Instruct 4D-to-4D: Editing 4D Scenes as Pseudo-3D Scenes Using 2D Diffusion

Linzhan Mou[1†] Jun-Kun Chen[2†] Yu-Xiong Wang[2]
[1]Zhejiang University [2]University of Illinois Urbana-Champaign [†]Equal Contribution
moulz@zju.edu.cn {junkun3,yxw}@illinois.edu

Figure 1. **Our Instruct 4D-to-4D edits 4D scenes as pseudo-3D scenes with 2D diffusion**, achieving much sharper results with detailed textures across a variety of editing tasks and scenes. Notably, Instruct 4D-to-4D generates realistic and 4D consistent editing results in both monocular scenes and challenging multi-camera indoor scenes. **Please refer to the supplementary video for additional visualization**.

Abstract

This paper proposes Instruct 4D-to-4D that achieves 4D awareness and spatial-temporal consistency for 2D diffusion models to generate high-quality instruction-guided dynamic scene editing results. Traditional applications of 2D diffusion models in dynamic scene editing often result in inconsistency, primarily due to their inherent frame-by-frame editing methodology. Addressing the complexities of extending instruction-guided editing to 4D, our key insight is to treat a 4D scene as a pseudo-3D scene, decoupled into two sub-problems: achieving temporal consistency in video editing and applying these edits to the pseudo-3D scene. Following this, we first enhance the Instruct-Pix2Pix (IP2P) model with an anchor-aware attention module for batch processing and consistent editing. Additionally, we integrate optical flow-guided appearance propagation in a sliding window fashion for more precise frame-to-frame editing and incorporate depth-based projection to manage the extensive data of pseudo-3D scenes, followed by iterative editing to achieve convergence. We extensively evaluate our approach in various scenes and editing instructions, and demonstrate that it achieves spatially and temporally consistent editing results, with significantly enhanced detail and sharpness over the prior art. Notably, Instruct 4D-to-4D is general and applicable to both monocular and challenging multi-camera scenes. Code and more results are available at immortalco.github.io/Instruct-4D-to-4D.

1. Introduction

Being able to synthesize photo-realistic novel-view images through rendering, neural radiance field (NeRF) [19] and its variants have become the leading neural representation for 3D and even 4D dynamic scenes. Moving beyond the mere representation of existing scenes, there is a growing interest in creating new, varied scenes sourced from an original scene via scene editing. The most convenient and straightforward way for users to communicate scene editing operations is through natural language – a task known as instruction-guided editing.

Success in this task for 2D images has been achieved by a 2D diffusion model, namely Instruct-Pix2Pix (IP2P) [1]. However, extending this capability to NeRF-represented 3D or 4D scenes poses a significant challenge. The inherent difficulty arises from the implicit nature of the NeRF representation, which lacks direct ways to modify the parameters in a targeted direction, along with the significantly increased complexity emerging in new dimensions. Recently, there has been noticeable progress in instruction-guided 3D scene editing, as exemplified by Instruct-NeRF2NeRF (IN2N) [10]. IN2N achieves 3D editing through distillation from 2D diffusion models such as IP2P to edit NeRF, *i.e.*, generating edited multi-view images from IP2P and fitting them on the NeRF-represented scenes. Due to the high diversity in generation results of diffusion models, IP2P may produce multi-view inconsistent images, with the same ob-

ject having different appearances in different views. Therefore, IN2N consolidates the results by training on NeRF to make it converge to an "average" editing result, which is reasonable but often encounters challenges in practice.

Further extending the editing task from 3D to 4D, however, introduces fundamental difficulties. With the additional time dimension beyond 3D scenes, it requires not only 3D *spatial consistency* for the 3D scene slice at each frame, but also the *temporal consistency* between different frames. Notably, as recent 4D NeRFs [7, 31] model the property of each absolute 3D location in the scene instead of the movement of individual object, the same object in different frames is not modeled by the same parameter. This deviation prevents NeRF from achieving spatial consistency by fitting inconsistent multi-view images, making the IN2N pipeline unable to effectively perform editing on 4D scenes.

This paper introduces Instruct 4D-to-4D, making the *first* attempt in instruction-guided 4D scene editing that overcomes the aforementioned issues. *Our key insight is to regard a 4D scene as a pseudo-3D scene*, where each pseudo-view is a video consisting of all frames from the same viewpoint. Subsequently, the task on the pseudo-3D scene can be tackled in a similar way as real 3D scenes, decoupled into two sub-problems: 1) achieve temporal-consistent editing for each pseudo-view, and 2) use the method from (1) to edit the pseudo-3D scene. Then, we can solve (1) with a video editing method, and leverage a distillation-guided 3D scene editing method to solve (2).

Specifically, we utilize an *anchor-aware attention* module to augment the IP2P model, inspired by [36]. The "anchor" in our module is a pair of an image and its editing result as a reference for the IP2P generation. The augmented IP2P now supports batched input of multiple images, and the self-attention module in the IP2P pipeline is substituted with a cross-attention mechanism against the anchor image of this batch. Consequently, IP2P generates editing results based on the correlation between the current image and the anchor image, ensuring consistent editing within this batch. However, the attention module may not always correctly associate objects in different views, introducing potential inconsistency.

To this end, we further propose an *optical flow-guided sliding window method* to facilitate video editing. Leveraging RAFT [32], we predict optical flow for each frame to establish pixel correspondence between two adjacent frames. This enables us to propagate editing results from one frame to the next, similar to a warping effect. With the augmented IP2P and optical flow, we can edit the video in temporal order, by segmenting frames and then applying editing to each segment while propagating the editing to the next segment. The process involves utilizing optical flow to initialize editing based on previous frames and subsequently applying the augmented IP2P with the last frame of the preceding segment serving as the anchor.

As a 4D scene contains a large number of frames at each view, it becomes time-consuming to compute all the views. To address this, we adopt a strategy inspired by ViCA-NeRF [5] to edit pseudo-3D scenes based on key views. We first randomly select key pseudo-views and edit them using the aforementioned method. Then for each frame, we employ depth-based projection to warp the results from the key views to other views, and utilize weighted average to aggregate the appearance information, obtaining the editing results for all the frames. Given the complexity of 4D scenes, we apply the iterative editing procedure of IN2N to iteratively generate edited frames and fit the NeRF on the edited frames, until the scene converges.

We conduct extensive experiments on both *monocular* and *multi-camera* dynamic scenes to validate the effectiveness of our approach. The evaluation shows the remarkable capabilities of our approach in both achieving sharper rendering results with significantly enhanced detail and ensuring spatial-temporal consistency in 4D editing (Fig. 1).

Our contribution is three-fold. (1) We introduce Instruct 4D-to-4D, a simple yet effective framework to perform instruction-guided 4D editing, by editing 4D scenes as pseudo-3D scenes via distillation from 2D diffusion models. (2) We propose the anchor-aware IP2P and the optical flow-guided sliding window method, enabling efficient and consistent editing of long videos or pseudo-views of any length. (3) With the proposed method, we develop a pipeline to iteratively generate fully and consistently edited datasets, achieving high-quality 4D scene editing in various tasks. Our work represents the first effort to investigate and address the general instruction-guided 4D scene editing, laying the foundation for this promising task.

2. Related Work

Diffusion-Based Video Editing. The diffusion-based generative models have achieved remarkable success in text-based image editing [1, 4, 11, 18, 22, 28, 34]. However, extending these models to video editing introduces greater complexity, necessitating the manipulation of visual attributes while maintaining temporal consistency. A prevalent approach in video editing using diffusion models is the transformation of Text-to-Image (T2I) models into Text-to-Video (T2V) models. Tune-A-Video [36] incorporates temporal self-attention layers into UNet and performs the one-shot tuning. Make-A-Video [30] and MagicVideo [41] augment their networks by introducing the spatio-temporal attention (ST-Attn) mechanism, enabling the seamless transition of a pre-trained Text-to-Image model to the temporal dimension. Further, there is a growing focus on localized editing through the manipulation of attention maps inspired by Prompt-to-Prompt [11] and Plug-and-Play [34]. Video-P2P [16] introduces decoupled-guidance attention

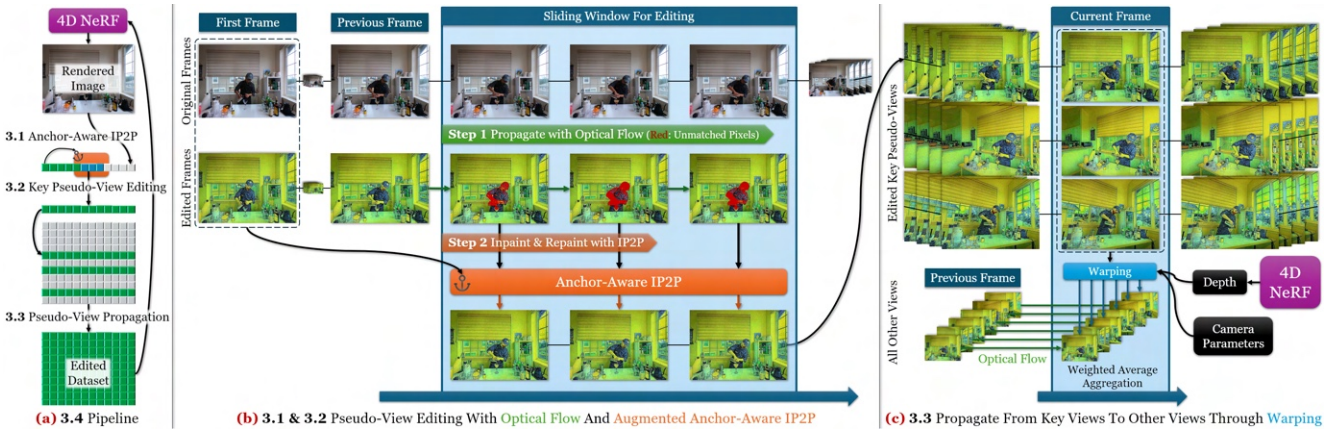

Figure 2. Our Instruct 4D-to-4D edits a 4D scene by regarding it as a pseudo-3D scene with multiple pseudo-views, and then editing these pseudo-views in an iterative key frame-based pipeline. (a) Our pipeline edits the 4D scene by iteratively generating a fully edited dataset used to fit 4D NeRF. In each iteration, we first (b) edit each key pseudo-view through optical flow propagation and IP2P inpainting and repainting, and then (c) edit other pseudo-views by aggregating propagated results from both previous frames through optical flow, and the key pseudo-views at current frame through depth-based warping.

control to preserve the semantic consistency. Pix2Video [3] utilizes the self-attention feature injection to propagate modifications made in the anchor frame to other frames. Fatezero [26] fuses self-attentions with a blending mask extracted from cross-attention features to achieve the zero-shot shape-aware editing.

Diffusion-Based NeRF Editing. Diffusion-based NeRF editing has been gaining increasing attention in recent times. Some works leverage powerful SD as 2D prior to modifying the appearance of scenes, producing impressive results. Instruct 3D-to-3D [13] and Instruct-NeRF2NeRF (IN2N) [10] employ Instruct-Pix2Pix (IP2P) [1], an image-conditioned diffusion model, to enable instruction-based 2D image editing. Specifically, Instruct 3D-to-3D uses score distillation sampling (SDS) [24] loss to edit 3D NeRFs using the 2D diffusion-prior. Meanwhile, Instruct-NeRF2NeRF proposes Iterative Dataset Update (Iterative DU) to alternate between editing the images rendered from NeRF using the diffusion model and updating the NeRF representation with the supervision of edited images during the training process. ViCA-NeRF [5] follows IN2N and utilizes the depth information derived from the NeRF to propagate the modification in key views to other views, achieving spatial consistency. DreamEditor [43] leverages DreamBooth [28] as 2D prior and utilizes the SDS loss to optimize the meshed-based neural field, performing faithful editing to the text. Control4D [29] proposes to build a more continuous 4D space by learning a 4D GAN [9] from the ControlNet [39] to avoid inconsistent supervision signals for 4D portrait editing.

NeRF-Based Dynamic Scene Representation The field of representing dynamic scenes using Neural Radiance Fields (NeRFs) [19] has seen significant advancements, which are essential for various real-world applications. Various methods have been developed to extend the capabilities of NeRFs in capturing and rendering dynamic scenes. DNeRF [25] and Nerfies [20] employ individual MLPs to represent a deformation field and a canonical field for capturing complex scene changes over time. DyNeRF [15] integrates time-conditioning into NeRFs using a set of compact latent codes. TiNeuVox [6] employs an explicit voxel grid to model temporal information. Additionally, NeuralBody [23] and [35, 38, 40] focus on acquiring precise dynamic human body motion information, building upon the SMPL [17] model. HexPlane [2] and K-Planes [7] propose a planar factorization to decompose 4D spatiotemporal volumes into six feature planes. NeRFPlayer [31] decomposes the 4D space into static, deforming, and new areas based on their temporal characteristics. Despite these advancements, a common limitation across these methods is the lack of user-friendly editing capabilities for dynamic scenes. Users are currently unable to freely edit or modify these scenes, particularly in terms of following specific instructions. This limitation highlights an area for potential future research and development, where user interactivity and editing capabilities could be integrated into the dynamic scene representation models. Addressing this challenge would significantly enhance the practicality and applicability of NeRF-based dynamic scene representations.

3. Method

We propose Instruct 4D-to-4D, a novel pipeline that edits 4D scenes by distilling from Instruct-Pix2Pix (IP2P) [1], a powerful 2D diffusion model that supports instruction-guided image editing. The basic idea of our method roots in ViCA-NeRF [5], a key view-based editing. By regarding the 4D scene as a pseudo-3D scene where each pseudo-view is a video of multiple frames, we apply the key view-based editing, broken down into two steps: key pseudo-view editing, and propagation from key pseudo-views to other views, as shown in Fig. 2. We propose several key components to

enforce and achieve spatial and temporal consistency during these steps, generating 4D consistent editing results.

3.1. Anchor-Aware IP2P for Consistent Batched Generation

Batch Generation with Pseudo-3D Convs. The editing process for a pseudo-view can be regarded as editing a video. Therefore, we need to enforce temporal consistency when editing each frame. Inspired by previous work on video editing [16, 36], we edit a batch of images together in IP2P, and augment the UNet in IP2P to make the generation in consideration of the whole batch. We upgrade its 3×3 2D convolutional layers to $1 \times 3 \times 3$ 3D convolutional layers by reusing the original parameters of kernels.

Anchor-Aware Attention Module. Limited by the GPU memory, we cannot edit all frames of a pseudo-view all together in one batch, and need to separate the generation into multiple batches. Therefore, it is crucial to keep the consistency between batches. Following the idea of Tune-a-Video [36], instead of generating the edited result of the new batch from scratch, we allow the model to reference an anchor frame shared across all the generation batches, with its original and edited version, to "propagate" the editing style from it to the new edited batch. By substituting the self-attention module in the IP2P with the cross-attention model against the anchor frame, we would be able to connect the same objects between the current image and the anchor image, and generate the new edited images by mimicking the anchor's style, to perpetuate the consistent editing style from the anchor. Notably, our usage of the anchor-attention IP2P is different from Tune-a-Video, which queries cross-attention between the anchor frame and the previous frame instead of the current frame. Our design also further facilitates the inpainting procedure in Sec. 3.2, which also requires a focus on the existing part of the current frame.

Effectiveness. Fig. 3 shows the generation results of different versions of IP2P. The original IP2P edits all images inconsistently, in different color distributions, even for images within one batch. With anchor-aware attention layers, IP2P is able to generate the batch as an entirety and, therefore, generates consistent editing results within one batch. However, it is still unable to generate consistent images within different batches. With reference to the same anchor image shared across batches, the full anchor-aware IP2P is able to generate consistent editing results for all 6 images across 2 batches, showing that even without additional training, the anchor-aware IP2P would be able to achieve consistent editing results.

3.2. Optical Flow-Guided Sliding Window Method for Pseudo-View Editing

Optical Flow as 4D Warping. To enforce the temporal consistency in a pseudo-view, we need the correspondence of the pixels between different frames. 3D scene editing methods [5, 14, 37] exploits depth-based warping to find the correspondence of different views, using a deterministic method with NeRF-predicted depth and camera parameters. In 4D, however, there are no such explicit ways. Therefore, we use an optical flow estimation network RAFT [32] to predict the optical flow, in a format of 2D motion vector for each pixel, which can be derived into correspondence pixel in another frame. Using RAFT, we are able to warp between adjacent frames, like in 3D. As each pseudo-view is taken at a fixed camera location, optical flow performs well.

Sliding Window Method. We follow the idea of video editing methods [12, 16, 33, 36] to edit a pseudo-view. However, those methods focus only on short videos and apply video editing by editing all the frames in a single batch, making them unable to deal with long videos. Therefore, we propose a novel sliding window (of width B being the maximum allowed batch size) method to exploit the anchor-aware IP2P along with the optical flow. As shown in Fig. 2 (a), for the current window containing B images, say frames $t, t+1, \cdots, t+B-1$, we first propagate the editing results from frame $t-1$ to all these B images one by one with 4D warping. For the unmatched pixels, which correspond to the occluded part in frame $t-1$, we set their value as their origin, obtaining a fused image of the original and warped editing images.

Then, similar to the idea in ViCA-NeRF [5], we use IP2P to inpaint and repaint the fused image at each view in the sliding window, by adding noise to the fused image and denoising using IP2P, so that the generated edited image will follow a similar pattern on the warped results, while repainting the whole image to make it natural and reasonable. To make the style all-consistent over the pseudo-view, we use the first frame as the anchor shared across all the windows, so that the model will generate images in a consistent style like the first view. As camera is fixed for one pseudo-view, there are many common objects between different frames, therefore such a method is very effective in producing consistent editing results for the frames in the window.

After completing the editing in the current window, we will advance the window by B frames. Therefore, for a pseudo-view of T frames in total, our Instruct 4D-to-4D only needs to call IP2P for T/B times. By caching the optical flow prediction between adjacent frames in one view, we could achieve temporal-consistent pseudo-view editing very efficiently.

3.3. Pseudo-View Propagation Based on Warping

Generating First Frame. As we need to propagate the edited pseudo-views to all other views while achieving spatial consistency, it is crucial to edit the first frames at all key pseudo-views in a spatially consistent way – they are not only used to start the editing of the current pseudo-view,

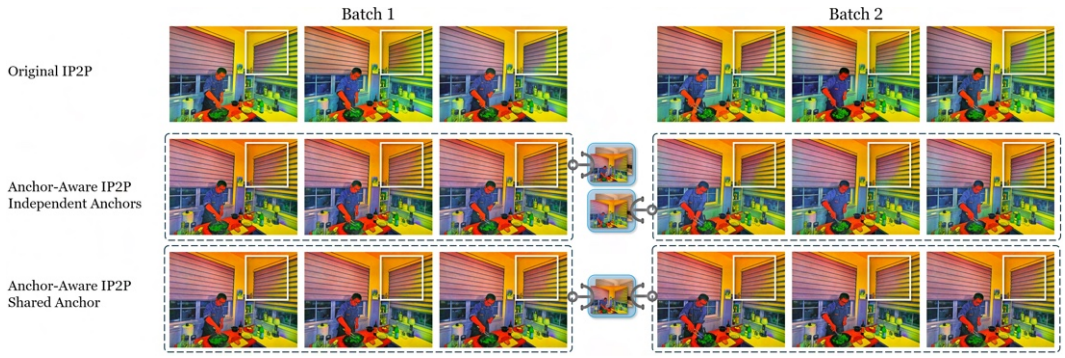

Figure 3. Generation results show that our augmented IP2P achieves *consistency within a batch* via our anchor-aware attention module, and achieves *consistency between different batches* via the same anchor shared across batches. The white bounding box shows the most noticeable part of inconsistency.

but also used as the anchor or the reference for all the proceeding generations. Therefore, we first edit one first frame in an arbitrary pseudo-key view, then use our anchor-aware IP2P with it as the anchor to generate other first frames. In this way, all the first frames are edited in a consistent style, being a good start to editing the key pseudo-views.

Propagate from Key Views to Other Views. After editing the key pseudo-views, aligned with ViCA-NeRF [5], we propagate their editing results to all other key views. ViCA-NeRF uses depth-based *spatial warping* to warp an image from another view at the same timestep, while we also propose optical-flow-based *temporal warping* to warp from the previous frame at the same view. With these two types of warping, we can warp the edited images from multiple sources.

We propagate for each timestep in the order of time. When we propagate at timestep t, for each frame (v, t) at view v, we obtain its edited version as the weighted average of warped results from two sources: (1) the edited results of the previous frame at the same view, namely frame $(v, t-1)$, using temporal warping; and (2) the edited results of the current frame at one each of the key view, using spatial warping. By propagating the frames for all the timesteps, we obtain a consistent edited dataset containing all the editing frames. We use such a dataset to train NeRF towards the edited results.

With this propagation method, we would be able to efficiently generate a full dataset of consistently edited frames within nT/B time-consuming IP2P generations, with n key pseudo-views out of all V pseudo-views, where in our experiments $n = 5$ while V can be more than 20. Such high efficiency makes it possible to deploy an iterative pipeline to update the datasets.

3.4. Overall Editing Pipeline

Iterative Dataset Update. Following the idea of IN2N [10], we apply iterative dataset replacement on our baseline that iteratively re-generates the full dataset using the methods in Secs. 3.1,3.2,3.3, and fits our NeRF on it. In each iteration, we first randomly select several pseudo-views as the key views in this generation. We use the method in Sec. 3.3 to generate spatial-consistent editing results for the first frames of all these key pseudo-views, then propagate the editing for all pseudo-views using the sliding window method in Sec. 3.2. After obtaining all the edited key pseudo-views, we use the method in Sec. 3.3 to generate spatial and temporal consistent editing results for all other pseudo-views, ending up with a consistent edited dataset. We replace the 4D dataset with this edited dataset, and fit NeRF on it.

Improving Efficiency Through Parallelization and Annealing Strategies. In our pipeline, the NeRF only needs to be trained on the dataset and provide current rendering results, while IP2P only needs to generate results according to NeRF's rendering to form new datasets - there are few dependencies and interactions between IP2P and NeRF. Therefore, we parallelize our pipeline by running these two parts asynchronously on two GPUs. In the first GPU, we train NeRF continuously with the current dataset, while caching NeRF's rendering results in a rendering buffer; while in the second GPU, we apply our iterative dataset-generation pipeline to generate new datasets, using the images from the rendering buffer, and update the dataset used to train NeRF. In this case, we maximize the parallelization by minimizing the interactions, leading to a significant reduction in the training time.

On the other hand, to improve the generation results and convergence speed, we apply the annealing trick from HiFA [42] to achieve fine-grained editing on NeRF. The high-level idea is that we use the noise level to control the similarity of rendered results and IP2P's editing results. We generate the dataset at a high noise level to generate sufficiently edited results, and then gradually anneal the noise level to stick to the edited results that NeRF is converging to and refine such results. Instead of IN2N which always generates at a random noise level, our Instruct 4D-to-4D could converge to high-quality editing results at a fast speed.

With these two techniques, our Instruct 4D-to-4D is able to edit a large-scale 4D scene with 20 views and hundreds of frames in only hours.

Figure 4. **Qualitative results on various scenes** demonstrate that our Instruct 4D-to-4D generates high-qualify editing results in style transfer tasks on various scenes. The edited scenes are well-consistent with the instructed style, showing bright colors and natural textures.

4. Experiments

Editing Tasks and NeRF Backbone. The 4D scenes we use for evaluation are captured by single hand-held cameras and multi-camera arrays including: (I) *Monocular* Scenes in DyCheck [8] and HyperNeRF [21], which are simple, object-centric scenes with a single moving camera; and (II) *Multi-camera* Scenes in DyNeRF/N3DV [15], including indoor scenes with face-forward perspective and human motion structure. For monocular scenes, we edit all the frames as a single pseudo-view. We use the NeRFPlayer[31] as our NeRF backbone to produce high-quality rendering results of 4D scenes.

Baselines. Instruct 4D-to-4D is the first work on instruction-guided 4D scene editing. No previous work focuses on the same task, while the only similar work Control4D [29] has not released their code. Therefore, we cannot conduct any baseline comparison with existing methods. To show the effectiveness of our Instruct 4D-to-4D, we construct a baseline IN2N-4D, by naively extending IN2N [10] to 4D, which iteratively generates one edited frame and add it to the dataset. We compare our Instruct 4D-to-4D with IN2N-4D both qualitatively and quantitatively. To quantify the results, as both our pipeline and the model are training NeRF with generated images, we use traditional NeRF [19] metrics to evaluate the results, namely PSNR, SSIM, and LPIPS, between the IP2P generated images (generating from pure noise so that it will not be conditioned on NeRF's rendering image) and the NeRF's rendering results. We conduct our ablation studies against Instruct 4D-to-4D variants in the supplementary.

Qualitative Results. Our qualitative results are shown in Figs. 6, 5, and 4. The qualitative comparison with baseline IN2N-4D is in Figs. 5 and 6. As shown in Fig. 5, in the task of changing the cat into a fox in the monocular scene, IN2N-4D generates blur results with multiple artifacts: multiple ears, multiple noses and mouths, *etc*., while our Instruct 4D-to-4D generates photo-realistic results where the shape of the fox is well aligned with the cat in the original scene, with clear textures on the fur and no artifacts. These results show that our anchor-aware IP2P, optical flow-based warping, and sliding window method for pseudo-view editing produces temporal-consistent editing results for a pseudo-view. Without such a module, the original IP2P in IN2N-4D produces inconsistent edited images for each frame, consolidating to a strange result on the 4D NeRF. Fig. 6 shows the style transfer results on multi-camera scenes. Our parallelized Instruct 4D-to-4D achieves consistent style transfer results that match the description in a very short time period of two hours, while IN2N-4D takes 24$\times$ longer than our Instruct 4D-to-4D but still fails to get the 4D NeRF converged to the indicated style. This shows that 4D scene editing is highly non-trivial, while our Instruct 4D-to-4D's strategy to iteratively generate a full edited dataset facilitates high-efficiency editing. All these results collectively show that all

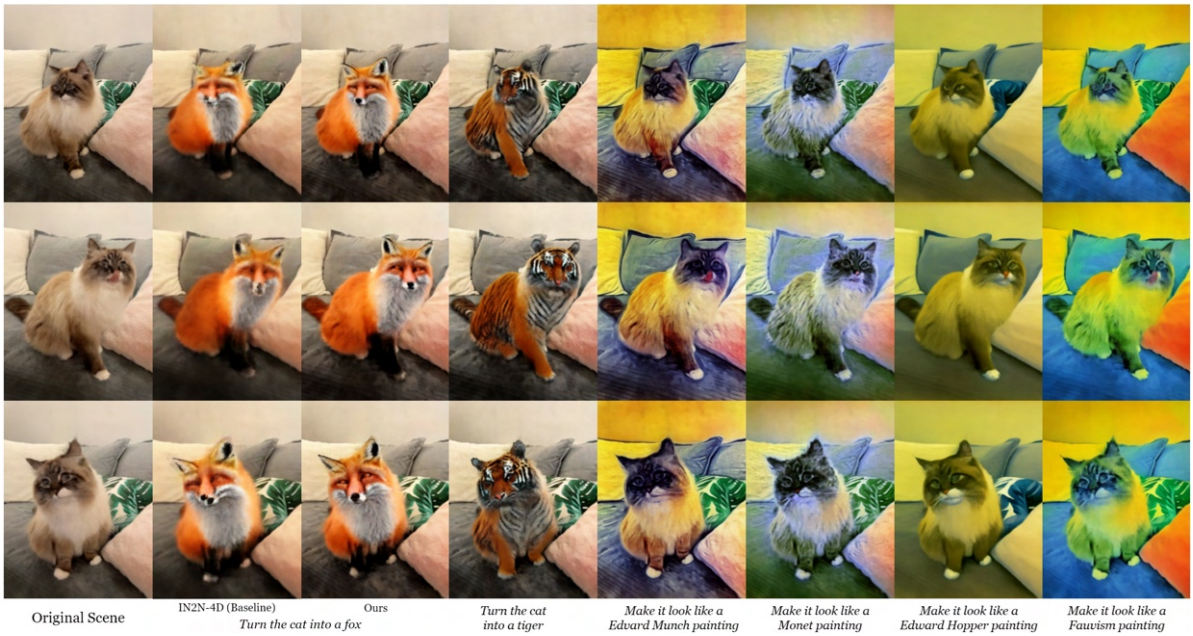

Figure 5. **Qualitative results on `mochi-high-five` scene in DyCheck dataset** show that our Instruct 4D-to-4D achieves high-quality editing results over various editing instructions in the monocular scene. Our Instruct 4D-to-4D can even achieve consistent editing with complicated textures, *e.g.*, in the Tiger editing, while baseline IN2N-4D generates blurred results with lots of artifacts.

Figure 6. **Qualitative comparison with baseline IN2N-4D** on multi-camera `coffee_martini` shows that our Instruct 4D-to-4D generates high-quality, faithful style transfer editing results in a very short time. As a comparison, IN2N-4D even fails to converge at any style with **24**$\times$ time consumption.

our design of Instruct 4D-to-4D is reasonable and effective, and Instruct 4D-to-4D can produce high-quality editing results in a very efficient way.

The experiments in Fig. 5 show the monocular scene `mochi-high-five` under different instructions, including local editing on the cat, or style transfer instructions for the whole scene. Our Instruct 4D-to-4D achieves photo-realistic local editing results in the Fox and Tiger instructions, with clear and consistent textures *e.g.*, the stripes of the tiger. In the style transfer instructions, the edited scene faithfully reflects the style indicated in the prompts. These show Instruct 4D-to-4D's great ability in editing monocular scenes under various prompts.

The experiments in Fig. 4 show other style transfer results, including monocular scenes in HyperNeRF and DyCheck and multi-camera scenes in DyNeRF. Instruct 4D-to-4D consistently produces high-fidelity style transfer results with bright colors and clear appearance in various styles.

Quantitative Comparison. The quantitative comparison between our Instruct 4D-to-4D and baseline IN2N-4D on the multi-camera `coffee_martini` scene is in Tab. 1. Consistent with the qualitative comparison, our Instruct 4D-

Instruction	Method	PSNR↑	SSIM↑	$\text{LPIPS}_{\text{Alex}}$↓	$\text{LPIPS}_{\text{VGG}}$↓
Van Gogh	Ours	**23.62**	**0.820**	**0.220**	**0.349**
	IN2N-4D	17.43	0.645	0.466	0.573
Hopper	Ours	**17.47**	**0.533**	**0.356**	**0.429**
	IN2N-4D	11.96	0.299	0.655	0.645
Munch	Ours	**12.92**	**0.362**	**0.520**	**0.598**
	IN2N-4D	11.96	0.299	0.655	0.645
Fauvism	Ours	**18.86**	**0.728**	**0.245**	**0.377**
	IN2N-4D	14.15	0.520	0.408	0.532
(Average)	Ours	**19.67**	**0.635**	**0.323**	**0.405**
	IN2N-4D	14.11	0.457	0.512	0.587

Table 1. In the quantitative evaluation on the multi-camera `coffee_martini` scene, our Instruct 4D-to-4D significantly and consistently outperforms the baseline IN2N-4D in all metrics.

to-4D significantly and consistently outperforms the baseline IN2N-4D. This shows that the NeRF trained by Instruct 4D-to-4D fits the IP2P's editing results much better than the baseline, further validating the effectiveness of our Instruct 4D-to-4D.

Ablation Study: Variants and Settings. We validate our design choices by comparing our approach to the following variants.

- *IN2N-4D.* A baseline method based on IN2N [10].
- *Video Editing.* This variant serves as the most basic implementation of our Instruct 4D-to-4D – edit each pseudo-frame with any video editing methods, and propagate to other frames using 3D warping. We use a zero-shot text-driven video editing model, FateZero [26], via pretrained stable diffusion models. We follow the settings of the official Fatezero implementation in style editing and attribute editing. Since they can only process 8 video frames in a batch, we use a batch-by-batch editing strategy for pseudo-view editing.
- *Anchor-Aware IP2P w/o Optical-Flow.* In this variant, we perform video editing without optical flow guidance, in which anchor-aware IP2P directly edits all training images using the same diffusion model setting.
- *One-time Pseudo-View Propagation.* In this variant, we perform only one-time pseudo-view propagation, in which all rest-pseudo-views are warped from 4 randomly selected key-pseudo-views, and the NeRF is trained until convergence on those edited images.

The task for ablation study is "What if it was painted by Van Gogh" on the `coffee_martini` in DyNeRF dataset. As the "Video Editing" variant does not use the same diffusion model, IP2P [1], to edit the video, we cannot use the metrics in the main paper. Therefore, consistent with IN2N [10], we use CLIP [27] similarity to evaluate how successful an editing operation is applied.

Ablation Study: Results. The qualitative results are in Fig. 7 and the demo video. Most of the variants do not apply sufficient editing to the scene, with a gloomy appearance and no Van Gogh's representative color. This shows that our Instruct 4D-to-4D's design choices are effective and crucial to achieve high-quality editing.

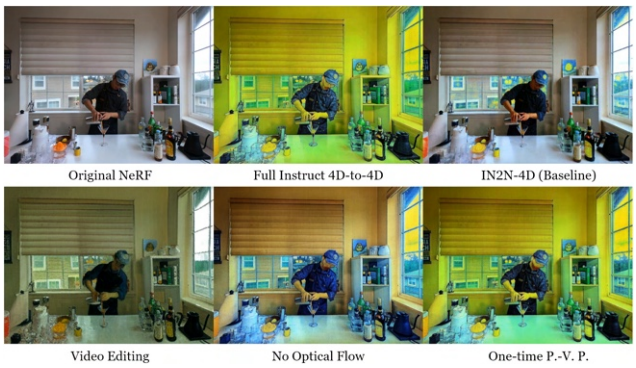

Figure 7. Most of the variants are unable to edit the scene sufficiently, showing that the design choices of our Instruct 4D-to-4D are reasonable and effective. These results are also shown in the demo video.

Type	Name	CLIP Similarity↑
Baselines	IN2N-4D	0.2790
Our Variants	Video Editing	0.2631
	Ours w/o Optical Flow	0.2792
	One-time P.-V. P.	0.2876
	Full (Ours)	**0.3085**

Table 2. Ablation study showing the effectiveness of our design choices. The optical flow and iterative editing pipeline help achieve successful editing.

The quantitative comparisons are in Tab. 2. Our full Instruct 4D-to-4D achieves significantly higher CLIP similarity in the ablation task, showing that our design is effective. Also, we observe that the Video Editing strategy cannot even achieve a better metric than IN2N-4D, which shows that it is non-trivial to edit the 4D scene even after converting it to a pseudo-3D scene.

5. Conclusion

This paper proposes Instruct 4D-to-4D, the first instruction-guided 4D scene editing framework that edits 4D scenes by regarding them as pseudo-3D scenes and applies an iterative strategy to edit pseudo-3D scenes using a 2D diffusion model. Qualitative experimental results show that Instruct 4D-to-4D achieves high-quality editing results in various tasks, including monocular and multi-camera scenes. Instruct 4D-to-4D also significantly outperforms the baseline, a naive extension of the state-of-the-art 3D editing method to 4D, showing the difficulty and non-trivialness of the task and the success of our method. We hope that our work could inspire more future work on 4D scene editing.

Acknowledgement. This work was supported in part by NSF Grant 2106825, NIFA Award 2020-67021-32799, the Jump ARCHES endowment, and the IBM-Illinois Discovery Accelerator Institute. This work used NVIDIA GPUs at NCSA Delta through allocations CIS220014 and CIS230012 from the ACCESS program.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 2, 3, 8

[2] Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *CVPR*, 2023. 3

[3] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023. 3

[4] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[5] Jiahua Dong and Yu-Xiong Wang. ViCA-NeRF: View-consistency-aware 3D editing of neural radiance fields. In *NeurIPS*, 2023. 2, 3, 4, 5

[6] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia*, 2022. 3

[7] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2, 3

[8] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *NeurIPS*, 2022. 6

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 3

[10] 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, 8

[11] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[12] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Stylizing video by example. *TOG*, 2019. 4

[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

[14] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative score distillation for consistent visual editing. In *NeurIPS*, 2023. 4

[15] 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. 3, 6

[16] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761*, 2023. 2, 4

[17] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. ACM, 2023. 3

[18] 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. *arXiv preprint arXiv:2108.01073*, 2021. 2

[19] 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, 3, 6

[20] 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. 3

[21] 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. *arXiv preprint arXiv:2106.13228*, 2021. 6

[22] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023. 2

[23] 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. 3

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3

[25] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 3

[26] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535*, 2023. 3, 8

[27] 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. 8

[28] 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. 2, 3

[29] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Dynamic portrait editing by learning 4D gan from 2d diffusion-based editor. *arXiv preprint arXiv:2305.20082*, 2023. 3, 6

[30] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[31] 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. 2, 3, 6

[32] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV 2020*, 2020. 2, 4

[33] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menclei Chai, Sergey Tulyakov, and Daniel Sỳkora. Interactive video stylization using few-shot patch-based training. *TOG*, 2020. 4

[34] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 2

[35] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 3

[36] 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. 2, 4

[37] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3D-aware image generation using 2D diffusion models. In *ICCV*, 2023. 4

[38] Zhen Xu, Sida Peng, Chen Geng, Linzhan Mou, Zihan Yan, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Relightable and animatable neural avatar from sparse-view video. *arXiv preprint arXiv:2308.07903*, 2023. 3

[39] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3

[40] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *CVPR*, 2022. 3

[41] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 2

[42] Joseph Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3D with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023. 5

[43] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. Dreameditor: Text-driven 3D scene editing with neural fields. *arXiv preprint arXiv:2306.13455*, 2023. 3