HDEdit: Editing Videos and 3D Scenes with Video Diffusion Through Hierarchical Task Decomposition

Yanming Zhang[1,2†] Jun-Kun Chen[1†] Jipeng Lyu[1] Yu-Xiong Wang[1]
[1]University of Illinois Urbana-Champaign [2]University of Maryland [†]Equal Contribution
{junkun3,jipeng2,yxw}@illinois.edu yanmingz@umd.edu

immortalco.github.io/HDEdit

Figure 1. Our **HDEdit** supports *training-free* instruction-guided editing for both videos and 3D scenes through **H**ierarchical task **D**ecomposition. **Left:** HDEdit achieves high-quality editing satisfying both *original content preservation* and *editing instruction fulfillment* in video editing. **Right:** HDEdit supports challenging 3D scene editing tasks involving *significant geometric changes*, which baselines [4, 11] fail to achieve. More results are provided in our Project Page and Supplementary Video.

Abstract

*We introduce HDEdit, a training-free framework for instruction-guided video and 3D scene editing that resolves the fundamental tension between instruction fulfillment and original content preservation through **H**ierarchical task **D**ecomposition. Our key insight is to progressively decompose complex edits into simpler subtasks. This hierarchical strategy aligns with dual objectives: an LLM-guided planner structures high-level subgoals for reliable instruction fulfillment, while embedding-space interpolation further refines each subgoal to preserve unedited content. Two tailored control mechanisms – word-level attention map propagation and parallel denoising synchronization – ensure temporally consistent, hyperparameter tuning-free execution. Beyond video, we extend HDEdit to 3D editing via a simple yet effective render-edit-reconstruct process that maintains strong geometric consistency. Extensive experiments demonstrate our state-of-the-art results across diverse and challenging edits, including long-duration videos, fast camera motion, and significant 3D geometric changes.*

1. Introduction

Video diffusion models have recently emerged as a powerful class of generative models capable of synthesizing high-resolution, high-fidelity videos from text prompts [1, 2, 15, 26, 41], extending the transformative success of image diffusion models [35] into the temporal domain. Alongside generation, a particularly promising direction is *instruction-guided video editing*, which involves modifying existing videos through natural language descriptions. This paradigm enables the creation of new videos through minimal edits, offering greater efficiency and control than generation from scratch.

Despite their potential, instruction-guided video editing remains significantly underdeveloped. A key reason is the lack of large-scale paired datasets for supervised learning, which makes training-based methods difficult to scale. Consequently, training-free approaches have gained traction, leveraging pre-trained diffusion models to perform editing without additional optimization [10, 16, 27, 36]. Yet, these methods struggle with three fundamental challenges: maintaining temporal coherence, handling dynamic content with

fast camera motion or object movement, and, critically, *balancing the fulfillment of editing instructions with the preservation of original content*. Moreover, they typically require careful tuning of hyperparameters to strike this balance, limiting usability and robustness.

We observe that this balance is especially difficult to achieve in *complex edits* as illustrated in Fig. 1, which demand both precise control over multiple modifications and reliable retention of unedited regions. Our key insight is to treat such tasks as inherently **multi-stage problems**. Specifically, we decompose the global objective – editing with preservation – into two entangled but asymmetrical subgoals: (1) **editing fulfillment**, ensuring the desired modification is fully expressed; and (2) **original content preservation**, ensuring unrelated regions remain unchanged.

To achieve both goals, we propose HDEdit, a novel *training-free* framework that introduces a ***H**ierarchical task **D**ecomposition* strategy, explicitly aligning the *dual-stage* decomposition with the twofold objectives of instruction-guided editing: *fulfilling intended edits* and *preserving original content*.

At the *high level*, we address the challenge of editing fulfillment by leveraging large language models (LLMs) for semantic task planning. Given a complex instruction, the LLM analyzes its intent and visual implications, and then schedules a sequence of simpler, semantically meaningful sub-instructions. Each sub-instruction targets a distinct editing goal that is easier to achieve in isolation – for example, modifying a specific object, attribute, or region – thus enabling progressive, interpretable fulfillment of the original instruction. This decomposition not only structures the editing process but also mitigates the risk of over-editing or semantic drift that often arises from attempting to satisfy complex prompts in a single step.

However, even when each subgoal is semantically clear and localized, executing it through a single diffusion step can still compromise content outside the target region. To mitigate this, we introduce a *low-level decomposition* mechanism focused on the preservation aspect. Specifically, we apply *embedding-space interpolation* to further decompose each high-level subgoal into a series of intermediate representations. These steps form a smooth transition path that gradually transforms the video content, reducing semantic discontinuities and minimizing visual artifacts. In effect, this fine-grained decomposition eases preservation control within each subtask.

We further propose two synergistic strategies to achieve preservation control across the hierarchically decomposed subtasks. First, a *word-level attention map management* selectively retains visual features from prior subgoals by storing and reusing the attention maps from previous subtasks, enabling spatially-aware content inheritance across steps. Second, a *parallel denoising control scheme* coordinates the diffusion process over time, ensuring temporal coherence and consistent appearance preservation. These components form a unified editing system that is both *hyperparameter tuning-free* and highly reliable in preserving content integrity while achieving faithful edits.

Beyond its core application to video editing, HDEdit naturally extends to 3D scene editing. We introduce a simple yet effective *render-edit-reconstruct (RER)* process that renders a video from a 3D scene, edits the video using our framework, and reconstructs the scene from the edited result. This process preserves 3D consistency by leveraging the temporal coherence of the rendered video and benefits from the same hierarchical decomposition used in 2D. Notably, HDEdit can robustly edit scenes with significant geometric changes and large-scale camera motion – scenarios that existing methods [7, 10, 19, 29, 36] fail to handle.

As demonstrated in Fig. 1, HDEdit achieves state-of-the-art editing quality across a wide range of challenging scenarios, including long-duration videos, rapid camera movement, and fine-grained scene edits. In 3D, it supports geometry-altering edits such as object insertion, providing superior geometric consistency and flexibility while outperforming prior approaches [4, 11, 38] that rely on costly view-specific optimization.

Our contributions are threefold:

- We propose HDEdit, a training-free framework for instruction-guided video editing through hierarchical task decomposition, and further extend it to 3D scene editing via a simple yet effective render-edit-reconstruct process.
- We introduce a hierarchical subtask decomposition aligned with editing goals: high-level LLM-based scheduling for instruction fulfillment and low-level interpolation-based refinement for content preservation.
- We design two robust preservation control strategies – word-level attention map propagation and parallel denoising synchronization – enabling consistent, hyperparameter tuning-free editing.

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, 37, 46] 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, 37]. Later, Stable Video Diffusion (SVD) [1] scales up 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 [45] contents. The release of SORA [26] has lit a new way to scale up video diffusion models with diffusion transformers (DiTs). Instead of applying down-sampling 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 [47] is proposed upon its previous effort CogVideo [15], using DiT-based video diffusion models and significantly improving the video length, resolution, and generation quality. Hunyuan [20] further scales up the model capacity, producing even more superior results.

Video Editing. Due to the lack of paired training data for video editing, *i.e.*, triples of "editing instruction, original video, and edited video," *most existing methods operate in a training-free manner.* Traditional video editing methods [7, 17, 23, 29, 33] are image-based methods, which rely on an underlying model with image editing capability and introduce other add-ons to control the consistency. For example, FateZero [33], Tune-A-Video [43], and Instruct 4D-to-4D [29] adapt 2D diffusion models for video editing by extending the spatial attention layers in UNets into spatio-temporal attention layers in a zero-shot manner, incorporating both the first and previous frames during generation.

Following the emergence of video diffusion models, several methods have explored video editing by utilizing their generative power and temporal smoothness capabilities. BIVDiff [36] uses a pre-trained video diffusion model to refine temporally inconsistent, per-frame edited images into a smooth, temporally consistent video. VideoShop [10], AnyV2V [22], and StableV2V [25] take the edited first frame as input, along with the original video, and propagates the edits through subsequent frames. CogVideoX-V2V [47] applies SDEdit [27] to edit videos by using the generative capability. However, unlike our HDEdit, these methods struggle with complex scenarios, such as fast-moving cameras, dynamic backgrounds and contents, and significant changes in geometry or motion. Notably, many training-free methods also require task-specific hyperparameter tuning to balance instruction fulfillment and original content preservation, whereas our *hyperparameter tuning-free* HDEdit achieves a consistent preservation strategy across diverse editing tasks.

Diffusion-Based 3D Scene Editing. In instruction-guided 3D scene editing, a common approach is to apply 2D diffusion models to individual views and distill the resulting edits into the 3D scene using score distillation sampling (SDS) [32]. Instruct-NeRF2NeRF [11] pioneers this direction by employing an SDS-equivalent iterative update mechanism to refine a dataset of edited views to train the NeRF [28] representation towards the edited scene. Subsequent work has aimed to improve various aspects of this paradigm, such as editing efficiency [38], distillation quality [19, 21], 3D consistency [4, 5, 18, 42], and even extensions to 4D scenes [29]. On the other side, video diffusion models offer a natural alternative by directly editing a rendered video of the scene, replacing image-based diffusion with temporally-aware generation. Since temporal consistency in the edited video is essential for maintaining 3D consistency in the underlying scene, this approach has the potential to significantly reduce the challenge of maintaining geometric coherence. However, achieving this in practice is difficult: the rendered video must span diverse viewpoints and scene content, requiring substantial variation in both visual appearance and camera trajectory to sufficiently cover the full scene. These factors make the video editing task itself particularly challenging. To the best of our knowledge, no existing work has successfully leveraged video diffusion models for editing 3D or 4D scenes.

Task Decomposition in Visual Content Editing. While the idea of decomposing complex editing tasks into simpler, more manageable subtasks shows initial promise, it remains under-explored, particularly in the context of challenging video editing scenarios. ProEdit [4] introduces task decomposition and progression for 3D scene editing, which applies interpolation-based decomposition with a difficulty estimation based on perceptual metrics. In contrast, our hierarchical decomposition is guided by LLMs, which assess task difficulty based on their reasoning capabilities, leading to a more grounded and reliable subgoal generation. Such LLM-based approach not only simplifies the original editing instruction into high-level, interpretable subtasks, but also facilitates interpolation in more structured subspaces. Furthermore, the hierarchical structure enables more effective and principled content preservation control strategies. Empirically, our HDEdit outperforms ProEdit with stronger editing capability from hierarchical decomposition, extending the success of task progression to video editing.

3. Methodology

HDEdit is a *training-free* framework that leverages pre-trained video diffusion models for controllable, high-fidelity video editing. As illustrated in Fig. 2, our core design is a *hierarchical subtask decomposition* that reflects the dual objectives of instruction-guided editing: high-level scheduling simplifies *instruction fulfillment*, and low-level refinement controls *content preservation* of the original scene. Specifically, an LLM-guided semantic-aware planner first decomposes the original editing instruction into a sequence of high-level subtasks (Fig. 2-(a)), each annotated with an estimated difficulty score. Each high-level subtask is then further split, via interpolation-based decompo-

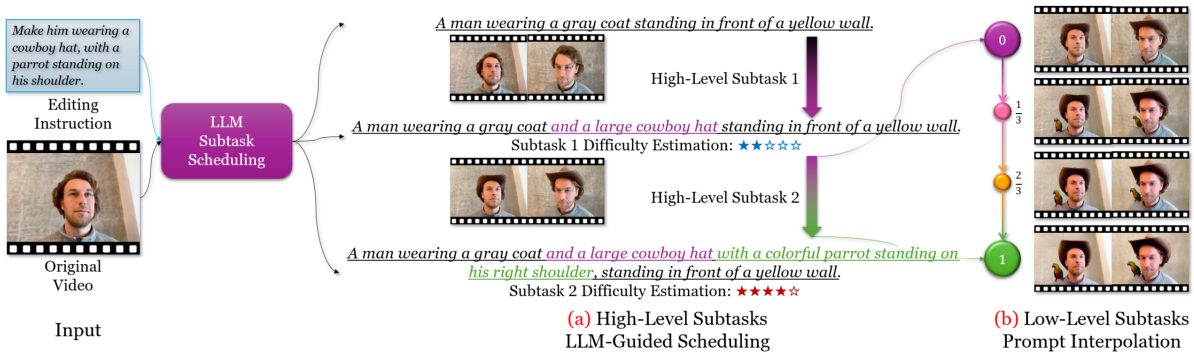

Figure 2. **Our HDEdit framework** features progressive editing via hierarchical subtask decomposition. Given an editing instruction and an original video, we first apply (a) an LLM-guided semantic-aware scheduling to decompose the task into high-level subtasks represented as video prompts, with difficulty estimation; and then (b) an interpolation-based low-level decomposition to adaptively further break down each high-level subtask into finer sub-word granularity, informed by the difficulty.

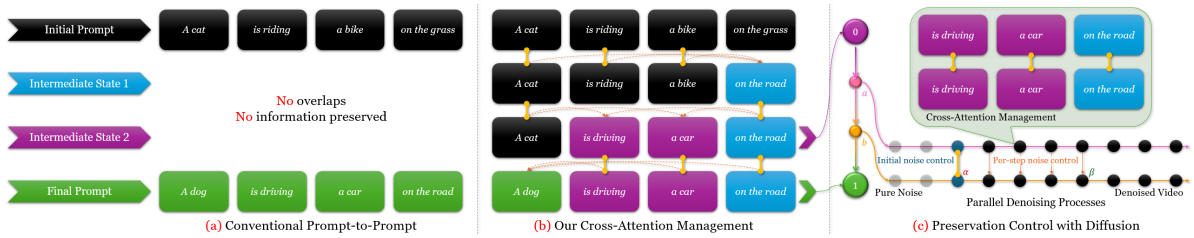

Figure 3. **Our original content preservation control mechanism in HDEdit** tailored for our hierarchical subtask decomposition. Compared to (a) directly applying conventional Prompt-to-prompt (P2P) [12] with two prompts, which can be highly ineffective, our (b) cross-attention management leverages the overlaps between adjacent high-level subtasks to effectively preserve the original content. For each low-level subtask, (c) our parallel denoising synchronization handles preservation at both high-level and low-level tasks, facilitating smooth progression through the subtasks.

sition (Fig. 2-(b)), into a smooth series of low-level modifications. To preserve relevant content from the source video throughout this progressive process, we introduce a *training-free preservation-control strategy* (Fig. 3) tailored to the hierarchy: (1) word-level attention map propagation between adjacent high-level states; and (2) a parallel denoising mechanism that enforces consistency during low-level execution. Together, these components deliver interpretable, high-quality edits for complex tasks without any manual hyperparameter tuning.

3.1. Hierarchical Subtask Decomposition

To decompose the editing task into subtasks with simpler structures and comparable difficulty, we propose a hierarchical subtask decomposition framework with (1) LLM-guided semantic-aware scheduling for high-level subtask planning, and (2) interpolation-based difficulty-informed refinement for low-level subtask generation.

Definition of States and Subtasks. Unlike the "editing instruction" or "editing prompt" which define the overall editing goal, the video diffusion model accepts only descriptive prompts as input. Therefore, we define a *state* in the video editing process as a descriptive prompt such that the corresponding video either matches or can be generated from it. The initial state s_0 describes the original video, and the final state s_n describes the desired edited video. Intermediate states $\{s_i\}$ may exist between them. We then define a *subtask* as the transformation from one state (prompt) s_{i-1} to the next state s_i. For simplicity, we refer to the subtask associated with state s_i as the transformation from s_{i-1} to s_i, and denote the resulting video of this subtask as v_i.

LLM-Guided Semantic-Aware Subtask Scheduling (Fig. 2-(a)). We observe that LLMs like [30] possess strong visual reasoning capability: given a few frames from the original video, an LLM can generate a descriptive prompt s_0; similarly, with s_0 and an editing instruction, it can infer the final prompt s_n. We thus leverage this ability for high-level subtask scheduling by prompting the LLM to generate a sequence of intermediate states $s_1, s_2, \cdots, s_{n-1}$, ensuring that each state s_i is visually similar to its adjacent states s_{i-1} and s_{i+1}. Additionally, we ask the LLM to estimate subtask difficulty based on the visual change between s_{i-1} and s_i. This approach enables interpretable, semantically meaningful scheduling while providing grounded and robust difficulty estimates – more reliable than those in [4] – which facilitate low-level decomposition. A concrete example is in Suppl. Sec. C.6.

Interpolation-Based Difficulty-Informed Subtask Decomposition (Fig. 2-(b)). The high-level scheduling provides *word-level* subtask granularity. However, some subtasks may still exhibit disproportionately high difficulty. For example, transitioning from s_0 = "a person" to s_1 =

"a person with a beard" is relatively easy, but s_1 to s_2 = "a person with a beard and a parrot standing on the shoulder" is significantly harder due to the complex geometry of the parrot. In such cases, decomposition at a finer level becomes necessary; however, inserting semantically meaningful intermediate prompts or states between s_1 and s_2 is non-trivial, especially for the LLM, since a parrot represents a semantically "atomic" concept. To this end, we introduce interpolation-based subtask decomposition inspired by [4], inserting intermediate states in the form $\alpha s_1 + (1-\alpha)s_2$ by interpolating their embeddings. This enables *sub-word level* granularity. We leverage LLM-derived difficulty estimates to determine the number of interpolation points, ensuring subtask-complexity-informed decomposition.

3.2. Original Content Preservation Control

After decomposing the editing task into subtasks, we execute each subtask s_i progressively, aiming to preserve as much information as possible from the previous state s_{i-1}. Building on our hierarchical decomposition, we design tailored preservation control strategies (Fig. 3). When executing subtask s_i, given the previous state s_{i-1} and the two adjacent high-level states s_l and s_r, where $l \leq i \leq r$, we perform: (1) attention map management to preserve high-level information from s_l, and (2) parallel denoising control to retain low-level details from s_{i-1}.

Attention Map Management (Fig. 3-(b)). Controlling attention maps is a common strategy for content preservation in editing. A representative method is Prompt-to-Prompt (P2P) [12], which reuses cross-attention maps of shared words to maintain consistency. However, as shown in Fig. 3-(a), directly applying P2P to s_0 and s_n may be ineffective for complex editing due to minimal prompt overlap. Our high-level decomposition provides intermediate states where adjacent prompts are more semantically similar and share more common words (Fig. 3-(b)). Leveraging this, we manage attention maps to exploit all available overlaps. Specifically, we store the attention maps from each word in the previous high-level state s_l, and directly replace the attention maps of the corresponding words during the diffusion generation of s_i (upper part of Fig. 3-(c)). This mechanism supports cross-subtask propagation, even for $i = r$, enabling attention maps to persist across multiple subtasks when words recur (Fig. 3-(b)). As a result, we achieve *long-term* content preservation: even if s_0 and s_n share no direct overlap, information is progressively transferred through adjacent subtask pairs, ensuring robust preservation control. More details are in Suppl. Secs. C.3-C.4.

Parallel Denoising Synchronization -(Fig. 3-(c)) . While attention map management effectively preserves information from s_l, which remains unchanged from s_l to s_i, we still need to retain additional information from s_{i-1}. However, since the prompts for s_l and s_r often differ significantly due to word substitutions, additions, or rephrasings, cross-attention map replacement becomes unreliable. To address this, we shift from preserving content based on *literal* prompt overlap to preserving it based on *visual appearance*, using parallel denoising processes. Specifically, we run two denoising procedures in parallel: (1) a fully controlled denoising generation process D_{i-1} that reconstructs s_{i-1}'s video v_{i-1}; and (2) a guided generation process D_i for s_i's video v_i, which receives and preserves information from D_{i-1}, *i.e.*, from v_{i-1}. To implement this, we adopt initial noise control from [27] and per-step noise control inspired by [16], within the denoising timestep range [α,β] (lower part of Fig. 3-(c)). Such a strategy not only ensures faithful reconstruction of v_{i-1} in D_{i-1}, but also transfers sufficient information from D_{i-1} to D_i, enabling robust content preservation across adjacent subtasks. More details are in Suppl. Sec. C.2.

3.3. Render-Edit-Reconstruct (RER) for 3D Scene Editing

Beyond its native video editing capabilities, HDEdit seamlessly extends to 3D scene editing via a straightforward *render-edit-reconstruct (RER)* process: render a video of the original scene along a fixed camera trajectory, edit the video using HDEdit, and then reconstruct and re-render the scene from the edited video.

To ensure 3D consistency, we modify the progressive editing framework such that, after obtaining the edited video v_i for each subtask s_i, we reconstruct it into 3D and re-render it back to *3D-consistent* video v_i^{3D}; all preservation control mechanisms then operate on the 3D-consistent v_{i-1}^{3D} and v_l^{3D} instead of v_{i-1} and v_l. This modification leverages both the temporal smoothness of rendered videos and the 3D consistency from reconstruction, ensuring strong 3D consistency in edited videos. Unlike previous 3D editing methods [4, 9, 11, 19, 38] that require iterative dataset updates and additional training, our approach is stable and efficient, achieving 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

HDEdit Settings. We utilize the open-source CogVideoX-5b [47] as the underlying video diffusion model. CogVideoX-5b is a text-to-video model based on a diffusion transformer (DiT), and supports SORA-like [26] long descriptions as input prompts. For LLM-guided planning, we employ GPT-4o [30] to generate s_0 and s_n, perform high-level task decomposition, and estimate subtask diffi-

culty. Based on this, we further decompose each high-level subtask into at most three low-level subtasks. For 3D scene editing tasks, our HDEdit is independent of the specific scene representation. Therefore, we adopt either Splact-Facto or NeRFacto from NeRFStudio [39] as the scene representation, depending on the scenario.

Video Editing Tasks. Consistent with previous work [36], we use videos from the DAVIS dataset [31, 44] as source videos. The editing tasks used for evaluation are suggested by GPT-4o, conditioned on the original video content.

Video Editing Baselines. We compare our HDEdit with several video editing baselines, which can be roughly divided into two categories: (1) Image-based methods that rely on an underlying image generative model, including Slicedit [7] and Instruct 4D-to-4D [29], primarily designed for monocular scenes; and (2) Video-based methods that utilize an underlying video generative model, including CogVideoX-V2V [47], VideoShop [10], StableV2V [25], AnyV2V [22], BIVDiff [36], and CSD [19]. These methods typically adopt either per-frame editing followed by overall refinement, or first-frame editing with propagation to subsequent frames. For baselines that require image editing, we consistently apply Instruct-Pix2Pix [3] to generate the corresponding frame. Notably, ProEdit [4] focuses exclusively on 3D scene editing and is not directly applicable to video editing; instead, we include a variant of HDEdit that mimics ProEdit's strategy in our ablation study for comparison.

3D Scene Editing Tasks. Consistent with previous scene editing methods [4, 6, 11, 40], we mainly use scenes from the Instruct-NeRF2NeRF (IN2N) [11] dataset for comparative evaluation. We also use several outdoor scenes from NeRFStudio [39] to serve as more challenging tasks. For camera trajectories, we use the official ones provided with the IN2N dataset or manually design them for other scenes.

3D Scene Editing Baselines. We compare our HDEdit with state-of-the-art image-based 3D scene editing methods, including Instruct-NeRF2NeRF (IN2N) [11], Efficient-NeRF2NeRF (EN2N) [38], and ProEdit [4]. In the **supplementary**, we also compare with another type of baseline: applying our RER strategy (Sec. 3.3) in combination with the video editing baselines mentioned above.

HDEdit Variants for Ablation Study. We conduct an ablation study on each core strategy: Decomposition and Progression ('Pro'), attention-map management ('AMM'), and the two strategies of parallel denoising control: initial noise control ('INC') and per-step noise control ('PNC').

Metrics. The evaluation of video editing tasks involves multiple aspects, including overall visual quality, preservation of the original video content, and fulfillment of the editing instruction. It is challenging to assess them using traditional metrics. Therefore, following ProEdit [4], we use GPT-4o [30] as an evaluator, which can be regarded as a Monte Carlo simulation of the VQAScore [24]. We provide GPT with a detailed prompt including the evaluation criteria, the editing instruction, and both the original and edited videos (frame-by-frame), and then ask GPT to assign a score from 1 to 100 for each aspect. To ensure consistency across comparisons, we present all edited videos from different methods (including ours and baselines) to GPT simultaneously, and ask it to score them in a single batch. This encourages the use of a uniform scoring standard. In addition to GPT-based evaluation, we report user study results and CLIP-based [34] scores introduced in [11], including CLIP Text-Image Direction Similarity (CTIDS) and CLIP Direction Consistency (CDC).

4.2. Experimental Results

Video Editing. Visualization results for video editing on the DAVIS [31] dataset are shown in Fig. 4, with additional examples provided in the Project Page and Suppl. Sec. D. Our HDEdit consistently edits successfully and produces high-fidelity results in various challenging tasks, *e.g.*, adding a fiery ring for the motorcyclist to drive through, or turning a fast-moving person into Batman; while successfully preserving unrelated content, *e.g.*, the wall and layout of the tennis court and the tennis player's motion in the "Batman" task, the background objects in the farm in the "pig" task, and the river in the "swan" task. On the contrary, all baselines either fail to complete the editing task or significantly alter unrelated parts from the original scene, especially the original pose and motion. Notably, the CogVideoX-V2V baseline is an official method that applies SDEdit [27] on CogVideoX, which can be regarded as a variant of our approach. This baseline produces videos with good appearance, but fails to preserve most original scene content. This highlights the importance of our preservation control mechanisms. Our results demonstrate that it is not merely the strength of the underlying CogVideoX model we use, but rather our novel task decomposition and progression framework and preservation control mechanisms that lead to our high-quality editing results.

3D Scene Editing. Results for 3D scene editing are shown in Figs. 5 and 6, with additional examples provided in the Project Page and Suppl. Sec. D. As illustrated in Fig. 6, our HDEdit succeeds in challenging editing tasks that contain significant geometric change, producing realistic appearance and coherent geometry, especially in the "lion cub" example (*e.g.*, object insertion). In contrast, all baselines fail on most of these tasks – either unable to fulfill the editing requirement or drastically altering the appearance of the original scene, or both. Beyond forward-facing scenes, our HDEdit also performs well in indoor and outdoor scenes in Fig. 5, handling diverse editing tasks while maintaining both faithful edits and strong preservation of the original scene content. Notably, with our flash-attention-based [8] acceleration (Suppl. Sec. C.4), editing a 72-frame video takes

Figure 4. Our HDEdit produces high-quality editing results in various video editing tasks, achieving superior visual appearance while effectively preserving original content. In contrast, the baselines often introduce visual artifacts, generate unrealistic appearances, or fail to retain regions unrelated to the editing. Notably, CogVideoX-V2V [47], the official video-to-video editing model of CogVideoX, generates visually appealing results but lacks content preservation. This highlights that the strength of HDEdit stems not from the underlying CogVideoX backbone, but from our novel task decomposition and progression framework and preservation control mechanisms. Please refer to Suppl. Sec. D for more results, and Project Page for the corresponding videos.

Figure 5. Our HDEdit achieves high-quality 3D editing results across various indoor and outdoor scenes, consistently fulfilling editing instructions while preserving original content for all the tasks. In contrast, the baselines either fail to complete the editing or alter many unrelated regions without adequate preservation. Please refer to Suppl. Sec. D for more results, and Project Page for the rendered videos.

only 10 minutes per subtask within the progression framework. Therefore, each editing task completes in roughly one to two hours, achieving efficiency comparable to simpler baselines [11, 38], while eliminating unstable iterative

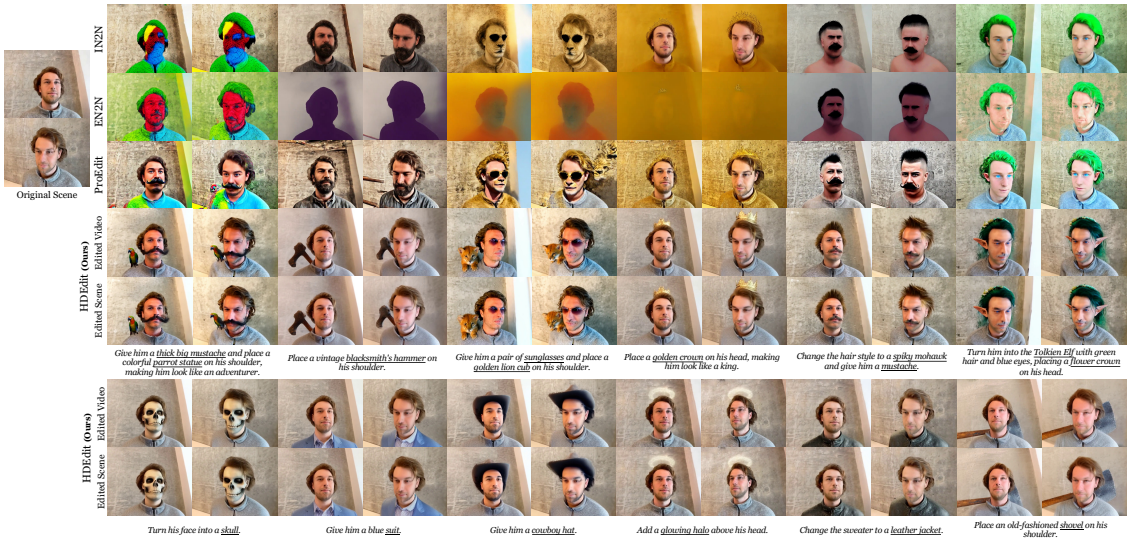

Figure 6. Our HDEdit achieves high-quality editing results in various challenging 3D scene editing tasks on the Face scene from the IN2N [11] dataset, demonstrating clear texture, coherent geometry, vivid color, and strong preservation of original content. Notably, HDEdit successfully handles editing operations involving significant geometric changes, like object insertion, with ease. In contrast, the baselines either fail to perform the desired editing or compromise key aspects of the original scene, such as background color or subject appearance. Please refer to Project Page for the rendered videos.

Method	CTIDS ↑	CDC ↑	GPT Score ↑	User Study ↑
Video Editing				
BIVDiff [36]	0.0755	0.1007	73.95	3.70
Instruct 4D-to-4D [29]	0.0502	0.0269	65.09	2.81
VideoShop [10]	0.0489	0.0967	60.00	3.03
Slicedit [7]	0.2867	0.1332	74.04	2.27
CSD [19]	0.1708	0.0534	48.48	2.22
CogVideoX-V2V [47]	0.2114	0.0452	75.00	3.86
StableV2V [25]	0.1506	0.0545	75.60	2.49
AnyV2V [22]	0.1891	0.0654	72.40	2.65
HDEdit **(Ours)**	**0.3098**	**0.1388**	**84.50**	**4.47**
3D Scene Editing				
Instruct-NeRF2NeRF [11]	0.2312	0.0397	30.4	2.14
Efficient-NeRF2NeRF [38]	0.1475	0.0329	20.0	2.25
ProEdit [4]	0.2464	0.0600	55.8	2.31
HDEdit **(Ours)**, Edited Scene	**0.2742**	**0.1474**	**88.6**	**4.51**

Table 1. Quantitative evaluation shows that our HDEdit consistently outperforms all the baselines under all metrics in both video and 3D scene editing tasks.

frame edits and delivering significantly superior results.

Quantitative Evaluations. We conduct quantitative evaluations on several representative editing tasks, with results presented in Tab. 1, including a user study involving 43 participants to assess subjective quality. Our HDEdit consistently outperforms all baseline methods across all metrics in both video and 3D scene editing. In particular, HDEdit achieves a strong balance between 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 both GPT-based evaluations and user study results. These findings establish HDEdit as a state-of-the-art framework for both video and 3D scene editing domains.

Ablation Study. The ablation study results are in Suppl. Tab. C, while the qualitative visualizations are on our Project Page and in Suppl. Sec. D. We observe that all these core strategies are crucial to our final results. More specifically: (1) Decomposition and 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. (2) Attention-map management is crucial for the preservation of the shape and appearance of unrelated objects. (3) Per-step noise control (PNC) is the most important control in the parallel denoising control method. Without PNC, the edited video will be significantly different from the original view, which implies a failure in original content preservation. (4) Initial noise control is crucial to the preservation of the overall color of the edited video. These show that all our designs are crucial to our high-quality results.

5. Conclusion

We presented HDEdit, a training-free framework for instruction-guided video and 3D scene editing through hierarchical task decomposition. By combining LLM-guided high-level planning with embedding-based low-level refinement, our method effectively balances instruction fulfillment and content preservation. Two complementary control strategies ensure consistent, high-quality editing without the need for hyperparameter tuning. Applied to both videos and 3D scenes via a render-edit-reconstruct process, HDEdit achieves state-of-the-art performance across complex and dynamic scenarios. We hope this unified and extensible framework will inspire future advances in generative video and scene editing.

Acknowledgments. This work was supported in part by NSF under Grants 2106825 and 2519216, the DARPA Young Faculty Award, the ONR Grant N00014-26-1-2099, and the NIFA Award 2020-67021-32799. This work used computational resources provided by the ACCESS program (CIS230012, CIS230013, and CIS240133) and the NAIRR Pilot.

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. *arXiv:2311.15127*, 2023. 1, 3

[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. *arXiv:2304.08818*, 2023. 1, 2, 3

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

[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, 4, 5, 6, 8

[5] Jun-Kun Chen, Samuel Rota Bulò, Norman Müller, Lorenzo Porzi, Peter Kontschieder, 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. 6

[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. *arXiv:2405.12211*, 2024. 2, 3, 6, 8

[8] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. *arXiv:2205.14135*, 2022. 6

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

[10] Xiang Fan, Anand Bhattad, and Ranjay Krishna. Videoshop: Localized semantic video editing with noise-extrapolated diffusion inversion. *arXiv:2403.14617*, 2024. 1, 2, 3, 6, 8

[11] 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

[12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv:2208.01626*, 2022. 4, 5

[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. *arXiv:2210.02303*, 2022. 2, 3

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv:2204.03458*, 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. *arXiv:2205.15868*, 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, 5

[17] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Stylizing video by example. *ACM Trans. Graph.*, 38 (4), 2019. 3

[18] Umar Khalid, Hasan Iqbal, Nazmul Karim, Jing Hua, and Chen Chen. LatentEditor: Text driven local editing of 3d scenes. *arXiv:2312.09313*, 2024. 3

[19] 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, 6, 8

[20] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv:2412.03603*, 2024. 3

[21] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, 2024. 3

[22] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *arXiv:2403.14468*, 2024. 3, 6, 8

[23] Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing demo. *arXiv:2301.13173*, 2023. 3

[24] 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. 6

[25] Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. StableV2V: Stablizing Shape Consistency in Video-to-Video Editing. *arXiv:2411.11045*, 2024. 3, 6, 8

[26] 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. *arXiv:2402.17177*, 2024. 1, 3, 5

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 1, 3, 5, 6

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

[29] 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. 2, 3, 6, 8

[30] 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 Gogineni, 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, Reiichiro 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 Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, 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:2303.08774*, 2023. 4, 5, 6

[31] 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. 6

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

[33] 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. 3

[34] 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. 6

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[36] 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. *arXiv:2312.02813*, 2024. 1, 2, 3, 6, 8

[37] 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. *arXiv:2209.14792*, 2022. 2, 3

[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. *arXiv:2312.08563*, 2023. 2, 3, 5, 6, 7, 8

[39] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 6

[40] Cyrus Vachha and Ayaan Haque. Instruct-GS2GS: Editing 3D Gaussian splats with instructions, 2024. 6

[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, 3

[42] Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3D editing with gaussian splatting. *arXiv:2403.11868*, 2025. 3

[43] 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. *arXiv:2212.11565*, 2023. 3

[44] 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. *arXiv:2310.16003*, 2023. 6

[45] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3D content generation with multi-frame and multi-view consistency. *arXiv:2407.17470*, 2024. 3

[46] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv:2203.09481*, 2022. 2

[47] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv:2408.06072*, 2024. 3, 5, 6, 7, 8