D editing. To constrain edits to specific regions, we incorporate BlendedDiffusion [2], enabling precise control within masked areas when a mask is provided. In image-driven editing, we utilize a reference image along with a mask to perform editing. Following the approach outlined in MimicBrush [9], we apply reference imitation to edit the image, ensuring that the edited output closely aligns with the content-specific attributes of the reference image. For stylization, we utilize NNST [30] to transfer style from a given reference style image.

Scene Optimization. We edit the image from the dataset for every 50 iterations. For the monocular scene, we adopt [70] to model the scene and perform stage 1 for the first 300 iterations; for the multi-camera scene, we utilize [65] to model the scene and perform stage 1 for the first 100 iterations. We set $\lambda_d = 0.2$ and $\lambda_t = 0.001$. Further details will be discussed in the appendix.

4. Experiments

Dataset. We evaluate our methods using several dynamic scenes from DyCheck [17], which consist of monocular and object-centric scenes, and N3DV [33], which includes dynamic indoor scenes with a multi-camera setup. To more effectively evaluate the local editing capability of our methods, we further use the same approach as DyCheck to capture human-centric dynamic scenes, similar to data shown in Instruct-NeRF2NeRF [21], including face-forward portraits and full-body scenes.

4.1. Qualitative Evaluation

Our qualitative results are presented in Fig. 3, where we evaluate our method on both monocular and multi-camera scenes. We apply various 2D editing techniques to the reference image, which is displayed in the bottom-right or bottom-left of each example. The results include text-prompt editing, image-prompt editing, and style transfer, all demonstrating high fidelity, quality, and controllability. Our method achieves strong temporal consistency across all scenes and multi-view consistency in multi-camera setups. Additionally, our method effectively supports local editing, preserving unrelated regions without modification.

Comparison with dynamic scene editing methods. Two recent works, Instruct 4D-to-4D (IN4D) [43] and Control4D [57], focus on text-driven dynamic scene editing. Since Control4D is primarily tailored for portrait editing and has not released its code, we use Instruct 4D-to-4D as the baseline. Although our method supports both text-driven and image-driven editing with various single-image editing techniques, we limit the single-image editing approach to IP2P [4] with a mask to ensure fairness. The qualitative results are presented in Fig. 4. As shown, our method enables precise local editing, avoiding modifications to unrelated regions. For instance, in the first row, while IN4D alters the background and sofa colors, our approach modifies only the cat's color, as intended. In the second and third

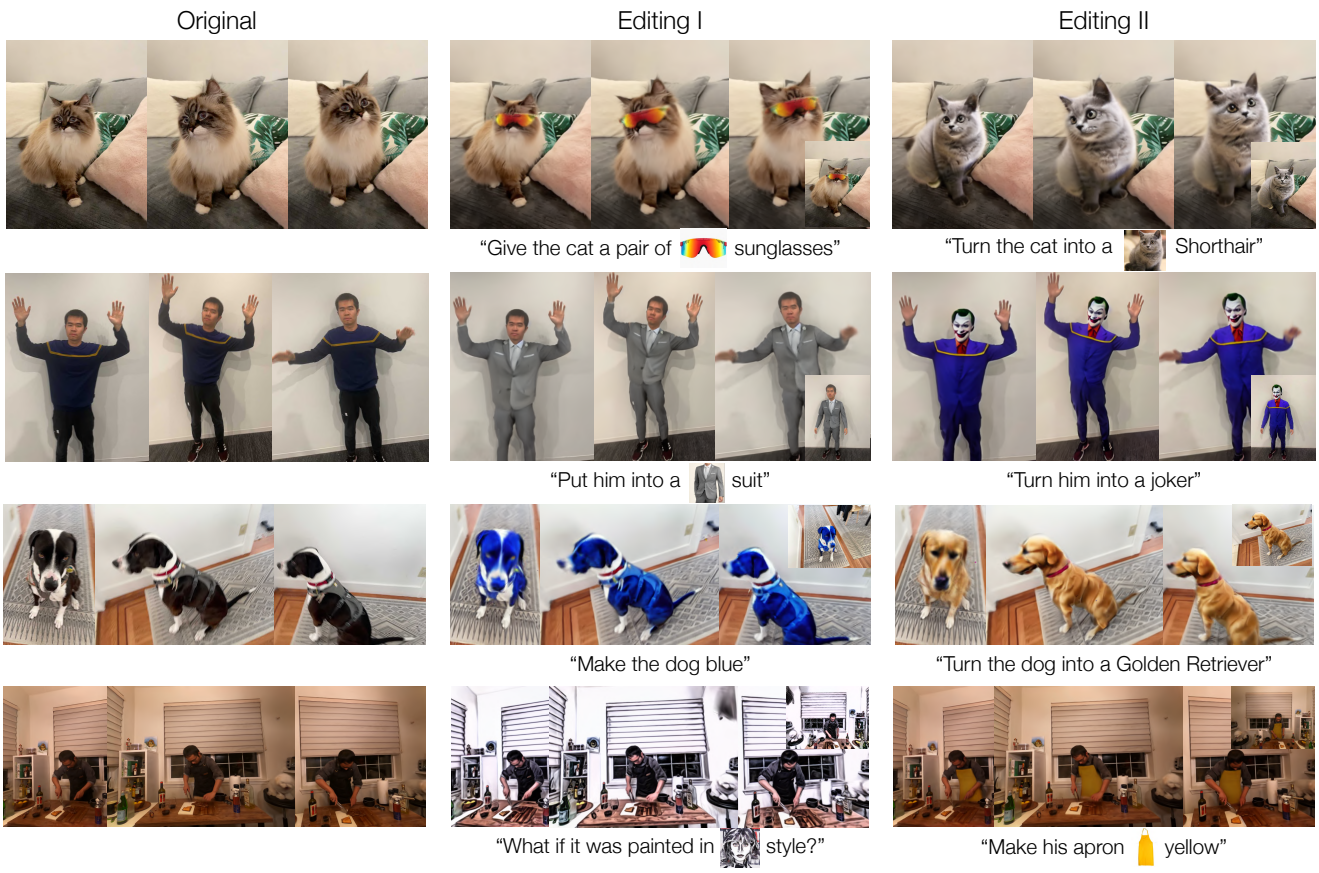


Figure 3. Qualitative results on both monocular and multi-camera scenes. For each scene, we show two edited versions based on the original, using various 2D editing techniques to demonstrate the high fidelity, quality, and controllability of our method. The reference 2D image for each edit appears in the bottom-right or bottom-left corner.

rows, our method demonstrates superior temporal consistency across all frames, as well as enhanced quality. In the last row, IN4D affects the T-shirt color, but our method restricts changes to the face, matching the reference edited image. Additionally, our approach significantly enhances temporal consistency; in the last row, IN4D produces inconsistent hair colors across frames, while our results maintain color consistency. Lastly, our close-up images clearly show that our method achieves higher quality and fewer artifacts.

Comparison with video editing methods. We further compare our method with existing video editing approaches on monocular scenes. We use AnyV2V [31] as our baseline, editing only the first frame and then propagating changes to the remaining frames, which is similar to ours. As shown in Fig. 5, we edit the first frame (leftmost) and generate our results directly using the personalized IP2P model, without dynamic scene optimization. Our method produces significantly higher quality and more consistent results, whereas AnyV2V outputs are blurry and display inconsistencies in the face and suit across frames. Furthermore, as the input video length increases, AnyV2V becomes increasingly unstable and produces lower-quality results. In contrast, our method edits frame-by-frame, handling inputs of any length. Additionally, our method supports both monocular and multi-camera scenes, while AnyV2V struggles to maintain consistency across multiple camera views.

Scene	Method	CLIP Score↑	Consistency↑	Time↓
Portrait	Ours	**27.75**	**0.953**	**60 mins**
	IN4D [43]	27.38	0.933	2 hours
Cat	Ours	**31.81**	**0.968**	**60 mins**
	IN4D	31.72	0.964	2 hours
Steak	Ours	**28.52**	**0.988**	**40 mins**
	IN4D	28.23	0.983	2 hours

Table 1. Quantitative comparison with Instruct 4D-to-4D (IN4D) on CLIP score, subject consistency score in video, and approximate running time. The results demonstrate that our method outperforms IN4D in better alignment with text prompts, superior temporal consistency, and faster runtime.

4.2. Quantitative Evaluation

We present a quantitative comparison with Instruct 4D-to-4D (IN4D) [43] in Tab. 1 on monocular scenes (Portrait,

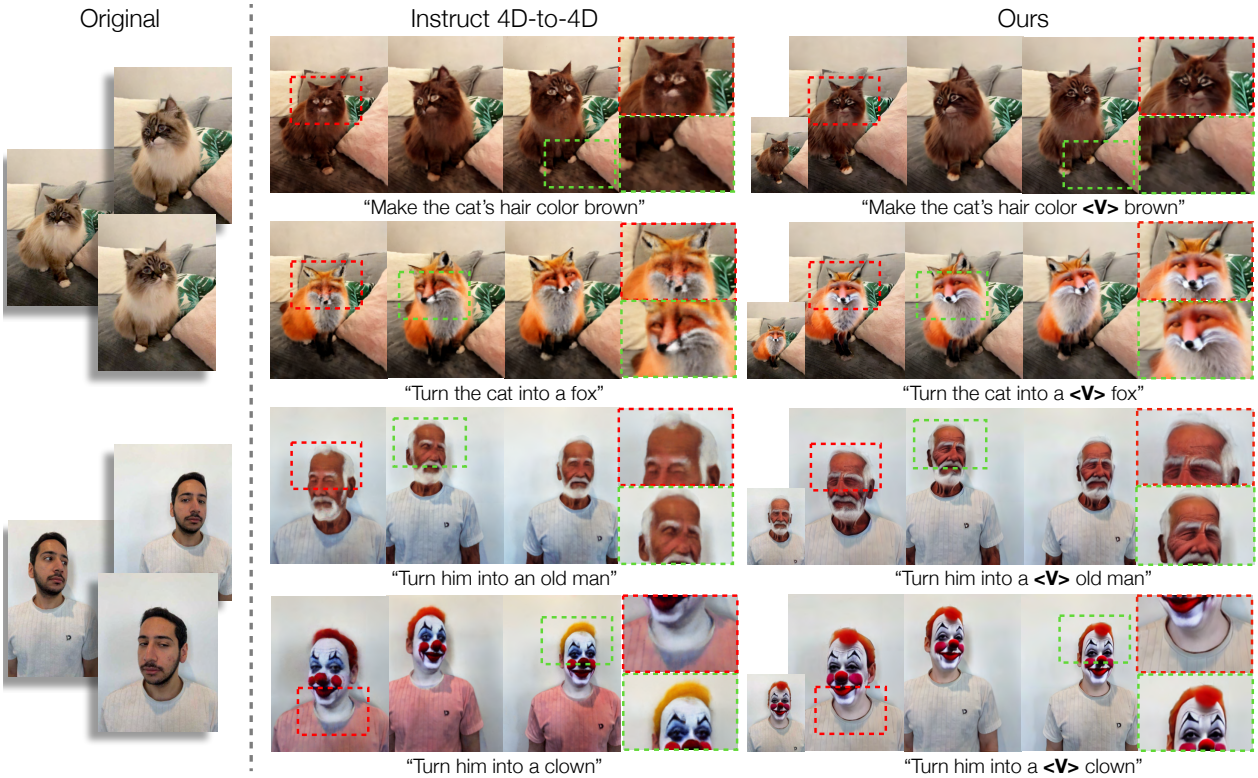


Figure 4. Qualitative comparison with Instruct 4D-to-4D (IN4D) [43] on text-driven scene editing. The reference 2D images used in our method are shown at the bottom-left of each edited scene. We also provide zoomed-in details of the edited scene on the right side. Our method demonstrates superior consistency, higher quality, and more precise local edits, which are not achievable with IN4D.



Figure 5. Qualitative comparison with AnyV2V [31] on monocular scenes. The leftmost image is the edited first frame. The prompt for our fine-tuned IP2P is "Put him in a <V> suit". Our results demonstrate higher quality and greater consistency.

Cat) and a multi-camera scene (Steak). We utilize the CLIP score to measure the alignment of the editing results with text instructions. To evaluate temporal consistency, we use a subject consistency score (denoted as **Consistency**) from VBench [25], which is designed to evaluate subject consistency in video generative models. Specifically, the consistency score is computed by measuring the DINO [6] feature similarity across frames. Our results achieve higher CLIP and consistency scores, indicating better alignment with the text prompts and superior temporal consistency. Additionally, we compare the running time of each method. The total time required for our method, including fine-tuning the IP2P model and optimizing dynamic scenes, is less than half that of IN4D. More details are discussed in the appendix.

4.3. Ablation Study

Data augmentation. We evaluate the impact of data augmentation in IP2P fine-tuning in the upper part of Fig. 6. First, we fine-tune IP2P without any data augmentation (denoted as **w/o aug**) and proceed with dynamic scene optimization. The results are blurry, and it fails to produce smooth and consistent editing across frames due to inconsistencies in the frames generated by the personalized IP2P. In contrast, fine-tuning IP2P with data augmentation (denoted as **w/ aug**) yields significantly higher quality and more consistent results.

Edited image buffer. We evaluate the effectiveness of our designed edited image buffer for optimizing dynamic 3D



Figure 6. Ablation study. The results demonstrate the effectiveness of our data augmentation and improved efficiency of using our edited image buffer.



Figure 7. Example of editing ability generalization. We demonstrate that by fine-tuning a personalized IP2P model using images of a cat, it can be successfully applied to other domains, such as portrait and full-body scenes.

Gaussians. For comparison, we introduce a variant that performs optimization by randomly selecting a frame from all frames to optimize the Gaussians (denoted as **w/o buffer**). In contrast, our method specifically selects frames from the edited image buffer (denoted as **w/ buffer**). We run both methods for 1000 optimization iterations. As shown in the lower part of Fig. 6, our method successfully produces the edited results, whereas the variant generates results nearly identical to the original scene, highlighting the importance of the edited image buffer.

4.4. Editing Ability Generalization

Since our personalization resembles IP2P "learning" a new editing ability based on the reference image, we validate its ability to generalize this ability to data outside the reference domain. The results are shown in Fig. 7. In this example, we begin by applying image-driven editing to a 2D image to add a specific pair of sunglasses to a cat. This edited image is then used as a reference to fine-tune IP2P, and we subsequently apply the personalized model to a portrait scene and a full-body scene. The results show that our method successfully generalizes the learned editing ability to other data domains, demonstrating the robust generalization capabilities of our personalized IP2P.

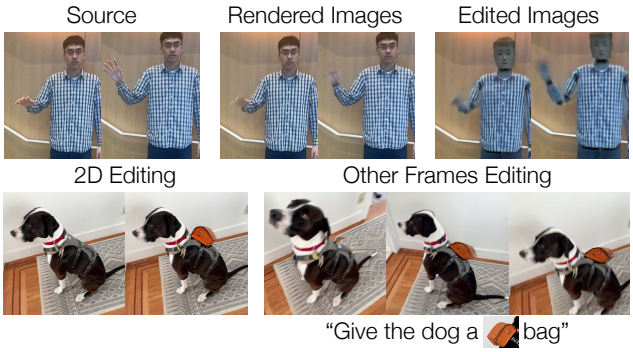


Figure 8. Examples of failure cases. The reconstruction backbone fails to produce clear rendering results, leading to blurry edited images (top). Our method exhibits limitations inherited from IP2P, such as adding a specific bag within empty regions (bottom).

4.5. Limitation

While our method produces high-quality editing results, it inherits certain limitations from both our reconstruction backbone and IP2P. As shown in the first row of Fig. 8, when the input trained dynamic 3D Gaussians scene fails to render clear images, especially the moving hands, our framework produces edits based on these unclear images, which results in blurry and unclear outcomes. Moreover, while our fine-tuning significantly alleviates the original IP2P's difficulty with controllable 2D edits, our personalized IP2P still generates 3D-inconsistent images when editing in empty regions. In the second row of Fig. 8, we give the dog a specific bag, but the model struggles to consistently render the bag from multiple views. This issue likely arises from IP2P's difficulty in adding complex content to empty regions. Future improvements to our method could benefit from a more robust reconstruction backbone and a base diffusion model with enhanced capabilities for adding content. Further limitations are discussed in the appendix.

5. Conclusion

In this paper, we propose CTRL-D, a novel framework for dynamic 3D scene editing by simplifying the task into 2D editing through fine-tuning InstructPix2Pix with a single pair of images, combined with an efficient two-stage optimization of deformable 3D Gaussians. Our approach is compatible with any 2D editing tool, enabling controllable and flexible local editing in dynamic scenes. Qualitative results demonstrate high-quality, high-fidelity, and consistent edits, while our comparisons show that our method significantly outperforms baselines, underscoring its effectiveness. We envision our framework advancing controllable scene editing and inspiring future exploration in this field.

Acknowledgments. The authors thank Shuhong Zheng, Mohammad Mohammadi, Yifei Sun, and Fengyuan Liu for their help in the collection of data for this project.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022.

[3] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *CVPR*, 2023.

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[5] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *CVPR*, 2023.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[7] Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023.

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

[9] Xi Chen, Yutong Feng, Mengting Chen, Yiyang Wang, Shilong Zhang, Yu Liu, Yujun Shen, and Hengshuang Zhao. Zero-shot image editing with reference imitation. *arXiv preprint arXiv:2406.07547*, 2024.

[10] Xuweiyi Chen, Tian Xia, and Sihan Xu. Unictrl: Improving the spatiotemporal consistency of text-to-video diffusion models via training-free unified attention control. *arXiv preprint arXiv:2403.02332*, 2024.

[11] Chong Bao and Bangbang Yang, Zeng Junyi, Bao Hujun, Zhang Yinda, Cui Zhaopeng, and Zhang Guofeng. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *ECCV*, 2022.

[12] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[13] Ziyi Dong, Pengxu Wei, and Liang Lin. Dreamartist: Towards controllable one-shot text-to-image generation via positive-negative prompt-tuning. *arXiv preprint arXiv:2211.11337*, 2022.

[14] Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024.

[15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

[16] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021.

[17] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022.

[18] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM ToG*, 2023.

[19] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. Videoswap: Customized video subject swapping with interactive semantic point correspondence. In *CVPR*, 2024.

[21] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, 2023.

[22] Runze He, Shaofei Huang, Xuecheng Nie, Tianrui Hui, Luoqi Liu, Jiao Dai, Jizhong Han, Guanbin Li, and Si Liu. Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training. In *CVPR*, 2024.

[23] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023.

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

[25] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[26] 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 *CVPR*, 2023.

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM ToG*, 2023.

[28] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[30] Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. Neural neighbor style transfer. *arXiv preprint arXiv:2203.13215*, 2022.

[31] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

[32] Tianye Li, Miroslava Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, S

Lovegrove, Michael Goesele, R Newcombe, et al. Neural 3d video synthesis from multi-view video. In *CVPR*, 2021.

[33] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *CVPR*, 2022.

[34] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *ICCV*, 2021.

[35] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *CVPR*, 2024.

[36] Jianxiang Lu, Cong Xie, and Hui Guo. Object-driven one-shot fine-tuning of text-to-image diffusion with prototypical embedding. *arXiv preprint arXiv:2401.15708*, 2024.

[37] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

[38] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024.

[39] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.

[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. In *ECCV*, 2020.

[41] 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.

[42] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Watch your steps: Local image and scene editing by text instructions. In *ECCV*, 2024.

[43] Linzhan Mou, Jun-Kun Chen, and Yu-Xiong Wang. Instruct 4d-to-4d: Editing 4d scenes as pseudo-3d scenes using 2d diffusion. In *CVPR*, 2024.

[44] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

[45] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021.

[46] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.

[47] 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. In *ACM ToG*, 2021.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[49] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.

[50] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2020.

[51] 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 *ICML*, 2021.

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

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

[54] 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.

[55] 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. In *NeurIPS*, 2022.

[56] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *ICCV*, 2023.

[57] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Efficient 4d portrait editing with text. In *CVPR*, 2024.

[58] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[59] 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. *TVCG*, 2023.

[60] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

[61] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021.

[62] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021.

[63] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *CVPR*, 2022.

[64] Penghao Wang, Zhirui Zhang, Liao Wang, Kaixin Yao, Siyuan Xie, Jingyi Yu, Minye Wu, and Lan Xu. Vˆ 3: Viewing volumetric videos on mobiles via streamable 2d dynamic gaussians. *arXiv preprint arXiv:2409.13648*, 2024.

[65] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024.

[66] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, 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. In *ICCV*, 2023.

[67] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021.

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

[69] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, 2023.

[70] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024.

[71] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *ICLR*, 2024.

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

[73] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. In *ICLR*, 2024.

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