

# V<sup>2</sup>Edit: Versatile Video Diffusion Editor for Videos and 3D Scenes

Yanming Zhang<sup>1†</sup> Jun-Kun Chen<sup>2†</sup> Jipeng Lyu<sup>2</sup> Yu-Xiong Wang<sup>2</sup>

<sup>1</sup>Zhejiang University <sup>2</sup>University of Illinois Urbana-Champaign <sup>†</sup>Equal Contribution

yanmingzhang@zju.edu.cn {junkun3, jipeng2, yxw}@illinois.edu

[immortalco.github.io/V2Edit](https://immortalco.github.io/V2Edit)

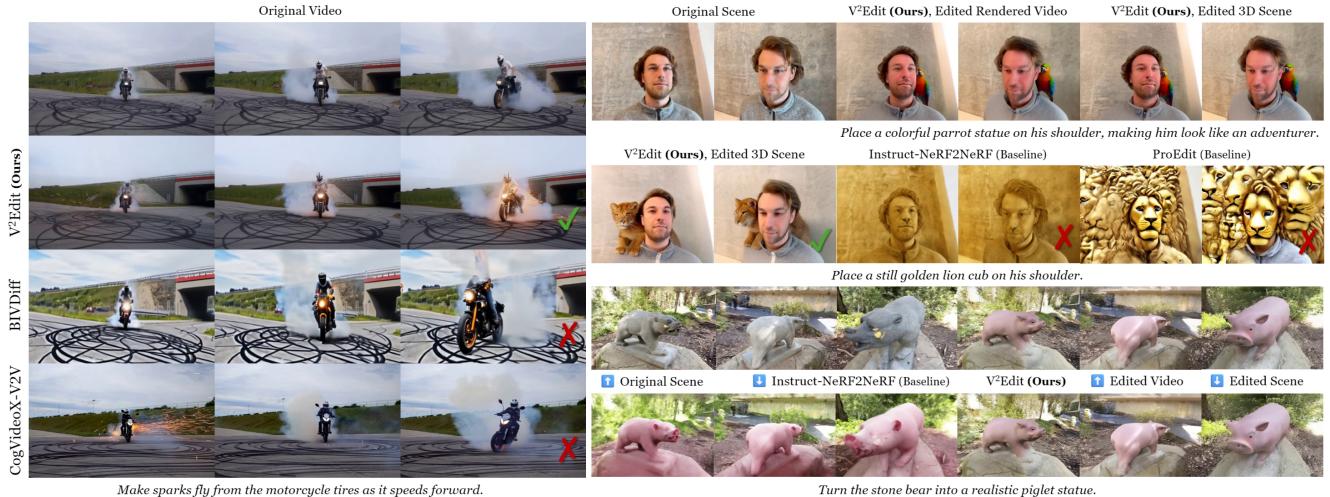


Figure 1. Our **V<sup>2</sup>Edit** is a versatile approach that supports *training-free* instruction-guided editing for both videos and 3D scenes. **Left:** V<sup>2</sup>Edit achieves high-quality editing satisfying both *original content preservation* and *editing instruction fulfillment* in video editing. **Right:** V<sup>2</sup>Edit supports challenging 3D scene editing tasks involving *significant geometric changes*, which baselines [4, 10] fail to achieve.

## Abstract

This paper introduces *V<sup>2</sup>Edit*, a novel training-free framework for instruction-guided video and 3D scene editing. Addressing the critical challenge of balancing original content preservation with editing task fulfillment, our approach employs a progressive strategy that decomposes complex editing tasks into a sequence of simpler subtasks. Each subtask is controlled through three key synergistic mechanisms: the initial noise, noise added at each denoising step, and cross-attention maps between text prompts and video content. This ensures robust preservation of original video elements while effectively applying the desired edits. Beyond its native video editing capability, we extend *V<sup>2</sup>Edit* to 3D scene editing via a “render-edit-reconstruct” process, enabling high-quality, 3D-consistent edits even for tasks involving substantial geometric changes such as object insertion. Extensive experiments demonstrate that our *V<sup>2</sup>Edit* achieves high-quality and successful edits across various challenging video editing tasks and complex 3D scene editing tasks, thereby establishing state-of-the-art performance in both domains.

## 1. Introduction

Video diffusion models have rapidly gained prominence in computer vision [1, 2, 15, 24, 41], following the success of image-based diffusion generative models [34]. These models now enable the generation of high-resolution, high-fidelity videos from text descriptions. Meanwhile, instruction-guided video editing – modifying existing videos through simple text instructions – has become an emerging area of focus. Using a high-quality initial video allows for efficient creation of new video assets through targeted edits, rather than generating them from scratch.

Despite such progress, video editing remains under-explored due to the lack of large-scale paired video datasets essential for training end-to-end models. Traditional image-based methods [7, 17, 21, 28, 32] approach video editing by applying image editing techniques to individual frames, one at a time. The advent of video diffusion models has enabled *training-free* video-based methods [9, 35] that leverage pre-trained video diffusion models without the need for additional training [16, 27]. However, these methods still face challenges, including temporal inconsistencies, as well

as difficulties in handling fast-moving camera trajectories, complex motions, and significant temporal variations.

Our main observation is that video editing tasks require simultaneously achieving two objectives: the edited video should both *fulfill the editing instruction* and *preserve the original content*. This ensures that only the targeted areas are modified, while all other components remain unchanged. Balancing these two aspects, however, presents a key challenge in video editing. The existing training-free models are often driven to produce videos that comply with the editing instructions but fail to preserve the original content, as well as requiring extensive hyperparameter tuning to achieve a balance.

This paper proposes V<sup>2</sup>Edit, a novel framework for versatile video editing that introduces effective and synergistic control mechanisms, robustly preserving the original content while remaining flexible to allow for the intended edits. To preserve the original video content during editing, *our first key insight* is to systematically control the denoising process in video diffusion models from complementary perspectives: **(i)** the initial noise, **(ii)** the noise added at each denoising step, and **(iii)** the cross-attention maps between text prompts and the video.

Specifically, the noise addition process in diffusion models initially disrupts high-frequency details and later affects low-frequency information. By restricting the noise addition to the early steps and starting generation from this initial noise, we can preserve low-frequency features, such as the *overall layout*. This observation further suggests that noise carries semantic information in diffusion-based generation. Therefore, using a noise scheduler that incrementally adds noise at each step can help transfer *semantic* information from the original video to the edited one. Additionally, during the denoising process, the model’s cross-attention maps – showing *correspondences* between textual prompts and specific objects or regions – can be explicitly leveraged to control the preservation of the original content.

Conceptually, these control mechanisms may need to be tailored to suit different editing tasks: mild edits can allow for stronger preservation of the original content, whereas significant edits may be compromised with too strict preservation control. To avoid such complexity, *our second key insight* is to *decompose* a complex editing task into a sequence of mild subtasks, progressively completing each subtask. For each subtask, it becomes easier to balance the original content preservation with editing sub-instruction fulfillment. More importantly, this can be achieved with a *consistent control strategy* across subtasks derived from different editing tasks, rather than relying on a more complex, task-varying, or *hyperparameter-sensitive* approach.

Beyond its native video editing capability, our V<sup>2</sup>Edit can also be seamlessly applied to 3D scene editing, making it a *unified* editing solution. We propose a simple yet

effective “render-edit-reconstruct (RER)” process to leverage video editing methods for 3D editing, by first rendering a video of the scene along a fixed camera trajectory, editing the rendered video, and then reconstructing the scene from the edited video. The temporal consistency of the rendered video ensures strong 3D consistency in the reconstructed scene. In fact, 3D scene editing is a *unique* capability of our V<sup>2</sup>Edit, where existing video editing methods [7, 9, 18, 28, 35] struggle to render videos with large-scale camera motions and significant temporal variations.

In a variety of challenging video and 3D scene editing tasks, our V<sup>2</sup>Edit achieves high-quality results, as shown in Fig. 1. For video editing, our method handles more complex scenarios, including longer videos, faster-moving camera trajectories, and greater temporal variations. In 3D scene editing, V<sup>2</sup>Edit notably supports significant geometric changes, such as object insertion, which previous 3D scene editing methods [4, 10, 38] have not been able to accommodate. Additionally, our approach enables efficient video editing without the need for time-consuming, iterative per-view adjustments, ensuring rapid convergence.

**Our contributions** are threefold. **(i)** We propose V<sup>2</sup>Edit, a simple yet versatile framework for training-free, instruction-guided video and 3D scene editing. **(ii)** We introduce synergistic mechanisms that systematically control the denoising process in video diffusion and enable progressive editing, effectively balancing the preservation of original video content with the fulfillment of editing instructions, all within a unified framework for diverse editing tasks. **(iii)** V<sup>2</sup>Edit consistently achieves high-quality, successful edits across various video and 3D scene editing tasks, including those previously unsolvable by existing methods, thereby establishing state-of-the-art performance in both domains.

## 2. Related Work

**Video Diffusion Models.** The success of diffusion models in image generation has been extended to video generation [14]. Early approaches [2, 13, 14, 36, 45] design the video diffusion model based on the UNets of image diffusion models, to support the 3D-shaped inputs for videos. To save memory and compute, instead of directly lifting the convolutional layers and attention layers from 2D to 3D, they keep the existing 2D layers to be applied individually to each frame, while inserting temporal convolutional and attention layers. This decomposes the computation of spatial and temporal components of videos, and also makes it possible to extend pre-trained image diffusion models by only tuning the temporal generation capability through fine-tuning [2, 13, 36]. Later, Stable Video Diffusion (SVD) [1] scales up the video diffusion models for high-resolution, high-quality video generation through careful data selection and multi-stage training, and also extends to the generation of 3D [41] and 4D [44] contents. The release of SORA

[24] has lit a new way to scale up video diffusion models with diffusion transformers (DiTs). Instead of applying downsampling and decomposed attention layers in UNets, DiTs directly turn the whole video (or video latents) into a sequence of patches, and apply a full 3D attention within all the patches. Inspired by this, CogVideoX [46] is proposed upon its previous effort CogVideo [15], using DiT-based video diffusion models and significantly improving the video length, resolution, and generation quality.

**Video Editing.** Due to the lack of paired training data for video editing methods – *i.e.*, triples of “editing instruction, original video, and edited video” – most existing video editing approaches are training-free. Traditional video editing methods [7, 17, 21, 28, 32] are image-based methods, which rely on an underlying model or method with image editing capability and introduce other add-ons to control the consistency. For example, FateZero [32], Tune-A-Video [42], and Instruct 4D-to-4D [28] use 2D diffusion models to edit videos by zero-shot extending the standard spatial attention layers in the UNets into spatial-temporal attention layers to account for the first and the previous frame in the generation. After the emergence of video diffusion models, there are also several efforts that edit videos by utilizing the generation and smoothness capability of video diffusion models. BIVDiff [35] uses a pre-trained video diffusion model to refine the temporal-inconsistent per-frame edited images into a smooth, temporal-consistent video. VideoShop [9] takes the edited first frame of the video as input along with the video and propagates the editing operations through the following frames. CogVideoX-V2V [46] applies SDEdit [27] to edit videos using the generation capability. However, unlike our V<sup>2</sup>Edit, all these methods struggle to perform aggressive editing on complex scenarios, like fast-moving cameras, changing backgrounds and contents, and geometry or motion changes. Many training-free methods require task-specific hyperparameter tuning to manually balance editing fulfillment and original preservation, while our hyperparameter-tuning-free V<sup>2</sup>Edit achieves a consistent preservation strategy across all editing tasks.

**Diffusion-Based 3D Scene Editing.** For instruction-guided 3D scene editing, the traditional way is to distill the editing signals from a 2D diffusion applied on each view to the 3D scene with score distillation sampling (SDS) [31]. Instruct-NeRF2NeRF [10] is the first paper in this direction, which applies an SDS-equivalent iterative dataset update to iteratively update the dataset of edited views to train the NeRF. There are also many works in this direction, aiming to improve the efficiency [38], distillation method [18, 19], 3D consistency [4, 5], or extension to 4D [28]. Video diffusion models show a natural and straightforward way to replace the image diffusion model with a video diffusion model, to edit a rendered video of the scene directly. As the temporal consistency of the edited rendered video is a strong prereq-

uisite of 3D consistency of the 3D scene, this could potentially significantly reduce the difficulty of maintaining 3D consistency. However, the rendered videos of a 3D scene should cover all possible viewpoints of the scene, which requires the content to change a lot to cover the whole scene, and the camera trajectory covers all possible viewpoints. All these make such video editing a very challenging case in video editing tasks. To our knowledge, no existing work utilizes video diffusion models to edit 3D or 4D scenes.

### 3. Methodology

In V<sup>2</sup>Edit, we leverage pre-trained video diffusion models as the foundation for versatile video editing without requiring specific training on paired datasets. Our framework, illustrated in Fig. 2, employs a progressive editing process that decomposes complex editing tasks into a sequence of simpler subtasks. To preserve the original video content while ensuring high-quality edits, we implement a training-free preservation control mechanism that systematically manages three key aspects of the diffusion process: (i) the initial noise, (ii) the noise added at each denoising step, and (iii) the cross-attention maps between text prompts and video content. This approach ensures that the original elements of the video are robustly maintained while effectively applying the intended modifications, through a consistent preservation control strategy without hyperparameter-tuning.

#### 3.1. Prompt Generation

We leverage large vision-language models (LVLMs) [29] to convert editing instructions into two descriptive prompts: one for the original video and another for the edited video. This is essential because most text-to-video diffusion models require prompts that describe the video content itself. By generating these tailored prompts, our framework ensures that the underlying diffusion model can effectively perform instruction-guided editing while maintaining the structure and integrity of the original video content.

#### 3.2. Original Preservation Control

To preserve the original video content during editing, V<sup>2</sup>Edit employs three complementary control mechanisms: (1) controlling the initial noise to maintain low-frequency information; (2) regulating the noise added at each denoising step to preserve semantic details; and (3) utilizing cross-attention maps to ensure alignment between text prompts and video content. These mechanisms work synergistically to maintain the integrity of the original video while enabling effective edits, ensuring successful progression across various editing tasks. A visualization of our preservation control method is shown in Fig. 3.

**Basic Formulation.** We utilize a diffusion model with  $T$  noise-adding steps to generate videos in formats such

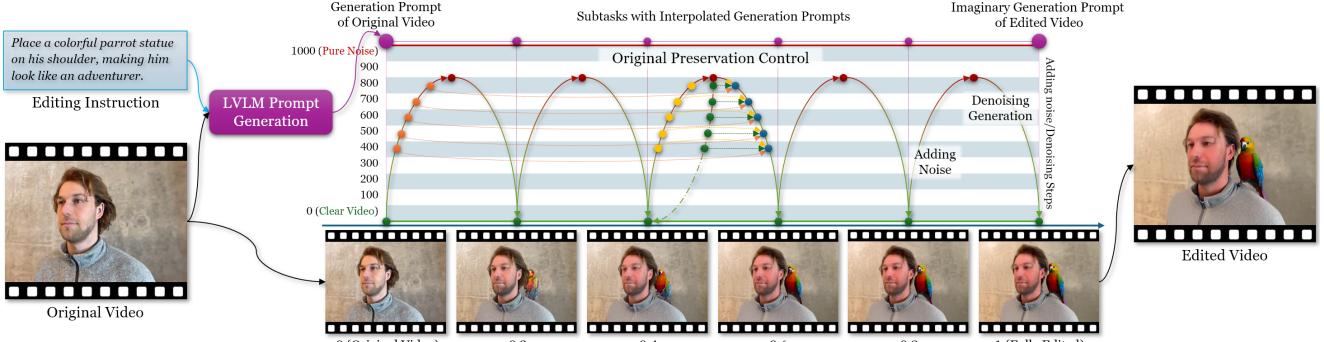


Figure 2. **Our V<sup>2</sup>Edit framework** features progressive editing. Given an editing instruction and the original video, a large vision-language model (LVLM) [29] generates prompts for both the original and edited videos. These prompts are interpolated to create a sequence of subtasks, which are executed progressively in our framework.

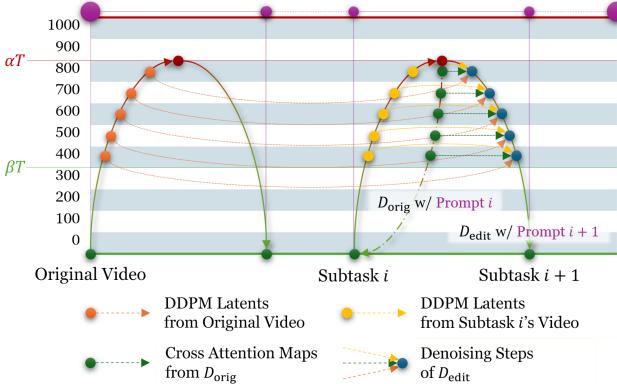


Figure 3. **V<sup>2</sup>Edit preservation control** integrates three key synergistic methods to preserve the original content during editing: (i) control of the initial noise ( $\alpha T$ ), (ii) management of noise added at each denoising step ('DDPM Latents'), and (iii) utilization of cross-attention maps between text prompts and video content. Each generation receives guidance on preservation from the previous subtask and the original video for a smooth progression.

as RGB or latent representations, referred to as “noisy videos” for generality. The noise-adding steps are denoted as  $A_1, A_2, \dots, A_T$ , and the denoising steps as  $D_T, D_{T-1}, \dots, D_1$ . Each denoising step  $D_i$  involves a denoising network and a noise scheduler. Let  $v_i$  represent the video after the  $i$ -th noise-adding step, where  $v_0$  is the original video and  $v_T$  is pure Gaussian noise. Formally, we define  $v_i = A_i(v_{i-1})$  and  $v_{i-1} = D_i(v_i)$  for  $i = 1, 2, \dots, T$ .

**Initial Noise Control.** To preserve the original video’s overall layout during editing, V<sup>2</sup>Edit controls the initial noise in the diffusion process. Inspired by SDEdit [27], instead of starting generation from pure Gaussian noise with  $T$  denoising steps, we limit noise addition to the first  $\alpha T$  steps and then perform denoising from this controlled noise with  $\alpha T$  denoising steps. This approach maintains low-frequency information, such as the video’s structure and layout, while allowing higher-frequency details to be destroyed and regenerated.

**Per-Step Noise Control.** The method above inspires us that the noise a diffusion model uses also carries semantic infor-

mation. Building on this observation, V<sup>2</sup>Edit leverages the noise added at each denoising step to preserve the original video content. Specifically, we utilize DDPM inverse [16] to extract DDPM latents  $n_1, n_2, \dots, n_{\alpha T}$  from the original video during the initial  $\alpha T$  noise-adding steps, which are the noises to be added at each step in a DDPM denoising procedure. These latents encapsulate rich semantic details essential for maintaining the video’s integrity. By applying them in the corresponding denoising steps  $D_{\alpha T}, \dots, D_{\beta T}$ , we ensure that semantic information is preserved without excessively constraining high-frequency details, allowing for effective and smooth edits.

However, the DDPM scheduler is inefficient for practical applications. To address this, we explore the intrinsic properties of DDPM inverse, which involves constructing noisy videos and solving for the precise noise required to denoise each step. By defining the denoising function as  $D_i(v_i | n_i) \stackrel{\text{Def}}{=} D_i(v_i) + n_i$  for schedulers that do not require random noise  $n_i$ , we adapt our preservation control to more advanced and efficient schedulers like DDIM [37] and DPMSolver++ [25, 26]. This novel adaptation allows V<sup>2</sup>Edit to benefit from the semantic preservation capabilities of DDPM inverse, while leveraging the high efficiency of advanced denoising schedulers.

**Cross-Attention Maps for Generation Control.** To further ensure the preservation of the original video’s semantic content, V<sup>2</sup>Edit manipulates cross-attention maps in the noise predictor model to align the edited scene’s generation process with the original scene. Inspired by attention map replacement strategies of prompt-to-prompt [11], our approach involves simultaneously performing two generations:  $D_{\text{orig}}$ , which generates the original video using the prompt of the original video; and  $D_{\text{edit}}$ , which generates the edited video using the prompt of the edited video. By controlling both generations concurrently, we can maintain a consistent semantic alignment between the original and edited content.

To address the high memory and computational cost of naively storing and replacing attention maps, we adopt a

fast and memory-efficient approach inspired by Flash Attention [8]. This technique allows us to compute the attention outputs for both  $D_{\text{orig}}$  and  $D_{\text{edit}}$  simultaneously during each cross-attention computation through the denoising process, enabling *real-time* replacement of cross-attention maps without the need to store them separately. By further combining the techniques in Flash Attention, we even eliminate the explicit construction of the cross-attention map and directly compute the final outputs. As a result, our method reduces the memory complexity from quadratic to constant, and achieves a fourfold speedup compared to explicit attention map storage and replacement. This efficient implementation ensures that V<sup>2</sup>Edit can seamlessly integrate cross-attention map controls within the progressive editing framework, maintaining high-quality and semantically consistent video edits without compromising performance.

**Synergy Between Control Mechanisms.** While the three preservation control mechanisms – initial noise control, intermediate noise control, and cross-attention map control – can be applied independently, their combination significantly enhances editing performance. By defining an interval  $[\beta T, \alpha T]$  during the denoising steps, our framework enables all three controls to operate synergistically *within* this interval. *Outside* of this interval, low-frequency details are preserved, and high-frequency textures are refined, ensuring both content preservation and high-quality edits.

The video editing procedure is shown in Fig. 3: First, we perform the initial  $\alpha T$  noise-adding steps to obtain the noisy video  $v_{\alpha T}$ . Then, we simultaneously generate two versions of the video,  $D_{\text{orig}}$  using the original prompt and  $D_{\text{edit}}$  using the editing prompt. During denoising steps between  $\beta T$  and  $\alpha T$ , the DDPM latents and cross-attention maps from  $D_{\text{orig}}$  guide the generation of  $D_{\text{edit}}$ . For denoising steps before  $\beta T$ , the model freely refines the video textures without additional guidance.

### 3.3. Progression-Based Editing Process

Different editing tasks may require different levels of preservation control. A mild and easy editing task can succeed with either a lower or higher level of preservation control, but a more challenging editing task that significantly changes the appearance may fail when the preservation control is too strict. To address the varying preservation control requirements across different editing tasks, V<sup>2</sup>Edit adopts a progression-based strategy, which decomposes a complex editing task into a sequence of simpler subtasks. As each decomposed subtask is mild and easy to achieve the trade-off between original content preservation and editing task fulfillment, this decomposition allows us to apply a *consistent* preservation control strategy across all subtasks without the need for task-specific adjustments.

As shown in Figs. 2 and 3, during the progression, for each subtask, V<sup>2</sup>Edit simultaneously performs *two* gener-

Method	CTIDS $\uparrow$	CDC $\uparrow$	GPT Score $\uparrow$	User Study $\uparrow$
Video Editing				
BIVDiff [35]	0.0755	0.1007	73.95	2.43
Instruct 4D-to-4D [28]	0.0502	0.0069	75.09	2.33
VideoShop [9]	0.0489	0.0967	60.00	2.29
Slicedit [7]	0.2867	0.1332	74.04	2.17
CSD [18]	0.1708	0.0534	48.48	2.14
V <sup>2</sup> Edit (Ours)	<b>0.3098</b>	<b>0.1388</b>	<b>84.50</b>	<b>3.83</b>
3D Scene Editing				
Instruct-NeRF2NeRF [10]	0.0542	0.1468	55.81	2.30
Efficient-NeRF2NeRF [38]	0.0367	0.1389	56.48	2.12
V2Edit [?]	0.1048	0.1150	48.65	2.42
V <sup>2</sup> Edit (Ours), Edited Scene	<b>0.2081</b>	<b>0.1716</b>	<b>88.97</b>	<b>3.90</b>

Table 1. Quantitative evaluation shows that our V<sup>2</sup>Edit consistently outperforms all the baselines under all metrics in both video and 3D scene editing tasks.

ations:  $D_{\text{orig}}$ , which regenerates the current subtask video using the original prompt, and  $D_{\text{edit}}$ , which generates the next subtask video using the editing prompt. The generation of  $D_{\text{edit}}$  is guided by the cross-attention maps and DDPM latents extracted from both  $D_{\text{orig}}$  and the original video generation (“ $D_{\text{orig}|\text{video}}$ ”) with a mixture coefficient.

By progressively completing each subtask with this dual-guided generation, V<sup>2</sup>Edit maintains high-quality and semantically consistent edits across various scenarios. This synergistic approach effectively balances the preservation of original content with the fulfillment of editing instructions, ensuring smooth and successful progression from one subtask to the next without the complexity of designing different levels of control mechanisms.

### 3.4. Efficient and Stable 3D Scene Editing

Beyond its native video editing capabilities, V<sup>2</sup>Edit seamlessly extends to 3D scene editing by incorporating a straightforward *render-edit-reconstruct (RER)* process: Render a video of the original scene along a fixed camera trajectory, perform video editing using V<sup>2</sup>Edit, and then reconstruct and re-render the scene from the edited video.

To ensure 3D consistency, we modify the progressive editing framework so that after obtaining the edited video for each subtask, we could reconstruct it to 3D and re-render it back to video for the next subtask. This modification leverages both the temporal smoothness of the rendered video and the 3D consistency of reconstruction, ensuring strong 3D consistency in the edited video. Unlike previous 3D editing methods that require iterative dataset updates and additional training, our approach remains stable and efficient, enabling high-quality edits with minimal diffusion generations. Furthermore, the temporal consistency of our edited videos allows for significant geometric changes, such as object insertion, which were previously challenging due to inconsistent per-view editing results.

## 4. Experiment

### 4.1. Experimental Settings

**V<sup>2</sup>Edit Settings.** We use CogVideoX-5b [46] as the underlying video diffusion model, which is a text-to-video model



Figure 4. Our V<sup>2</sup>Edit achieves successful editing results in various video editing tasks with superior overall appearance, while well preserving the original contents. The baselines either generate results with strange appearance and artifacts, or fail to preserve the areas unrelated to the editing. Notably, CogVideoX-V2V [46], an official video-to-video editing model of CogVideoX, generates good-looking results but is unable to preserve the original contents, showing that the key of our V<sup>2</sup>Edit lies in our novel progression framework and preservation control mechanism, instead of the strong underlying CogVideoX model. [More results are on our project website](#).

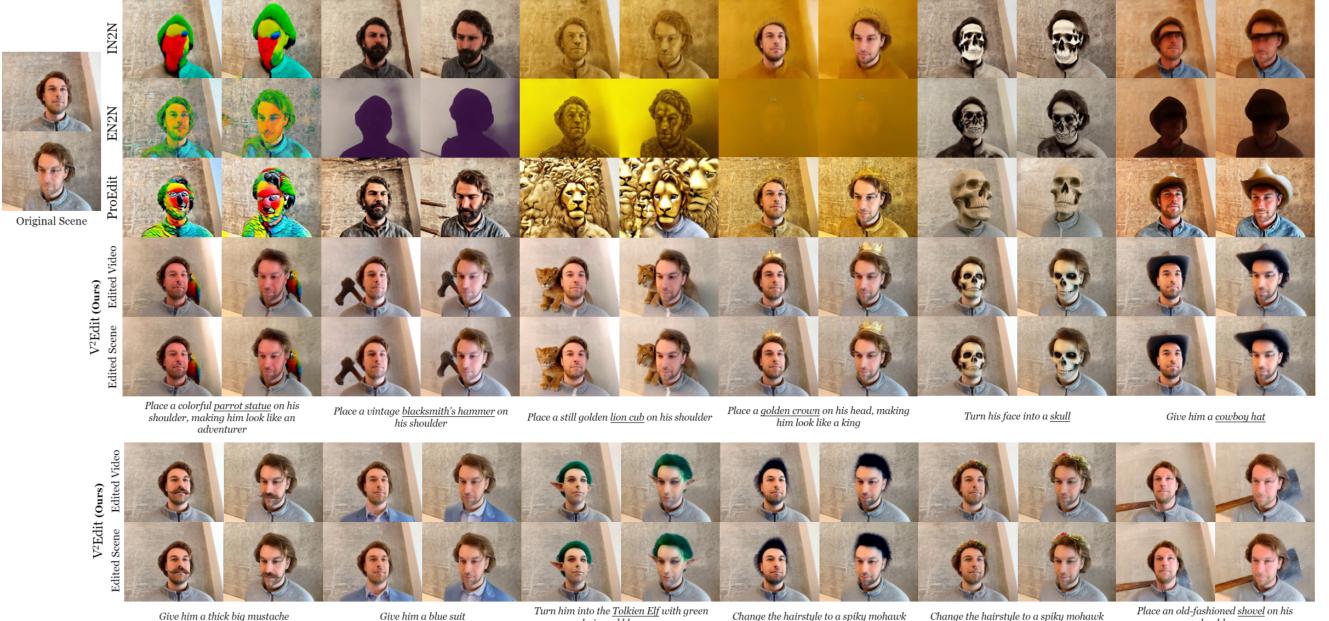


Figure 5. Our V<sup>2</sup>Edit achieves high-quality editing results in various challenging 3D scene editing tasks in the Face scene of the IN2N [10] dataset, with clear texture and geometry structure, bright color, and superior original content preservation. Notably, our V<sup>2</sup>Edit successfully performs editing operations with significant geometric changes like object insertion. On the contrary, the baselines either fail to perform the editing or do not preserve the contents in the original scene, *e.g.*, the background color, the appearance of the person, *etc.*

based of a diffusion transformer (DiT), and supports SORA-like [24] long descriptions as input prompts. We use GPT-4o [29] as the LVM to generate the prompts for underlying CogVideoX. For the subtask decomposition in our progres-

sive framework, we allow at most six (6) subtasks for each editing task. For 3D scene editing tasks, our V<sup>2</sup>Edit is independent of the specific scene representation. Therefore, we choose either SplactFacto or NeRFFacto from NeRFStudio



Figure 6. Our V<sup>2</sup>Edit achieves high-quality editing results across various indoor and outdoor scenes, while consistently achieving both editing task fulfillment and original content preservation for all the tasks. The baselines either fail to perform the editing or edit many unrelated areas without satisfactory preservation. **More results are on our project website.**

[39] as our scene representation.

**Video Editing Tasks.** Consistent with previous work [35], we use the videos from DAVIS dataset [30, 43] as the source videos. The editing tasks for evaluation are suggested by GPT-4o, given the original video as input.

**Video Editing Baselines.** We compare our V<sup>2</sup>Edit with video editing baselines, which can be roughly divided into two categories: (1) Image-based methods which rely on an underlying image generative model, including Slicedit [7], and Instruct 4D-to-4D [28] for monocular scenes; and (2) Video-based methods with utilizes an underlying video generative model, including CogVideoX-V2V [46], VideoShop [9], StableV2V [23], AnyV2V [20], BIVDiff [35] with per-frame editing and overall refinement, and CSD [18]. Some image-based methods require the edited first frame as a guidance, and we consistently apply Instruct-Pix2Pix [3] to generate this frame.

**3D Scene Editing Tasks.** Consistent with previous scene editing methods [6, 10, 40? ], we mainly use the scenes in Instruct-NeRF2NeRF (IN2N) [10] dataset for comparison evaluations. We also use several outdoor scenes from NeRFStudio [39] to serve as more challenging tasks. For the camera trajectory of the scene, we either use the existing trajectories (for IN2N dataset with officially provided trajectories) or manually draw one (for other scenes).

**3D Scene Editing Baselines.** We compare our V<sup>2</sup>Edit

with state-of-the-art traditional image-based 3D scene editing methods, including Instruct-NeRF2NeRF (IN2N) [10], Efficient-NeRF2NeRF [38], and V2Edit [? ]. In the **supplementary**, we also compare another type of baseline: applying the RER strategy (Sec. 3.4) with the video editing baselines mentioned above.

**V<sup>2</sup>Edit Variants for Ablation Study.** In the main paper, we provide the ablation study with the following key V<sup>2</sup>Edit variants: (1) CogVideoX-V2V, which also utilizes CogVideoX [46] as the underlying video diffusion model; (2) No Progression (NP), which only applies our original preservation control to the editing without progression. Due to limited space, we provide more ablation study results in the **supplementary** with more variants.

**Metrics.** The evaluation of video editing tasks contains many aspects, including the overall visual quality, the original video preservation, and the editing task fulfillment. It is challenging to evaluate them using traditional methods. Therefore, consistent with [? ], we use GPT-4o [29] for this evaluation, which can be regarded as a Monte Carlo simulation of the VQAScore [22]. We provide GPT with the requirements of each aspect, the editing instruction, and the original and edited videos frame-by-frame, and then ask GPT to provide a score from 1 to 100 for each aspect. To compare multiple videos between ours and different baselines, we provide all these videos simultaneously to GPT,

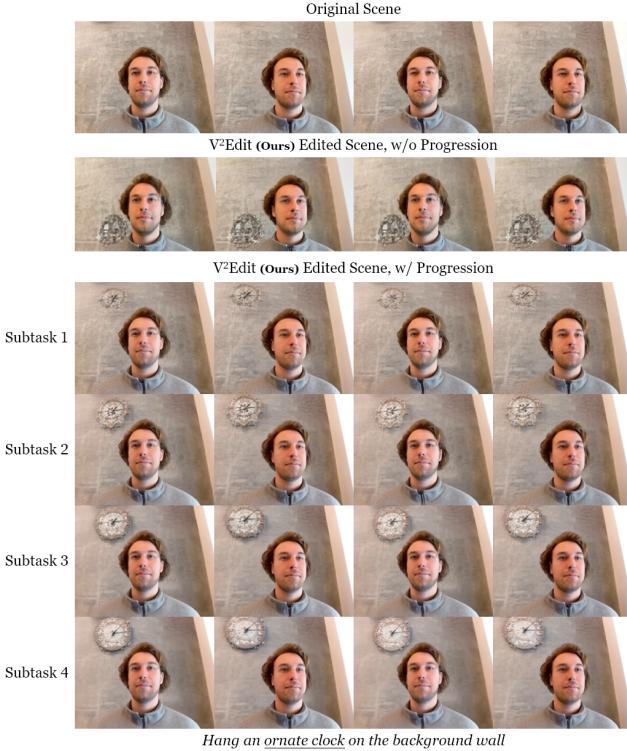


Figure 7. Ablation study on our progressive framework shows that progression is crucial to obtain high-quality editing results. Notably, the progressive editing process demonstrates a gradual procedure of editing to construct the clock on the wall, ending up with a clock with even 3D-consistent clock hands.

and ask GPT to score them all together to enforce a consistent scoring rule. To avoid randomness, we use the average of 20 independent evaluations as the final results. Leveraging the vision-language reasoning capability of GPT, this metric can quantify different aspects of the edited video. We also provide user study and the CLIP [33]-based scores from [10]: CLIP Text-Image Direction Similarity (CTIDS), and CLIP Direction Consistency (CDC).

## 4.2. Experimental Results

**Video Editing.** The visualization results of video editing on DAVIS [30] dataset are in Fig. 4, while more results are on [our project website](#). Our V<sup>2</sup>Edit consistently edits successfully and produce high-fidelity results in various challenging tasks, *e.g.*, adding a fiery ring for the motorcyclist to drive through, and turn a fast-moving person into a Batman; while successfully preserving the unrelated part, *e.g.*, the wall and layout of the tennis court and the tennis player’s motion in “Batman” task, the objects in the farm in “pig” task, and the river in the “swan” task. On the contrary, each baseline either fails to perform the editing or is unable to preserve the unrelated part from the original scene – especially the original pose and motion. Notably, the baseline CogVideoX-V2V is an official method that applies SDEdit [27] on CogVideoX, which can be regarded as a variant of

ours. This baseline produces videos with good appearance, but fails to preserve most of the information from the original scene. This validates the cruciality of our preservation control method. This shows that it is not the strong capability of the underlying CogVideoX we use, but our novel original preservation and progression pipeline that leads to our high-quality editing results.

**3D Scene Editing.** The results of 3D scene editing are shown in Figs. 5 and 6, and more results are on [our project website](#). As shown in Fig. 5, our V<sup>2</sup>Edit succeeds in challenging editing tasks that contain significant geometric change, with clear appearance and reasonable geometry structure, especially in the “lion cub” editing. *e.g.*, object insertion, while all the baseline fails to perform most these tasks – either unable to fulfill the editing requirement or completely changes the appearance of the original scene, or both. Despite of the face-forward scene, our V<sup>2</sup>Edit also performs well in the indoor or outdoor scenes in Fig. 6 in diversified editing instructions, with both great fulfillment of editing instruction and preservation of the original scene. Notably, with our self-implemented flash-attention-based [8] acceleration in Sec. 3.2, editing a 72-frame video only takes 10 minutes for each subtask in the progressive framework. Therefore, one editing task with at most six progression subtasks only takes roughly one to two hours to perform, achieving comparable efficiency as simple baselines [10, 38] but producing significantly superior results.

**Quantitative Evaluations.** We perform quantitative evaluations on several representative editing tasks, with results presented in Tab. 1, including a user study involving 43 participants conducted to assess subjective quality. Our V<sup>2</sup>Edit consistently outperforms all baseline methods across all metrics in both video and 3D scene editing. Specifically, V<sup>2</sup>Edit successfully balances original content preservation, as measured by the ‘CDC’ metric, which quantifies adjacent-frame similarity between the original and edited scenes; and editing task fulfillment, as demonstrated by GPT-based evaluations and user study results. These findings establish V<sup>2</sup>Edit as a state-of-the-art framework in both video and 3D scene editing domains.

**Ablation Study.** As illustrated in Fig. 4, the baseline CogVideoX-V2V generates high-quality videos across various editing tasks but consistently fails to preserve unrelated content from the original video. This baseline effectively represents a variant of our V<sup>2</sup>Edit with only initial noise control. These results indicate that leveraging a powerful video diffusion model alone is insufficient for high-quality editing without an effective content preservation mechanism, underscoring the necessity of our preservation control strategy. Additionally, as shown in Fig. 7, directly applying our content preservation mechanism without the progression framework results in failures in complex tasks, such as adding a clock. In contrast, when incorpo-

rating the progression-based editing strategy, V<sup>2</sup>Edit successfully constructs and refines the clock, achieving high-quality results. Notably, the clock hands remain consistent across all views, demonstrating excellent 3D consistency. These experiments validate that both our content preservation mechanism and progression framework are essential, which not only ensure content preservation but also achieve editing task fulfillment. Further ablation study results are provided in the **supplementary**.

## 5. Conclusion

In this paper, we introduce V<sup>2</sup>Edit, a novel and versatile framework for instruction-guided video and 3D scene editing. Our approach effectively balances the preservation of original content with the fulfillment of editing instructions by progressively decomposing a complex task into simpler subtasks, managed by a unified preservation control mechanism. For video editing, V<sup>2</sup>Edit excels in handling challenging scenarios involving fast-moving camera trajectory, complex motions, and significant temporal variations, ensuring smooth and consistent edits. For 3D scene editing, our framework supports challenging editing tasks with substantial geometric changes, while maintaining high 3D consistency and sufficiently preserving the original scene content. Extensive experiments demonstrate that V<sup>2</sup>Edit achieves state-of-the-art performance in both video and 3D scene editing. We hope that V<sup>2</sup>Edit paves the way for future advancements in video and 3D scene editing using video diffusion models.

## References

- [1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. [1](#), [2](#)
- [2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023. [1](#), [2](#)
- [3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Learning to follow image editing instructions. In *CVPR*, 2023. [7](#)
- [4] Jun-Kun Chen and Yu-Xiong Wang. ProEdit: Simple progression is all you need for high-quality 3D scene editing. In *NeurIPS*, 2024. [1](#), [2](#), [3](#)
- [5] Jun-Kun Chen, Samuel Rota Bulò, Norman Müller, Lorenzo Porzi, Peter Kortschieder, and Yu-Xiong Wang. ConsistDreamer: 3D-consistent 2D diffusion for high-fidelity scene editing. In *CVPR*, 2024. [3](#)
- [6] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and controllable 3D editing with Gaussian splatting. In *CVPR*, 2024. [7](#)
- [7] Nathaniel Cohen, Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Slicedit: Zero-shot video editing with text-to-image diffusion models using spatio-temporal slices, 2024. [1](#), [2](#), [3](#), [5](#), [7](#)
- [8] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. [5](#), [8](#), [1](#)
- [9] Xiang Fan, Anand Bhattacharjee, and Ranjay Krishna. Videoshop: Localized semantic video editing with noise-extrapolated diffusion inversion, 2024. [1](#), [2](#), [3](#), [5](#), [7](#)
- [10] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3D scenes with instructions. In *ICCV*, 2023. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [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, 2022. [4](#)
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. [1](#)
- [13] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022. [2](#)
- [14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. [2](#)
- [15] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. [1](#), [3](#)
- [16] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly DDPM noise space: Inversion and manipulations. In *CVPR*, 2024. [1](#), [4](#)
- [17] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Styling video by example. *ACM Trans. Graph.*, 38(4), 2019. [1](#), [3](#)
- [18] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn2, and Jinwoo Shin1. Collaborative score distillation for consistent visual editing. In *NeurIPS*, 2023. [2](#), [3](#), [5](#), [7](#)
- [19] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, 2024. [3](#)
- [20] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhua Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. [7](#)
- [21] Yao-Chih Lee, Ji-Ze Genevieve Jang Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing demo. *arXiv preprint arXiv:2301.13173*, 2023. [1](#), [3](#)
- [22] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024. [7](#)
- [23] Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. StableV2V: Stabilizing Shape Consistency in Video-to-Video Editing, 2024. [7](#)

- [24] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024. 1, 3, 6
- [25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps, 2022. 4
- [26] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. 4
- [27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Ji-jun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 1, 3, 4, 8
- [28] 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. 1, 2, 3, 5, 7
- [29] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogneni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Rei-ichiyo Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Toootchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Valalone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3, 4, 6, 7, 5
- [30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 7, 8
- [31] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 3
- [32] 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:2303.09535*, 2023. 1, 3
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 8
- [34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

- [35] Fengyuan Shi, Jiaxi Gu, Hang Xu, Songcen Xu, Wei Zhang, and Limin Wang. Bivdiff: A training-free framework for general-purpose video synthesis via bridging image and video diffusion models, 2024. 1, 2, 3, 5, 7
- [36] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 2
- [37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. 4
- [38] Liangchen Song, Liangliang Cao, Jiatao Gu, Yifan Jiang, Junsong Yuan, and Hao Tang. Efficient-NeRF2NeRF: Streamlining text-driven 3D editing with multiview correspondence-enhanced diffusion models, 2023. 2, 3, 5, 7, 8
- [39] 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 *SIGGRAPH*, 2023. 7
- [40] Cyrus Vachha and Ayaan Haque. Instruct-GS2GS: Editing 3D Gaussian splats with instructions, 2024. 7
- [41] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitrii Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel multi-view synthesis and 3D generation from a single image using latent video diffusion. In *ECCV*, 2024. 1, 2
- [42] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, 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, 2023. 3
- [43] Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, Rui He, Feng Hu, Junhua Hu, Hai Huang, Hanyu Zhu, Xu Cheng, Jie Tang, Mike Zheng Shou, Kurt Keutzer, and Forrest Iandola. Cvpr 2023 text guided video editing competition, 2023. 7
- [44] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency, 2024. 2
- [45] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation, 2022. 2
- [46] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazhen Xu, Yuanming Yang, Wenqi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3, 5, 6, 7, 1, 2

# V<sup>2</sup>Edit: Versatile Video Diffusion Editor for Videos and 3D Scenes

## Supplementary Material

This document contains additional analysis and extra experiments.

### A. Additional Experimental Results

#### A.1. Additional Visualizations

We show additional visualization results for video editing in Fig. H, and for 3D scene editing in Fig. I. Our V<sup>2</sup>Edit framework consistently outperforms baselines in both editing task fulfillment and original content preservation in all these tasks.

#### A.2. Video Editing Baselines for 3D Scene Editing

The qualitative results of video editing baselines for 3D scene editing are on our project website, and the quantitative results are in Tab. B. Our V<sup>2</sup>Edit significantly outperforms all these methods, especially in the GPT score, which evaluates the overall quality. Also, as shown in the videos on our project website, even when the videos edited by these baselines have reasonable appearance, the 3D reconstruction is still low-quality and contains lots of artifacts. This shows that our V<sup>2</sup>Edit is the only model that has sufficient editing capability and preservation control ability to edit 3D scenes.

#### A.3. Ablation Study

We conduct our ablation study on each of our core strategies: Progression ('Pro'), initial noise control ('INC'), per-step noise control ('PNC'), and attention-map replacement ('AMR').

The ablation study results are in Fig. J and on our project website. The quantitative measurements are in Tab. C. We observe that all these core strategies are crucial to our final results. More specifically:

- Progression is crucial to the success and clear appearance of the final results. Without progression, some geometry editing tasks for 3D scenes may even fail.
- Per-step noise control (PNC) is the most important control in our original preservation framework. Without PNC, the edited video will be significantly different from the original view, which implies a failure in original preservation.
- Initial noise control is crucial to the preservation of the overall color of the edited video.
- Attention-map replacement is crucial for the preservation of the shape and appearance of unrelated objects.
- The variant "w/o PNC, AMR" is equivalent to CogVideoX-V2V [46] with progression. The failure of this variant shows that CogVideoX-V2V cannot succeed in editing tasks even with progression, further validating

that the high-quality results of our V<sup>2</sup>Edit are not from the powerful underlying CogVideoX but our novel strategies.

### B. Implementation Details

#### B.1. Settings and Hyperparameters

**GPUs.** All of our experiments are run on NVIDIA A6000, A100, and H100 GPUs. Each instance of editing task only needs one GPU.

**Underlying Video Diffusion Model.** We use CogVideoX-5b [46] as our underlying diffusion model, and fetch the pre-trained weights from HuggingFace. The model is run in pure Brain Float16 ("bf16") data type. We consistently set the classifier-free guidance scale [12] as 7.0, and use CogVideoX's default CogVideoXDPMScheduler.

**Hyperparameters.** We choose the hyperparameters for our original preservation control method as follows. We set  $\alpha = 0.9, \beta = 0.5$ . As the  $T$  (number of denoising steps) of CogVideoX is 1000, this means that we first perform the first  $\alpha T = 900$  noise addition steps and then start denoising at this generation, and apply the guidance from DDPM inverse and the cross-attention map replacement until the  $\beta T = 500$ -th denoising step.

#### B.2. Flash Attention-Based Optimization for Attention Map Replacement

In dot-product attention

$$\text{Attn}(Q, K, V) = \text{SoftMax}\left(QK^\top/\sqrt{d}\right)V,$$

where  $Q$  is  $n \times d$ ,  $K$  is  $m \times d$ , and  $V$  is  $m \times d'$ , the most expensive operation is to explicitly construct  $M = (QK^\top/\sqrt{d})$ , a.k.a. "the attention map", which is  $n \times m$ . However, the SoftMax operation does not allow us to directly compute the output. The key insight of flash attention [8] optimization is as follows: consider each row of SoftMax( $M$ ), we have

$$\text{SoftMax}(M)_{i,j} = \frac{\exp(M_{i,j})}{\sum_j \exp(M_{i,j})}.$$

Therefore, we can first compute the denominator  $d_i = \sum_j \exp(M_{i,j})$  for each row  $i$ , and then directly calculate  $\text{SoftMax}(M)_{i,j} = \exp(M_{i,j})/d_i$ . To avoid the precision issue of  $\exp(M_{i,j})$  when  $M_{i,j}$  is large, we first compute



Figure H. **In additional video editing tasks, our V<sup>2</sup>Edit framework** consistently outperforms baselines in both editing task fulfillment and original content preservation.

Method	CTIDS $\uparrow$	CDC $\uparrow$	GPT Score $\uparrow$
Edited Video (Intermediate Results of 3D Scene Editing)			
BIVDiff [35]	0.1656	0.0813	51.00
Instruct 4D-to-4D [28]	0.0224	0.1723	29.60
VideoShop [9]	0.0068	-0.0389	21.60
Slicedit [7]	-0.0064	0.1817	31.00
CSD [18]	-0.0035	0.1930	30.60
CogVideoX-V2V [46]	0.2125	0.0554	67.80
<b>V<sup>2</sup>Edit (Ours)</b>	<b>0.2796</b>	<b>0.1934</b>	<b>90.20</b>
Rendered Edited Scenes			
BIVDiff [35]	0.0901	0.0691	44.00
Instruct 4D-to-4D [28]	-0.0117	0.1507	25.20
VideoShop [9]	0.0045	0.0271	14.20
Slicedit [7]	-0.0061	0.1728	24.00
CSD [18]	0.0035	0.1950	23.60
CogVideoX-V2V [46]	0.2042	0.0266	52.80
<b>V<sup>2</sup>Edit (Ours)</b>	<b>0.2817</b>	<b>0.2012</b>	<b>90.60</b>

Table B. Quantitative evaluation shows that our V<sup>2</sup>Edit significantly outperforms all the baselines that use video editing methods for 3D scene editing. This validates that our V<sup>2</sup>Edit is uniquely capable of performing 3D scene editing among video-based methods.

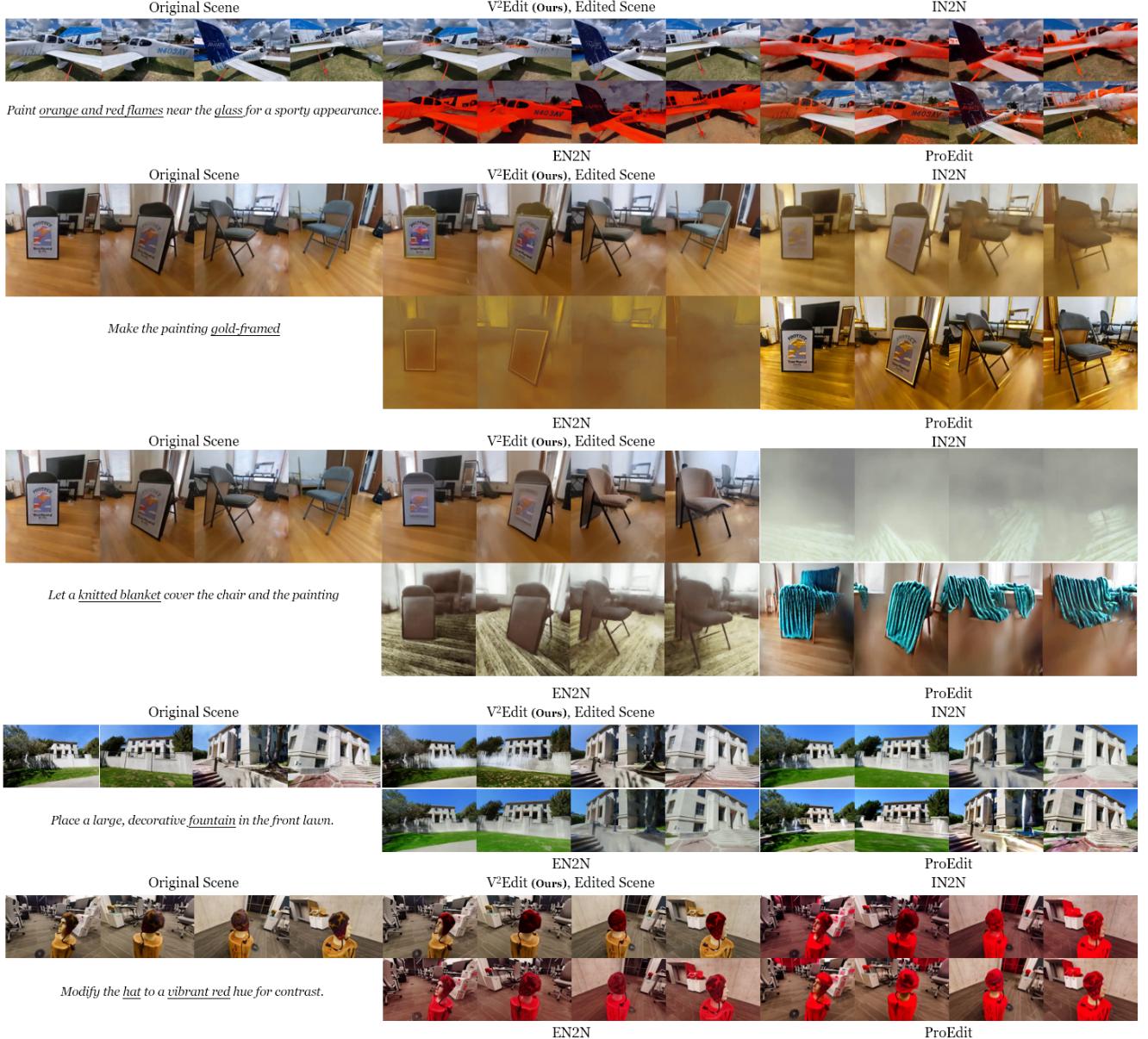
$m_i = \max_j M_{i,j}$ , and then define  $d'_i = \sum_j \exp(M_{i,j} - m_i)$  to take the place of  $d_i$ , so that

$$\text{SoftMax}(M)_{i,j} = \exp(M_{i,j} - m_i) / d'_i.$$

At last, the final output

$$\text{Attn}(Q, K, V)_i = (\text{SoftMax}(M)V)_i$$

$$= \sum_j \text{SoftMax}(M)_{i,j} v_j \\ = \sum_j \exp(M_{i,j} - m_i) v_j / d'_i$$



**Figure I. In additional 3D scene editing tasks, our **V<sup>2</sup>Edit** framework** consistently outperforms baselines in both editing task fulfillment and original content preservation.

All the elements  $M_{i,j}$ ,  $m_i$ ,  $d'_i$  can be computed at the same time complexity but without explicitly constructing the whole matrix  $M$ , which significantly saves the memory cost by reducing the additional memory complexity from  $O(nm)$  to  $O(n)$ , and the time to allocate and access the memory.

When we use attention map replacement, the only difference is that for the computation of  $M$ , some columns (corresponding to  $K, V$ , which is the prompt token sequences) need to be replaced from another  $M'$ . In this case, we perform the cross-attention of both operations in parallel, and

replace the attention map *on-the-fly*. We define

$$\text{AttnAMR}(Q^{(1)}, K^{(1)}, V^{(1)}, Q^{(2)}, K^{(2)}, V^{(2)}, I^{(1)}, I^{(2)})$$

as such function, where  $\text{Attn1} = (Q^{(1)}, K^{(1)}, V^{(1)})$  is the attention operation to be computed as usual, and  $\text{Attn2} = (Q^{(2)}, K^{(2)}, V^{(2)})$  is the attention operation to be computed with attention-map replacement (AMR),  $I^{(1)}, I^{(2)}$  is the index list that the  $I_k^{(2)}$ -th column of attention map  $M^{(2)}$  of Attn2 should be replaced with the  $I_k^{(1)}$ -th column of attention map  $M^{(1)}$  of Attn1 for each  $1 \leq k \leq |I^{(1)}| = |I^{(2)}|$ . In the computation, we only need to simply replace the



Figure J. Ablation studies show that all our strategies are crucial to successful and high-quality editing results. Removing any of them results in significant degradation. The video visualizations are on our project website.

Method	CTIDS ↑	CDC ↑	GPT Score ↑
Video Editing			
w/o Progression (Pro)	0.0611	0.2414	65.23
w/o Initial Noise Control (INC)	0.0763	0.1922	64.80
w/o Per-Step Noise Control (PNC)	0.0949	0.0291	55.83
w/o Attention-Map Replacement (AMR)	0.0898	0.1079	51.04
w/o PNC, AMR	0.0674	0.0394	40.21
<b>Full V<sup>2</sup>Edit (Ours)</b>	<b>0.1029</b>	<b>0.2933</b>	<b>76.60</b>
3D Scene Editing			
w/o Progression (Pro)	0.0571	0.2744	59.17
w/o Initial Noise Control (INC)	0.0655	0.2462	52.50
w/o Per-Step Noise Control (PNC)	0.0789	0.0088	34.17
w/o Attention-Map Replacement (AMR)	0.0829	0.1109	53.33
w/o PNC, AMR	0.0631	0.0370	41.67
<b>Full V<sup>2</sup>Edit (Ours)</b>	<b>0.0954</b>	<b>0.3658</b>	<b>77.50</b>

Table C. Ablation study shows that all our strategies are crucial to our final performance.

computation of  $M^{(2)}$  to fit the rule of attention map replacement, *i.e.*, calculate  $M_{i,j}^{(2)}$  with  $Q_i^{(1)}K_{j'}^{(1)\top}$  instead of  $Q_i^{(2)}K_j^{(2)\top}$  if  $j = I_k^{(2)}, j' = I_k^{(1)}$  for some  $k$ . In this way, we extend the optimization of flash attention to the attention with attention-map replacement.

In our experiments, this optimization could bring a 2 ~ 4 times speed-up.

### B.3. Attention Control for Long and Looping Videos

Most of the video diffusion models are trained to generate fixed-length videos. For example, CogVideoX is trained to generate 49-frame videos at 8 FPS. However, some videos

to be edited are longer than this length, especially in 3D editing tasks. If we speed up the video to fit the frames, the camera movement will be too fast and, therefore, introduce many challenges for the editing operation to be successful. To address this, we propose a way to enable the support of long video in a training-free manner.

One naive way to generate a long video with a diffusion model is just to extend the size of the input noise, which will not encounter out-of-memory issue with our flash attention-based optimization. However, as the model is only trained on 49-frame videos, it gets confused with the long-term temporal positional embeddings, which are unseen in the training dataset and may lead to low-quality videos. In the

generation, this happens in the self-attention, where both  $Q$  and  $K$  correspond to video patch tokens, and the temporal embedding of  $Q_i$  and  $K_j$  are too far away. Also, as the  $K$  is much larger than the trained situation, the attention also aggregates too much elements from  $K$  and further lead to blurred results.

Our insight is that, we can constraint the set of video tokens  $\{K_j\}$  that can be seen by each video token  $Q_i$ , so that it will only encounter the patterns of positional embeddings that seen in training, and also not see and aggregate too many  $K_j$  to make the video blurred. More specifically, if the  $Q_i$  belongs to frame  $k$ , we allow  $Q_i$  to see all the tokens within frames  $[k-l/2, k+l/2]$ , where  $l$  is the length of the pre-trained videos. With this control, each  $Q_i$  will only see the tokens within nearby  $l$  frames, which are seen patterns in training, and also not aggregating too much  $\{K_j\}$ s – which will be no more than  $l$  frames, *i.e.* the number of  $\{K_j\}$ s seen in training. With this optimization, the model is able to generate high-quality long videos in a training-free manner.

In 3D scene editing, some of the camera trajectories are looping, *i.e.*, the first frame continues the last frame. In this case, it is desirable to also make the edited video looping. Therefore, when generating the first several frames, *e.g.* frame  $i$  where  $i < l/2$ , we allow it to see the last several frames  $L + i - l/2 < j \leq L$ , where  $L$  is the total length of the current video, and apply the temporal positional embedding of  $j - L$  to frame  $j$ , to make it look like one of the previous frames of  $i$  to the model. In this case, the model learns to generate looping edited videos at few additional computation cost, and significantly improves the consistency.

#### B.4. LVLM-Based Prompt Generation

**LVLM Used for Generation.** We use GPT-4o [29] as the LVLM model to generate the prompts for video diffusion models.

**The Prompts for LVLM.** We use the following template of prompts for the LVLM:

- **System:** You are part of a team of bots that edits videos. These videos are 360 degree videos for still persons, objects or scenes, so we can use a single frame of these videos to describe the main content of the video. You work with an assistant bot that will draw anything you say in square brackets. Firstly, you will get several images and their descriptions. You should learn these reference prompts' style. And then you will get several images and several pairs of prompts that before editing instruction and after editing instruction. You should learn how to edit the prompt based on the instruction and make the after editing prompt as similar as possible with the before editing prompt but should reflect the meaning of the

instruction. At last, you should provide the prompt before editing and the prompt after editing. Remember two things: source prompt is the prompt before editing and target prompt is the prompt after editing.

There are a few rules to follow: When you need to edit the prompt, you should make the target prompt as similar as possible with the source prompt but should reflect the meaning of the instruction. The source prompt and target prompt should have similar style with the reference prompt.

- **User:** What's the video's prompt?  
**Attachment:** Several frames of an astronaut's video.
- **Assistant (Mimic Reply):** A suited astronaut, with the red dust of Mars clinging to their boots, reaches out to shake hands with an alien being, their skin a shimmering blue, under the pink-tinged sky of the fourth planet. In the background, a sleek silver rocket, a beacon of human ingenuity, stands tall, its engines powered down, as the two representatives of different worlds exchange a historic greeting amidst the desolate beauty of the Martian landscape.
- (Several examples like this in similar format)
- **User:** Instruction: make him a clown with green hair.  
**Attachment:** Several frames of a video of a human's face.
- **Assistant (Mimic Reply):**  
Before Edit: A person with light golden brown hair, normal face with clear forehead, and a normal nose and a light beard, dressed in a gray zip-up jacket, stands calmly against a minimalist concrete wall. The camera starts from a fixed position, then slowly rotates around the person, capturing their profile and facial expressions as the perspective shifts smoothly, with the plain, textured background highlighting the subject.  
After Edit: A clown with green hair, painted white face, and a red nose and colorful facial designs, dressed in a gray zip-up jacket, stands calmly against a minimalist concrete wall. The camera starts from a fixed position, then slowly rotates around the person, capturing their profile and facial expressions as the perspective shifts smoothly, with the plain, textured background highlighting the subject.  
item (Several examples like this in similar format)
- **User:** Instruction: {The actual instruction to query}.  
**Attachment:** Several frames of {the actual video to query}.
- **Assistant (Actual Query to GPT):** {The actual generated pairs of prompts}.

**Examples.** Here are some examples of editing tasks shown in our main paper.

- Instruction: Place a colorful parrot statue on his shoulder, making him look like an adventurer.

Before Edit: A person with light golden brown hair, normal face with clear forehead, and a normal nose and a light beard, dressed in a gray zip-up jacket, stands calmly against a minimalist concrete wall. The camera captures a slight profile, enhancing the depth and dimension of the scene.

After Edit: An adventurer with light golden brown hair, normal face, and a normal nose and a light beard, dressed in a gray zip-up jacket, stands calmly against a minimalist concrete wall, with a colorful parrot statue perched on his shoulder. The scene captures a slight profile, enhancing the sense of adventure.

- Instruction: Turn the stone bear into a realistic piglet statue.

Before Edit: A stone bear sculpture is perched on a large, flat rock, surrounded by lush greenery and tall trees. The bear’s rough texture and simplistic carving reflect a naturalistic style, capturing the essence of the forest environment, while the light filters through the leaves, casting dappled shadows on the statue.

After Edit: A realistic piglet statue with pink skin, soft pig ears, and an adorable appearance is perched on a large, flat rock, surrounded by lush greenery and tall trees. The piglet’s smooth texture and detailed carving highlight its domestic charm, capturing the essence of a small farm-yard animal, while the light filters through the leaves, casting dappled shadows on the statue.

- Instruction: Decorate the top edge of the stump with tiny, glowing fairy lights.

Before Edit: A mannequin head wearing a colorful knitted hat is placed on top of a wooden stump in an office setting. The room features several office chairs, desks with monitors, and boxes, all under white fluorescent lights, with a concrete wall in the background.

After Edit: A mannequin head wearing a colorful knitted hat is placed on top of a wooden stump adorned with tiny, glowing fairy lights around the top edge in an office setting. The room features several office chairs, desks with monitors, and boxes, all under white fluorescent lights, with a concrete wall in the background.

## C. Discussion

### C.1. Limitations

The limitations of our V<sup>2</sup>Edit mainly lie in these two aspects: the editing capability from the underlying video diffusion model, and the 3D awareness and motion constraint in 3D editing. Note that most of the video-based editing methods also face these challenges.

**Editing Capability.** As a general framework, our V<sup>2</sup>Edit is compatible with various underlying video diffusion models. When paired with a specific model, *e.g.*, CogVideoX

[46], our editing capability relies on that model and is, therefore, constrained by its limitation. For example, as CogVideoX is trained exclusively on  $720 \times 480$  landscape videos, our V<sup>2</sup>Edit using CogVideoX supports only landscape videos, not portrait videos. Also, if a diffusion model cannot recognize a specific concept, such as an object or a type of motion, we cannot perform effective editing related to that concept. Similarly, if the model cannot generate a plausible video for a prompt, we cannot produce a reasonable edited video by generating w.r.t. the corresponding editing prompt with our preservation control. Adopting a more advanced video diffusion model could help mitigate this limitation.

**3D Editing: 3D Awareness and Motion Constraint.** In 3D editing, the edited video should ideally represent a rendered video of a 3D scene, maintaining 3D-consistency and remaining static. However, since the video diffusion model lacks 3D input, it is not 3D-aware. As a result, the edited video may not be 3D-consistent. On the other hand, the edited video may contain some motion, which is not desired for 3D scene editing. Both limitations can be mitigated by our progression-based generation procedure. At each subtask, by reconstructing the edited video into the 3D scene and then re-rendering the video, we enforce the 3D consistency and ensure the video remains static at this step. By decomposing the process into subtasks with minimal modifications, such inconsistency and unwanted motion are reduced to minor levels, which can be corrected during reconstruction and re-rendering.

### C.2. Overcoming Limitations of Image-Based Editing Methods

Despite the limitations mentioned above, our video-based V<sup>2</sup>Edit framework successfully overcomes the limitations of image-based (per-frame/per-view) editing methods. While the image-based methods require numerous iterations (as seen in the “iterative dataset update” in [10]) to achieve convergence, our V<sup>2</sup>Edit can directly produce edited videos based on an end-to-end framework without time-consuming iterations. In 3D scene editing, our V<sup>2</sup>Edit leverages video diffusion models to its advantage. First, the generated video is smooth and temporally consistent, which is a strong prerequisite for achieving 3D consistency. This allows us to directly reconstruct the video into 3D, eliminating the need for iterative processes and enabling large-scale shape editing. Additionally, by analyzing the entire video, which contains a complete view of scene objects, our V<sup>2</sup>Edit avoids common issues like Janus or multi-face artifacts that plague image-based editing methods. It also supports view-dependent effects, such as specular effects generated by the video diffusion model, and successfully reconstructs them in 3D. By overcoming these challenges,

our V<sup>2</sup>Edit produces state-of-the-art results in both video and 3D scene editing.

### C.3. Future Directions

**4D Scene Editing.** While a video represents “dynamic 2D” and a 3D scene represents “static 3D,” one interesting direction is to extend this work to 4D, encompassing “dynamic 3D.” This would involve complex editing tasks such as motion editing, object moving, *etc.*

**3D-Aware Video-Based Scene Editing.** Another direction is to make the video diffusion model 3D-aware for 3D scene editing. This could be achieved by training a new video diffusion model on RGBD videos and integrating it with our V<sup>2</sup>Edit. Doing so would enable 3D awareness in the model and potentially make it easier to control the video content static.