UniEdit: A Unified Tuning-Free Framework for Video Motion and Appearance Editing

Anonymous Author(s)
Affiliation
Address
`email`

Project webpage: https://uni-edit.github.io/UniEdit/

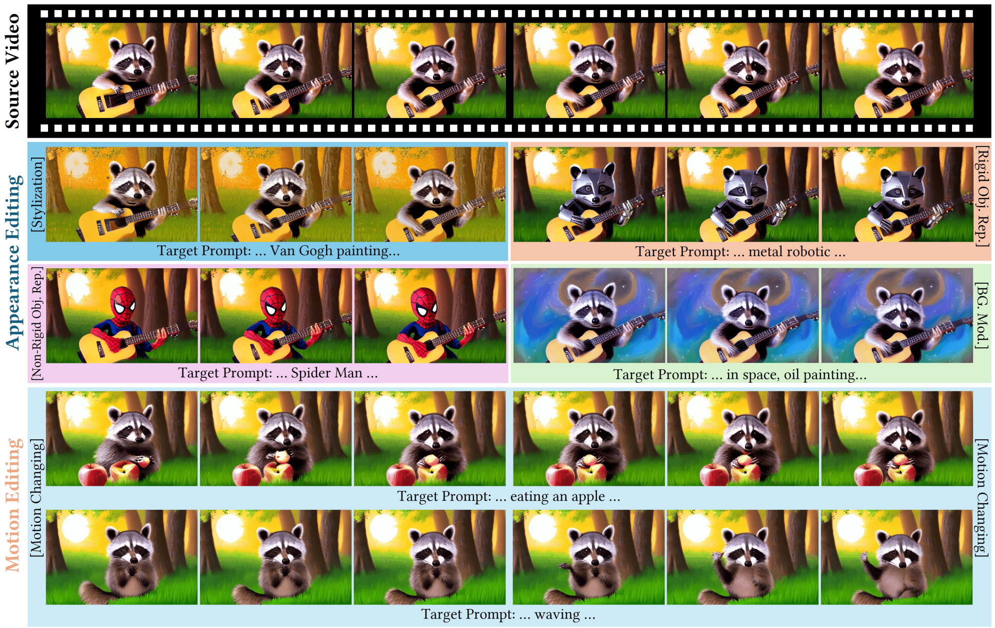

Source Video
Appearance Editing
[Stylization]
Target Prompt: ... Van Gogh painting...
[Rigid Obj. Rep.]
Target Prompt: ... metal robotic ...
[Non-Rigid Obj. Rep.]
Target Prompt: ... Spider Man ...
[BG. Mod.]
Target Prompt: ... in space, oil painting...
Motion Editing
[Motion Changing]
Target Prompt: ... eating an apple ...
[Motion Changing]
Target Prompt: ... waving ...

Figure 1: Examples edited by UniEdit. Our solution supports both video ***motion*** editing in the time axis (i.e., from playing guitar to eating or waving) and various video ***appearance*** editing scenarios (i.e., stylization, rigid/non-rigid object replacement, background modification). We encourage the readers to watch the videos on our project page.

Abstract

Recent advances in text-guided video editing have showcased promising results in appearance editing (e.g., stylization). However, video motion editing in the temporal dimension (e.g., from eating to waving), which distinguishes video editing from image editing, is underexplored. In this work, we present UniEdit, a tuning-free framework that supports both video motion and appearance editing by harnessing the power of a pre-trained text-to-video generator within an inversion-then-generation framework. To realize motion editing while preserving source video content, based on the insights that temporal and spatial self-attention layers encode inter-frame and intra-frame dependency respectively, we introduce auxiliary motion-reference and reconstruction branches to produce text-guided motion and source features respectively. The obtained features are then injected into the main editing path via temporal and spatial self-attention layers. Extensive experiments demonstrate that UniEdit covers video motion editing and various appearance editing scenarios, and surpasses the state-of-the-art methods. Our code will be publicly available.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

1 Introduction

The advent of pre-trained diffusion-based [26, 53] text-to-image generators [49, 50, 48] has revolutionized the fields of design and filmmaking, opening new vistas for creative expression. These advancements, underpinned by seminal works in text-to-image synthesis, have paved the way for innovative text-guided editing techniques for both images [42, 24, 4, 5] and videos [65, 6, 39, 70, 17, 46]. Such techniques not only enhance creative workflows but also promise to redefine content creation within these industries.

Video editing, in contrast to image editing, introduces the intricate challenge of ensuring frame-wise consistency. Efforts to address this challenge have led to the development of methods that leverage shared features and structures with the source video [6, 39, 37, 70, 46, 7, 33, 62, 18] through an inversion-then-generation pipeline [42, 53], exemplified by Pix2Video's approach [6] to consistent appearance editing across frames. To transfer the edited appearance from the anchor frame to the remaining frames consistently, it employs a pre-trained image generator and extends the self-attention layers to cross-frame attention to generate each remaining frame. Despite these advancements in performing video *appearance* editing (e.g., stylization, object appearance replacement, etc.), these methodologies fall short in editing video *motion* (e.g., replacing the movement of playing guitar with waving), hampered by a lack of motion priors and limited control over inter-frame dependencies, underscoring a critical gap in video editing capabilities.

Previous attempts [65, 44] at video motion editing through fine-tuning a pre-trained generator on the given source video and then editing motion through text guidance. Although effective, they necessitate a delicate balance between the generative prowess of the model and the preservation of the source video's content. This compromise often leads to restricted motion diversity and unwanted content variations, indicating a pressing need for a more robust solution.

In response, our work aims to explore a *tuning-free* framework that adeptly navigates the complexities of editing both the *motion* and *appearance* of videos. To achieve this, we identify three technical challenges: 1) it is non-trivial to incorporate the text-guided motion into the source content, as directly applying video appearance editing [46, 18] or image editing [5] schemes leads to undesirable results (as shown in Fig. 5); 2) preserving the non-edited content of the source video; 3) inheriting the spatial structure of the source video during appearance editing.

Our solution, UniEdit, harnesses the power of a pre-trained text-to-video generator (e.g., LaVie [63]) within an inversion-then-generation framework [42], tailored to overcome the identified challenges. Particularly, we introduce three key innovations: 1) To inject text-guided motion into the source content, we highlight the insight that ***the temporal self-attention layers of the generator encode the inter-frame dependency***. Acting in this way, we introduce an auxiliary motion-reference branch to generate text-guided motion features, which are then injected into the main editing path via temporal self-attention layers. 2) To preserve the non-edited content of the source video, motivated by the image editing technique [5], we follow the insight that ***the spatial self-attention layers of the generator encode the intra-frame dependency***. Therefore, we introduce an auxiliary reconstruction branch, and inject the features obtained from the spatial self-attention layers of the reconstruction branch into the main editing path. 3) To retain the spatial structure during the appearance editing, we replace the spatial attention maps of the main editing path with those in the reconstruction branch.

To our best knowledge, UniEdit represents a pioneering leap in text-guided, tuning-free video motion editing. In addition, its unified architecture not only facilitates a wide array of video appearance editing tasks, as shown in Fig. 1, but also empowers image-to-video generators for zero-shot text-image-to-video generation. Through comprehensive experimentation, we demonstrate UniEdit's superior performance relative to existing state-of-the-art methods, highlighting its potential to significantly advance the field of video editing.

2 Related Works

2.1 Video Generation

Researchers have achieved video generation with generative adversarial networks [58, 51, 61], language models [69, 71], or diffusion models [28, 52, 25, 23, 3, 60, 72, 19, 63, 8, 47]. To make the generation more controllable, recent endeavors have also incorporated additional structure guidance (e.g., depth map) [16, 10, 74, 11, 20, 64], or conducted customized generation [65, 67, 34, 75, 59, 41].

These models have generally learned real-world video distribution from large-scale data, and achieved promising results on text-to-video or image-to-video generation. Based on their success, we leverage the learned prior in the pre-trained model to achieve tuning-free video motion and appearance editing.

2.2 Video Editing

Video editing aims to produce a new video that is aligned with the provided editing instructions (e.g., text) while maintaining the other characteristics of the source video. It can be categorized into appearance and motion editing.

For appearance editing [70, 15, 17, 35, 12], like turn a video into the style of Van Gogh, the main challenge is to achieve temporal-consistent generation across different frames. Early attempts [6, 37, 46, 7, 33, 62] leveraged text-to-image models with inter-frame propagation to ensure consistency. For instance, Pix2Video [6] replaces the key and value of the current frame with those of the first and previous frame. Video-P2P [39] achieved local editing via video-specific fine-tuning and unconditional embedding optimization [43]. Follow-up studies [18, 70, 45] also leveraged the edit-then-propagate framework with neatest-neighbor field [18], estimated optical flow [70], or temporal deformation field [45]. Despite the promising results, due to the constraint on the source video structure, these approaches are specialized in appearance editing and can not be applied to motion editing directly.

Recent studies have also explored video motion editing with text guidance [65, 44], user-provided motion [32, 54, 15], or specific motion representation [55, 36, 22]. For example, Dreamix [44] proposed fine-tuning a pre-trained text-to-video model with mixed video-image reconstruction objectives for each source video. Then the editing is realized by conditioning the fine-tuned model on the given target prompt. MoCA [68] decoupled the video into the first-frame appearance and the optical flow, and trained a diffusion model to generate video conditioned on the first frame and the text. However, it struggled to preserve the non-edited motion (e.g., background dynamics) as it generates the entire motion from the text. Different from the aforementioned approaches that require fine-tuning or user-provided motion input, we are the first to achieve tuning-free motion and appearance editing with text guidance only.

3 Preliminaries: Video Diffusion Models

Our proposed UniEdit is built upon video diffusion models. Therefore, we first recap the architecture that is used in common text-guided video diffusion models [63, 2].

Overall Architecture Modern text-to-video (T2V) diffusion models typically extend a pre-trained text-to-image (T2I) model [49] to the video domain with the following adaptations. 1) Introducing additional temporal layers by inflating 2d convolutional layers to 3d form, or adding temporal self-attention layers [57] to model the correlation between video frames. 2) Due to the extensive computational resources for modeling spatial-temporal joint distribution, these works typically first train video generation models on low spatial and temporal resolutions, and then upsampling the generated results with cascaded models. 3) Other improvements like efficiency [1], training strategy [19], or additional control signals [16], etc. During inference, given standard Gaussian distribution $z_T \sim \mathcal{N}(0, 1)$, the denoising UNet is used to perform T denoising steps to obtain the outputs [26, 53]. If the model is trained in latent space [49], a decoder is employed to reconstruct videos from the latent domain.

Attention Mechanisms In particular, for each block of the denoising UNet, there are four basic modules: a convolutional module, a spatial self-attention module (SA-S), a spatial cross-attention module (CA-S), and a temporal self-attention module (SA-T). Formally, the attention operation [57] can be formulated as:

$$\texttt{attn}(Q, K, V) = \texttt{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where Q (query), K (key), V (value) are derived from inputs, and d is the dimension of hidden states.

Intuitively, CA-S is in charge of fusing semantics from the text condition, SA-S models the intra-frame dependency, SA-T models the inter-frame dependency and ensures the generated results are temporally consistent. We leverage these intuitions in our designs as elaborated below.

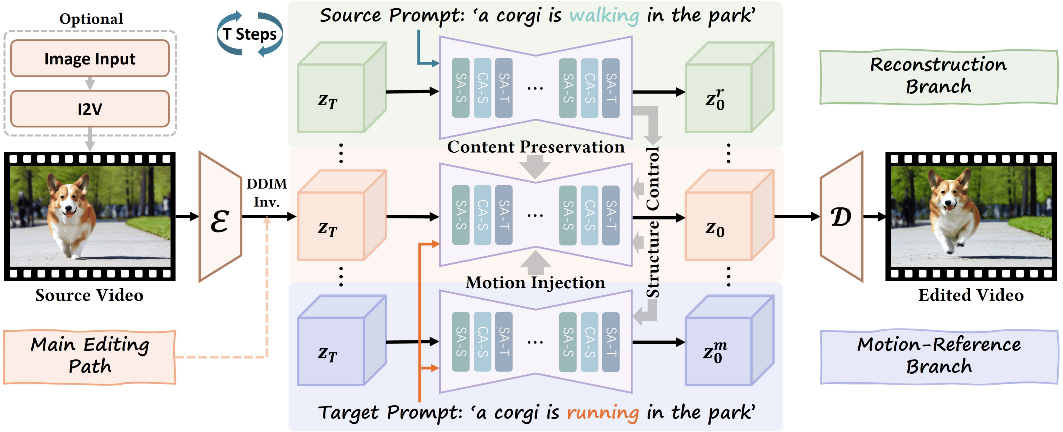


Figure 2: Overview of UniEdit. It follows an inversion-then-generation pipeline and consists of a main editing path, an auxiliary reconstruction branch and an auxiliary motion-reference branch. The reconstruction branch produces source features for content preservation, and the motion-reference branch yields text-guided motion features for motion injection. The source features and motion features are injected into the main editing path through spatial self-attention (SA-S) and temporal self-attention (SA-T) modules respectively (Sec. 4.1). We further introduce spatial structure control to retain the coarse structure of the source video (Sec. 4.2).

4 UniEdit

Method Overview. As shown in Fig. 2, our main editing path is based on an inversion-then-generation pipeline: we use the latent after DDIM inversion [53] as the initial noise z_T[1], then perform denoising process starting from z_T with the pre-trained UNet conditioned on the target prompt P_t. For motion editing, to achieve source content preservation and motion control, we propose to incorporate an auxiliary reconstruction branch and an auxiliary motion-reference branch to provide desired source and motion features, which are injected into the main editing path to achieve content preservation and motion editing (as shown in Fig. 3). We propose the pipeline of motion editing and appearance editing in Sec. 4.1 & Sec. 4.2 respectively. To further alleviate the background inconsistency, we introduce a mask-guided coordination scheme in Sec. 4.3. We also extend UniEdit to text-image-to-video generation (TI2V) in Sec. 4.4.

4.1 Tuning-Free Video Motion Editing

Content Preservation on SA-S Modules. One of the key challenges of editing tasks is to inherit the original content (e.g., textures and background) in the source video. To this end, we introduce an auxiliary reconstruction branch. The reconstruction path starts from the same inversed latent z_T similar to the main editing path, and then conducts the denoising process with the pre-trained UNet conditioned on the source prompt P_s to reconstruct the original frames. As verified in image editing [56, 24, 5], the attention features in the denoising model during reconstruction contain the content of the source video. Hence, we inject attention features of the reconstruction path into the main editing path on spatial self-attention (SA-S) layers for content preservation. At denoising step t, the attention operation of the l-th SA-S module in the main editing path is formulated as:

$$\text{SA-S}^l_{\text{edit}} := \begin{cases} \texttt{attn}(Q, K, V^r), & t < t_0 \text{ and } l > L, \\ \texttt{attn}(Q, K, V), & \text{otherwise}, \end{cases} \tag{2}$$

where Q, K, V are features in the main editing path, V^r refer to the value feature of the corresponding SA-S layer in the reconstruction branch, $t_0 = 50$ and $L = 10$ are hyper-parameters following previous work [5]. By replacing the value of spatial features, the video synthesized by the main editing path retains the non-edited characters (e.g., identity and background) of the source video, as exhibited in Fig. 7a. Unlike previous video editing works [37, 29] which introduces a cross-frame attention mechanism (i.e., using the key and value of the first/last frame), we implement Eq. 2 frame-wisely to better tackle source video with large dynamics.

[1] For real source video, we set source prompt to null during both forward and inversion process to achieve high-quality reconstruction [43].

Motion Injection on SA-T Modules. After implementing the content-preserving technique introduced above, we can obtain an edited video with the same content in the source video. However, it is observed that the output video could not follow the text prompt P_t properly. A straightforward solution is to increase the value of L so that balancing between the impact of injected information and the conditioned text prompt. Nevertheless, this could result in a content mismatch with the original source video in terms of structures and textures.

To obtain the desired motion without sacrificing content consistency, we propose to guide the main editing path with reference motion. Concretely, an auxiliary motion-reference branch (which also starts from the inversed latent z_T) is involved during the denoising process. Different from the reconstruction branch, the motion-reference branch is conditioned on the target prompt P_t, which contains the description of the desired motion. To transfer the motion into the main editing path, our core insight here is that ***temporal layers model the inter-frame dependency of the synthesized video clip*** (as shown in Fig. 6). Motivated by the observations above, we design the attention map injection on temporal self-attention layers of the main editing path:

$$\text{SA-T}^l_{\text{edit}} := \texttt{attn}(Q^m, K^m, V) \tag{3}$$

where Q^m and K^m refer to the query and key of the motion-reference branch, note that we replace the query and key of SA-T modules in the main editing path with those in the motion-reference branch on all layers and denoising steps. It's observed that the injection of temporal attention maps can effectively facilitate the main editing path to generate motion aligned with the target prompt. To better fuse the motion with the content in the source video, we also implement spatial structure control (refer to Sec. 4.2) on the main editing path and motion-reference branch in the early steps.

4.2 Tuning-Free Video Appearance Editing

In Sec. 4.1, we introduce the pipeline of UniEdit for video motion editing. In this subsection, we aim to perform appearance editing (e.g., style transfer, object replacement, background changing) via the same framework. In general, there are two main differences between appearance editing and motion editing. Firstly, appearance editing does not require changing the inter-frame relationships. Therefore, we remove the motion-reference branch and corresponding motion injection mechanism from the motion editing pipeline. Secondly, the main challenge of appearance editing is to maintain the structural consistency of the source video. To address this, we introduce spatial structure control between the main editing path and the reconstruction branch.

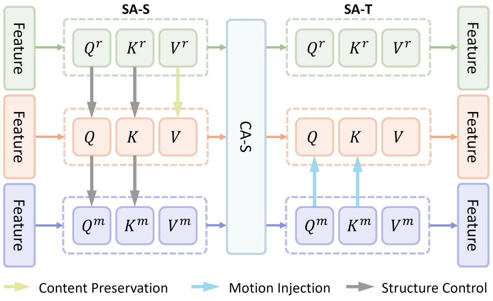


Figure 3: Detailed illustration of the relationship between the main editing path, the auxiliary reconstruction branch and the auxiliary motion-reference branch. The content preservation, motion injection and spatial structure control are achieved by the fusion of Q (query), K (key), V (value) features in spatial self-attention (SA-S) and temporal self-attention (SA-T) modules.

Spatial Structure Control on SA-S Modules. Previous approaches on video appearance editing [70, 18] mainly realize spatial structure control with the assistance of additional network [73]. When the auxiliary control model fails, it may result in inferior performance in preserving the structure of the original video. Alternatively, we suggest extracting the layout information of the source video from the reconstruction branch. Intuitively, the attention maps in spatial self-attention layers encode the structure of the synthesized video, as verified in Fig. 6. Hence, we replace the query and key of SA-S module in the main editing path with those in the reconstruction branch:

$$\text{SA-S}^l_{\text{edit}} := \begin{cases} \texttt{attn}(Q^r, K^r, V), & t < t_1, \\ \texttt{attn}(Q, K, V), & \text{otherwise}, \end{cases} \tag{4}$$

where Q^r and K^r refer to the query and key of the reconstruction branch, t_1 is used to control the extent of editing. It is worth mentioning that the effect of spatial structure control is distinct from the content preservation mechanism in Sec. 4.1. Take stylization as an example, the proposed structure control in Eq. 4 only ensures consistency in terms of each frame's composition, while enabling the model to generate the required textures and styles based on the text prompt. On the other hand, the content preservation technique inherits the textures and style of the source video. Therefore, we use structure control instead of content preservation for appearance editing. In addition, using the proposed structure control technique in motion editing can make the layout of the output video similar to the source video (shown in Fig. 11b in Appendix). Users have the flexibility to adjust the consistency between the edited video and the source video layout based on their specific requirements.

4.3 Mask-Guided Coordination (Optional)

To further improve the editing performance, we suggest leveraging the foreground/background segmentation mask M to guide the denoising process [14, 13]. There are two possible ways to obtain the mask M: the attention maps of CA-S modules with a threshold [24]; or employing an off-the-shelf segmentation model [38] on the source and generated videos. The obtained segmentation masks can be leveraged to 1), alleviate the indistinction in foreground and background; 2), improve content consistency between edited and source videos. To this end, we leverage mask-guided self-attention in the main editing path to coordinate the editing process. Formally, we define:

$$\texttt{m-attn}(Q, K, V; M) = \texttt{softmax}(\frac{QK^T}{\sqrt{d}} + M)V. \tag{5}$$

Then the mask-guided self-attention:

$$\text{SA}_{\text{mask}} := \texttt{m-attn}(Q, K, V; M^f) \odot M_m + \texttt{m-attn}(Q, K, V; M^b) \odot (1 - M_m), \tag{6}$$

where $M^f, M^b \in \{-\infty, 0\}$ indicate the foreground and background masks in the editing path respectively, $M_m \in \{0, 1\}$ denotes the foreground mask from the motion-reference branch, and $\odot$ is Hadamard product. In addition, we leverage the mask during the content preservation and motion injection for the features obtained from the reconstruction branch and the motion-reference branch (e.g., we replace Q^m with $M_m \odot Q^m + (1 - M_m) \odot Q$).

4.4 T2V Models are Zero-Shot TI2V Generators

To make our framework more flexible, we further derive a method to incorporate images as input and synthesize high-quality video conditioned on *both* image and text-prompt. Different from some image animation techniques [2], our method allows the user to guide the animation process with text prompts. Concretely, we first achieve image-to-video (I2V) generation by: 1) transforming input images with simulated camera movement to form a pseudo-video clip [44] or 2) leveraging existing image animation approaches (e.g., SVD [2], AnimateDiff [21]) to synthesis a video with random motion (which may not consistent with the text prompt). Then, we perform text-guided editing with UniEdit on the vanilla video to obtain the final output video.

5 Experiments

5.1 Comparison with State-of-the-Art Methods

Implementation Details UniEdit is not limited to specific video diffusion models. In this section, we build UniEdit upon LaVie [63] as an instantiation to verify the effectiveness of our method. To demonstrate the flexibility of UniEdit across different base models, we also implement the proposed method on VideoCrafter2 [9] and exhibit the editing results in Appendix B.1. For each input video, we follow the pre-processing step in LaVie to the resolution of 320 × 512. Then, the pre-processed video is fed into the UniEdit to perform video editing. It takes 1-2 minutes to edit on an NVIDIA A100 GPU for each video. More details can be found in Appendix A.

Baselines. To evaluate the performance of UniEdit, we compare the editing results of UniEdit with state-of-the-art motion and appearance editing approaches. For motion editing, due to the lack of open-source tuning-free (zero-shot) methods, we adapt the state-of-the-art non-rigid image editing technique MasaCtrl [5] to a T2V model [63] (denoted as MasaCtrl* in Fig. 5) and a one-shot video editing method Tune-A-Video (TAV) [65] as strong baselines. For appearance editing, we use the latest methods with strong performance, including FateZero [46], TokenFlow [18], and Rerender-A-Video (Rerender) [70] as baselines.

Evaluation Set. The evaluation set consists of 100 samples, including: **a)** 20 randomly sampled video clips from the open-source LOVEU-TGVE-2023 [66] dataset, along with their corresponding 80 text prompts, and **b)** 20 videos from online sources (`www.pexels.com` and `www.pixabay.com`), with manually designed prompts, as the baseline methods do not have an open-source evaluation set.

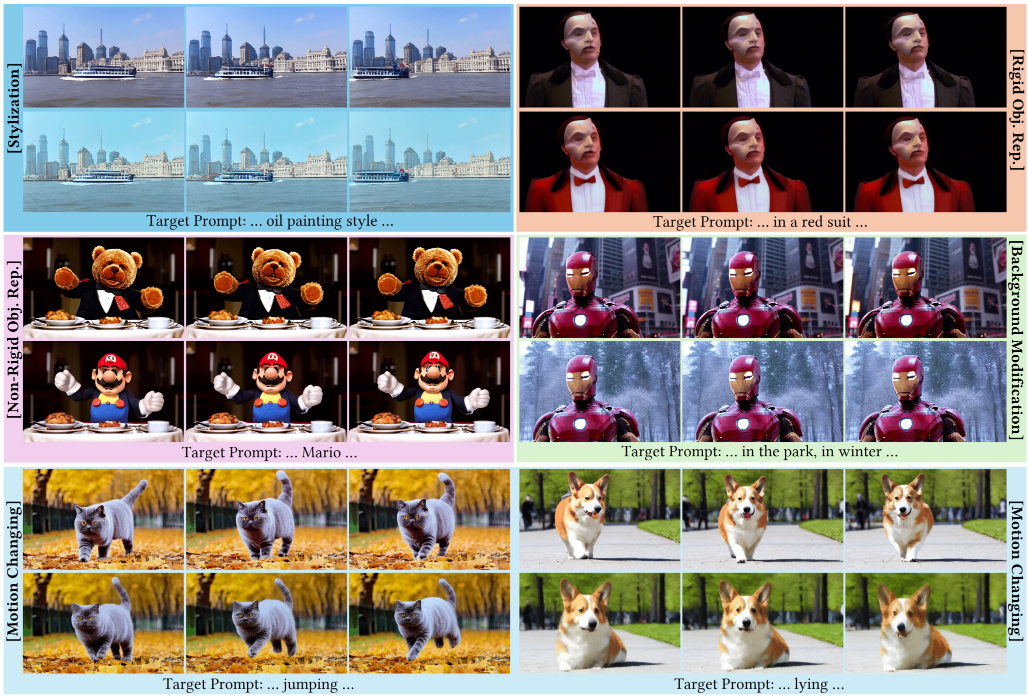

[Stylization]
Target Prompt: ... oil painting style ...
[Rigid Obj. Rep.]
Target Prompt: ... in a red suit ...
[Non-Rigid Obj. Rep.]
Target Prompt: ... Mario ...
[Background Modification]
Target Prompt: ... in the park, in winter ...
[Motion Changing]
Target Prompt: ... jumping ...
[Motion Changing]
Target Prompt: ... lying ...

Figure 4: Examples edited by UniEdit. For each case, the upper frames come from the source video, and the lower frames indicate the edited results with the target prompt. We encourage the readers to watch the videos and make evaluations.

Qualitative Results. We present editing examples of UniEdit in Fig. 1, Fig. 4 (additional examples in Fig. 16-21 of Appendix B.8). Please visit our project page for more videos. UniEdit demonstrates the ability to: 1) edit in various scenarios, including motion-changing, object replacement, style transfer, and background modification; 2) align with the target prompt; and 3) maintain excellent temporal consistency. Additionally, we compare UniEdit with state-of-the-art methods in Fig. 5 (further comparisons in Fig.13,14,15 of Appendix B.7). For a fair comparison, we also migrated all baselines to LaVie [63], using the same base model as our method. The results are presented in Fig. 15. For appearance editing, we showcase two scenarios: non-rigid object replacement and stylization. In object replacement, our method outperforms baselines in terms of prompt alignment and background consistency. In stylization, UniEdit excels in preserving content. For example, the grassland retains its original appearance without any additional elements. In motion editing, UniEdit surpasses baselines in aligning the video with the target prompt and preserving the source content.

Quantitative Results. We quantitatively evaluate our method using two approaches: **1)** CLIP scores and user preference, as employed in previous work [65]; and **2)** VBench [31] scores, a recently proposed benchmark suite for T2V models. The summarized results are in Tab. 1. Following previous work [65], we assess the effectiveness of our method in terms of temporal consistency and alignment with the target prompt. Additionally, we conducted a user study involving 10 participants who rated the edited videos on a scale of 1 to 5. We also utilize the recently proposed VBench [31] benchmark to provide a more comprehensive assessment, which includes 'Frame Quality' metrics and 'Temporal Quality' metrics. UniEdit outperforms the baseline methods across all metrics. Furthermore, the mask-guided coordination technique introduced in Sec. 4.3 further enhances performance (see Appendix B.3). For more detailed quantitative results, please refer to Appendix B.2&B.3&B.5.

5.2 Ablation Study and Analysis

How UniEdit Works? To better understand how UniEdit works and reveal our insight on the spatial and temporal self-attention layers, we visualize the features in the SA-S and SA-T modules and compare them with the magnitude of optical flow between adjacent frames in Fig. 6a. It is evident that, in comparison to the spatial query maps (2nd row), the temporal cross-frame attention maps (3rd row) exhibit a notably higher degree of overlap with the optical flow (4th row). This indicates that the temporal self-attention layers encode inter-frame dependencies and facilitate motion injection, while content preservation and structure control are carried out in the spatial self-attention layers.

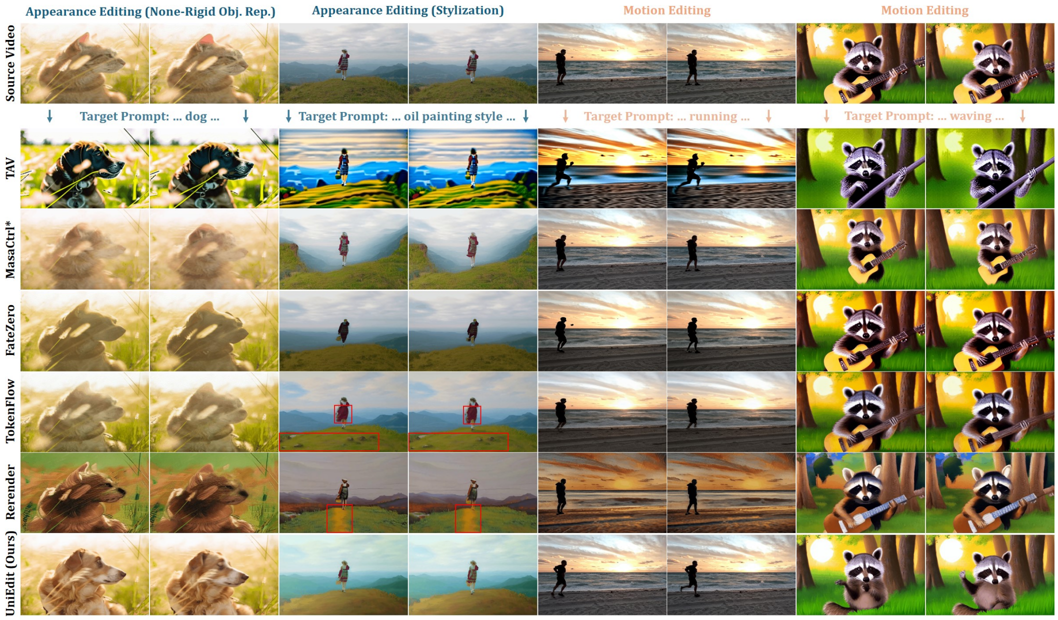

Appearance Editing (None-Rigid Obj. Rep.)
Appearance Editing (Stylization)
Motion Editing
Motion Editing
Source Video
Target Prompt: ... dog ...
Target Prompt: ... oil painting style ...
Target Prompt: ... running ...
Target Prompt: ... waving ...
TAV
MasaCtrl*
FateZero
TokenFlow
Rerender
UniEdit (Ours)

Figure 5: Comparison with state-of-the-art methods for both video appearance and motion editing. It shows that UniEdit achieves better source content preservation, and outperforms baselines in motion editing by a large margin.

Table 1: Quantitative comparison with state-of-the-art video editing techniques.

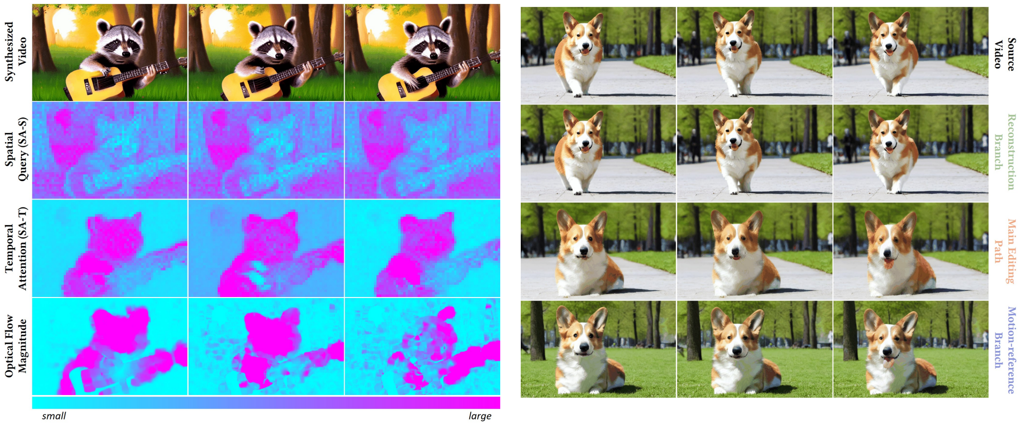
	Frame Consistency		Textual Alignment		Frame Quality		Temporal Quality		
Method	CLIP Score	User Pref.	CLIP Score	User Pref.	Aesthetic Quality	Imaging Quality	Subject Consistency	Motion Smoothness	Temporal Flickering
TAV [65]	95.39	3.74	27.89	3.30	51.97	49.60	93.10	93.27	91.48
MasaCtrl* [5]	97.61	4.31	25.58	3.17	54.58	58.72	93.04	95.70	94.29
FateZero [46]	96.72	4.48	27.30	3.48	53.77	56.99	93.55	94.80	93.42
Rerender [70]	97.18	4.16	27.94	3.55	54.59	57.97	93.08	95.57	94.36
TokenFlow[18]	97.02	4.50	28.58	3.34	52.60	60.65	91.97	95.04	93.50
UniEdit	98.35	4.72	31.43	4.79	58.25	62.94	95.73	**97.30**	96.74
UniEdit-Mask	**98.36**	**4.73**	**31.50**	**4.90**	**58.77**	**63.12**	**95.86**	97.28	**96.79**

Output Visualization of the Two Auxiliary Branches. Recall that to perform motion editing, we propose to transfer the targeted motion from the motion-reference branch and realize content preservation via feature injection from the reconstruction branch. To verify the effectiveness, we visualized the output of each branch in Fig. 6b. It is observed that the motion-reference branch (4th row) generates video with the target motion, and effectively transfers it to the main path (3rd row); meanwhile, the main path inherits the content from the reconstruction branch (2nd row), thus enhancing the consistency of unedited parts.

Table 2: Impact of various components.

Content Preservation	Motion Injection	Structure Control	Frame Similarity	Textual Alignment	Frame Consistency
			90.54	28.76	96.99
✓			97.28	29.95	98.12
	✓	✓	91.30	31.48	98.08
✓	✓		96.11	31.37	98.12
✓	✓	✓	96.29	31.43	98.09

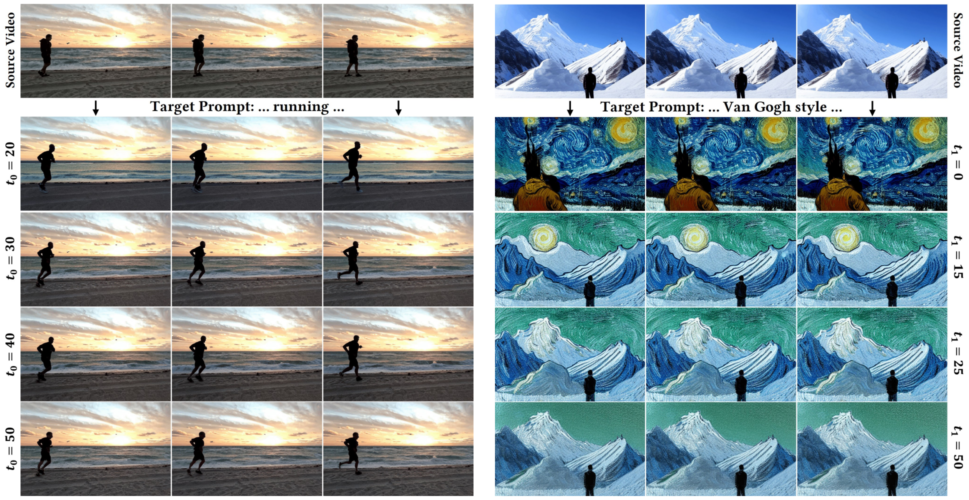
The Effectiveness of Each Component. To demonstrate that all the designed feature injection techniques in Sec. 4.1 & 4.2 contribute to the final results, we make a quantitative evaluation on 15 motion editing cases, as we utilize all three components in motion editing. To assess the similarity between the edited video and the source video (e.g., background and identity), we introduce the 'Frame Similarity', which is the average frame cosine similarity between the source frame embedding and the edited frame embedding. As shown in Tab. 2, editing with *content preservation* results in high frame similarity, suggesting that replacing value features in SA-S modules can effectively retain the content of the source video. The use of *motion injection* and *structure control* significantly enhances 'Textual Alignment', indicating successful transfer of the targeted motion to the main editing path. Ultimately, the best results are achieved through the combined use of all components.



(a) Visualization of attention features.

(b) Visualization of each branch's output.

Figure 6: (6a): Visualization of spatial query in SA-S (second row), cross-frame temporal attention maps in SA-T (third row), and the magnitude of optical flow (fourth row). (6b): Visualization of the video output of the main editing path, the reconstruction branch and the motion-reference branch.



(a) Ablation study on t_0 in Eq. 2.

(b) Ablation study on t_1 in Eq. 4.

Figure 7: Ablation study on hyper-parameters.

Ablation on Hyper-parameters. We utilize content preservation in Eq. 2 to maintain the original content from the source video. By varying the feature injection steps in Fig. 7a, we observe that replacing the value features at a few steps introduces inconsistencies in the background (footprints on the beach). In practice, we adhere to the hyper-parameter selection outlined in [5] (last row). Simultaneously, we note that adjusting the blend layers and steps in Eq. 4 can effectively regulate the extent to which the edited image adheres to the original image. For instance, in the stylization demonstrated in Fig. 7b, injecting the attention map into fewer (15) steps yields a stylized output that may not retain the same structure as the input, while injecting into all 50 steps results in videos with nearly identical textures but less stylization. Users have the flexibility to adjust the blended steps to achieve their preferred balance between stylization and fidelity.

6 Conclusion and Limitations

In this paper, we design a novel tuning-free framework UniEdit for both video motion and appearance editing. By leveraging a motion-reference branch and a reconstruction branch and injecting features into the main editing path, it is capable of performing motion editing and various appearance editing. There are nevertheless some limitations. Firstly, we observe performance degradation when performing both types of editing simultaneously. Secondly, since our work is based on T2V models, the proposed method also inherits some of the shortcomings of the existing models, such as inferior performance in understanding complex prompts. We exhibit the failure cases in Appendix B.6.

References

[1] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[3] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[6] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23206–23217, 2023.

[7] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023.

[8] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[9] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. *arXiv preprint arXiv:2401.09047*, 2024.

[10] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023.

[11] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023.

[12] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023.

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

[14] Paul Couairon, Clément Rambour, Jean-Emmanuel Haugeard, and Nicolas Thome. Videdit: Zero-shot and spatially aware text-driven video editing. *arXiv preprint arXiv:2306.08707*, 2023.

[15] Yufan Deng, Ruida Wang, Yuhao Zhang, Yu-Wing Tai, and Chi-Keung Tang. Dragvideo: Interactive drag-style video editing. *arXiv preprint arXiv:2312.02216*, 2023.

[16] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[17] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. *arXiv preprint arXiv:2309.16496*, 2023.

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

[19] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023.

[20] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint arXiv:2311.16933*, 2023.

[21] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[22] Tianyu He, Junliang Guo, Runyi Yu, Yuchi Wang, Jialiang Zhu, Kaikai An, Leyi Li, Xu Tan, Chunyu Wang, Han Hu, et al. Gaia: Zero-shot talking avatar generation. In *International Conference on Learning Representations (ICLR)*, 2024.

[23] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022.

[24] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *International Conference on Learning Representations (ICLR)*, 2023.

[25] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[28] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[29] Hanzhuo Huang, Yufan Feng, Cheng Shi, Lan Xu, Jingyi Yu, and Sibei Yang. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator. *arXiv preprint arXiv:2309.14494*, 2023.

[30] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. *arXiv preprint arXiv:2312.06739*, 2023.

[31] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[32] Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. Vmc: Video motion customization using temporal attention adaption for text-to-video diffusion models. *arXiv preprint arXiv:2312.00845*, 2023.

[33] Hyeonho Jeong and Jong Chul Ye. Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models. *arXiv preprint arXiv:2310.01107*, 2023.

[34] Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and Ziwei Liu. Videobooth: Diffusion-based video generation with image prompts. *arXiv preprint arXiv:2312.00777*, 2023.

[35] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. *arXiv preprint arXiv:2312.04524*, 2023.

[36] Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. Dreampose: Fashion image-to-video synthesis via stable diffusion. *arXiv preprint arXiv:2304.06025*, 2023.

[37] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

[38] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

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

[40] Qi Mao, Lan Chen, Yuchao Gu, Zhen Fang, and Mike Zheng Shou. Mag-edit: Localized image editing in complex scenarios via mask-based attention-adjusted guidance. *arXiv preprint arXiv:2312.11396*, 2023.

[41] Joanna Materzynska, Josef Sivic, Eli Shechtman, Antonio Torralba, Richard Zhang, and Bryan Russell. Customizing motion in text-to-video diffusion models. *arXiv preprint arXiv:2312.04966*, 2023.

[42] 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 *International Conference on Learning Representations*, 2022.

[43] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[44] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

[45] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926*, 2023.

[46] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[47] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[50] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[51] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017.

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

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[54] Yao Teng, Enze Xie, Yue Wu, Haoyu Han, Zhenguo Li, and Xihui Liu. Drag-a-video: Non-rigid video editing with point-based interaction. *arXiv preprint arXiv:2312.02936*, 2023.

[55] Shuyuan Tu, Qi Dai, Zhi-Qi Cheng, Han Hu, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. Motioneditor: Editing video motion via content-aware diffusion. *arXiv preprint arXiv:2311.18830*, 2023.

[56] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016.

[59] Cong Wang, Jiaxi Gu, Panwen Hu, Songcen Xu, Hang Xu, and Xiaodan Liang. Dreamvideo: High-fidelity image-to-video generation with image retention and text guidance. *arXiv preprint arXiv:2312.03018*, 2023.

[60] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[61] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. *Advances in Neural Information Processing Systems*, 32, 2019.

[62] Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023.

[63] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023.

[64] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *arXiv preprint arXiv:2312.03641*, 2023.

[65] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[66] 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.

[67] Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, et al. Make-your-video: Customized video generation using textual and structural guidance. *arXiv preprint arXiv:2306.00943*, 2023.

[68] Wilson Yan, Andrew Brown, Pieter Abbeel, Rohit Girdhar, and Samaneh Azadi. Motion-conditioned image animation for video editing. *arXiv preprint arXiv:2311.18827*, 2023.

[69] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

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

[71] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[72] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023.

[73] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[74] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023.

[75] Yuxin Zhang, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Motioncrafter: One-shot motion customization of diffusion models. *arXiv preprint arXiv:2312.05288*, 2023.

Supplementary Materials

We organize the Appendix as follows:

● Appendix A: detailed descriptions of experimental settings.

● Appendix B: more experimental results, including:

- Editing results on different T2V model (Appendix B.1).
- Quantitative ablation on hyper-parameter selection (Appendix B.2).
- Ablation study on mask-guided coordination (Appendix B.3).
- Observation and analysis on the proposed components (Appendix B.4).
- Analysis and comparison on inference time (Appendix B.5).
- Failure cases visualization (Appendix B.6).
- More Comparisons with baseline methods (Appendix B.7).
- More Editing results of UniEdit (Appendix B.8).

● Appendix C: Broader Impacts.

We encourage the readers to watch the videos on our project page.

A Detailed Experimental Settings

Base T2V Model. We instantiate the proposed method on LaVie [63], which is a pre-trained text-to-video generation model that produces consistent and high-quality videos. To achieve a fair comparison, we only leverage the base T2V model in LaVie and load the open-source pre-trained weights for video editing tasks in the experiments. Note that the edited video clip could further be seamlessly fed into the temporal interpolation model and the video super-resolution model to obtain video with a longer duration and higher resolution.

Video Preprocessing. For each input video, we resize it to the resolution of 320×512, followed by normalization, which is consistent with the training configuration of LaVie. Then, the pre-processed video is fed into the base model of Lavie to perform video editing. To maximize the generation power of LaVie, we set all input videos to 16 frames. For a source video, it takes 1-2 minutes to edit on an NVIDIA A100 GPU.

Configurations. For real source videos, we inverse them with 50 DDIM inversion steps and perform DDIM deterministic sampling with 50 steps for generation. For the generated videos, we use the same start latent of synthesizing the source video as the initial noise z_T for the main editing path and two auxiliary branches. We use the commonly used classifier-free guidance technique [27] with a scale of 7.5.

Details of User Study. As a text-guided editing task, in addition to CLIP scores, it is crucial to evaluate results through human subjective assessment. To achieve this, we utilized MOS (Mean Opinion Score) as our metric and collected feedback from 10 experienced volunteers. We randomly selected 20 editing samples and permuted results from different models. Volunteers were then tasked to evaluate the results based on two perspectives: frame consistency and textual alignment. They provided ratings for these aspects on a scale of 1-5. Specifically, frame consistency measures the smoothness of the video, aiming to avoid dramatic jittering and ensure coherence between the content of each frame. Textual alignment assesses whether the editing results adhere to the text guidance and maintain the content of the source video. In the end, we computed the average user ratings for each method as our final results.

As illustrated in Tab. 1, UniEdit shows the best performance on frame consistency. Regarding textual alignment, UniEdit significantly outperforms all other baselines, demonstrating its capacity to support diverse editing scenarios.

Baselines. We implement all baseline methods with their official repositories. For MasaCtrl [5], we adapt it to video editing by first setting the base model to a T2V model [63], then performing MasaCtrl on all frames of the source video. Moreover, since most baselines use StableDiffusion (SD) as the base model, we resize the source video to 512×512 to align with the default configuration of SD, then feed it into the denoising model, which can maximize the power of SD.

B Additional Experimental Results and Analysis

B.1 Results on Different T2V Model

We additionally implement our method on VideoCrafter2 [9], a concurrent work on T2V generation to demonstrate the flexibility of UniEdit. The results are shown in Fig. 8.

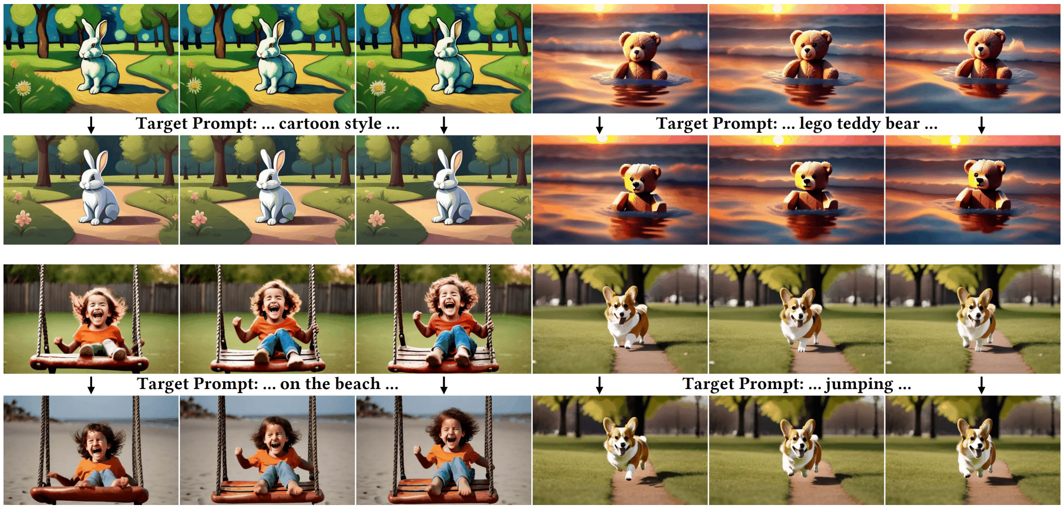

Target Prompt: ... cartoon style ...
Target Prompt: ... lego teddy bear ...
Target Prompt: ... on the beach ...
Target Prompt: ... jumping ...

Figure 8: Editing results with UniEdit on VideoCrafter2 [9].

B.2 Quantitative Ablation on Hyper-parameter Selection

In practice, we empirically found set these values to fixed values, i.e., $t_0 = 50, L = 10$ (same as MasaCtrl [5]) and $t_1 = 25$ can achieve satisfying results on most cases, and we further perform a quantitative study when applying different hyper-parameters in Tab. 3&4.

Table 3: Quantitative comparison on hyper-parameter selection.

Metric	Frame Similarity	Textual Alignment	Frame Consistency
$t_0 = 20, L = 10$	94.33	31.57	98.09
$t_0 = 50, L = 10$	96.29	31.84	98.12
$t_0 = 50, L = 8$	96.76	31.25	98.11

Table 4: Quantitative comparison on hyper-parameter selection.

Metric	Frame Similarity	Textual Alignment	Frame Consistency
$t_1 = 20$	96.21	30.92	98.06
$t_1 = 25$	96.29	31.43	98.09
$t_1 = 30$	96.50	31.04	98.08

B.3 Ablation Study on the Impact of Mask-Guided Coordination

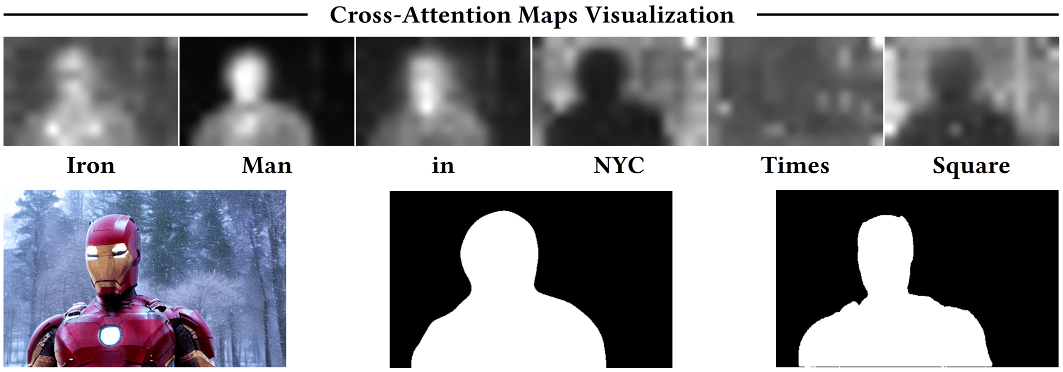
To investigate the impact of mask-guided coordination, we begin by visualizing masks obtained from 1) the attention map in CA-S modules; 2) the off-the-shelf segmentation model SAM [38], followed by presenting both qualitative and quantitative results of implementing UniEdit with or without mask-guided coordination.

As verified by previous work [24], the attention maps in CA-S modules contain correspondence information between text and visual features. The underlying intuition is that the attention maps between each word and the spatial features at point (i, j) indicate 'how similar this token is to the spatial feature at this location'. We visualize the text-image cross attention map alongside the synthesized frame in Fig. 9. We observe spatial correspondences that align with the video output from the attention map. For instance, areas with higher values of the token 'man' and 'NYC' correspond to the foreground and background, respectively. We further employ a fixed threshold (0.4 in practice) to derive binary segmentation maps from the attention maps. For comparison, we also display the segmentation mask obtained by point prompt on SAM. It's observed that the cross-attention mask is generally accurate and could serve as a reliable proxy in practice when an external segmentor is not available.

We examine the impact of mask-guided coordination through both qualitative and quantitative results across 4 settings: {w/o UniEdit, UniEdit w/o mask, UniEdit with mask from CA-S, UniEdit with mask from SAM}. Qualitatively, shown in Fig. 10, the implementation of UniEdit significantly enhances the consistency between the edited videos and the original video. The application of the mask-guided coordination technique further improves the consistency of unedited areas (e.g., color and texture). The quantitative results in Tab. 5 align coherently with this analysis.

Table 5: Ablation on the proposed mask-guided coordination.

Metric	Textual Alignment	Frame Consistency
TAV	27.89	95.39
MasaCtrl*	25.58	97.61
FateZero	27.30	96.72
Rerender	27.94	97.18
TokenFlow	28.58	97.02
UniEdit (w/o mask)	31.43	98.35
UniEdit (w CA-S mask)	31.49	98.33
UniEdit (w SAM mask)	31.50	98.36



(a) Output frame.

(b) CA-S Mask.

(c) SAM Mask.

Figure 9: Visualization of attention maps and masks in mask-guided coordination (Sec. 4.3). The top row are attention maps corresponding to different tokens in CA-S modules, (a) is the final output frame, (b) and (c) are the foreground/background binary mask obtained by employing a threshold on the attention map of 'Man' token and point prompt segmentation with SAM, respectively.

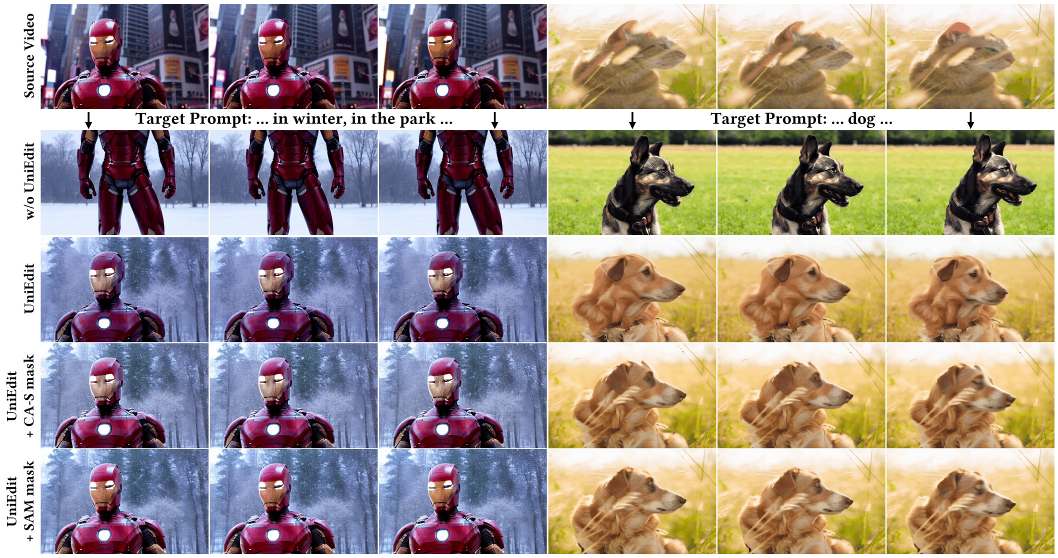
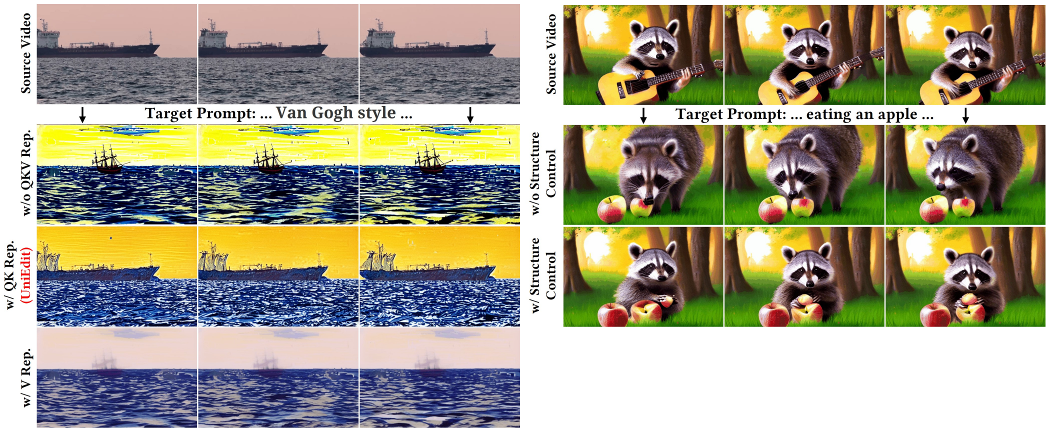

Source Video
Target Prompt: ... in winter, in the park ...
Target Prompt: ... dog ...
w/o UniEdit
UniEdit
UniEdit + CA-S mask
UniEdit + SAM mask

Figure 10: Qualitative editing results across 4 settings: w/o UniEdit (2nd row), UniEdit w/o mask (3rd row), UniEdit with mask from CA-S (4th row), UniEdit with mask from SAM (5th row).

B.4 More Observation and Analysis on the Proposed Components

Difference Between QK and V Features in SA-S Modules To comprehend why we can have inhomogeneous QK and V and their differences, we visualized the results of swapping different features (QK or V) in SA-S modules during style transfer tasks on the source video in Fig. 11a. As can be seen, compared to editing with no feature replacement (2nd row), replacing QK in the 3rd row results in the edited video adopting the same spatial structure as the source video. Simultaneously, replacing V eradicates the style information in the 4th row, meaning the texture details from the source video are utilized to replace the style depicted by the target prompt. To summarize, the query and key features (in SA-S modules) dictate the spatial structure of the generated video, while the value features tend to influence the texture, including details such as color tones.

Influence of Spatial Structure Control in Motion Editing We explored the role of spatial control in motion editing. The proposed method synthesizes videos with larger modifications when removing the spatial control mechanism on both the motion-reference branch and the main editing branch. We visualized the results in Fig. 11b. It can be observed that although the motion-reference branch can still generate the target motion without the control of spatial structure, the layout deviates significantly, for example, the raccoon assumes a different pose and location. We regard this as a suboptimal solution because, compared to the results presented in the 3rd row, the results w/o spatial structure control modifies the object position of the source video, leading to a decrease in consistency between the edited result and the source video.



(a) Replacing different features in SA-S modules.

(b) Motion editing w/ or w/o structure control.

Figure 11: Ablation on the proposed feature injection techniques. (11a): comparison of appearance editing without feature replacement (2nd row), with QK replacement (3rd row), with V replacement (4nd row); (11b): comparison of motion editing with and without the designed spatial structure control mechanism.

B.5 Analysis and Comparison on Inference Time

We conduct a theoretical analysis of the additional cost of UniEdit and an empirical comparison with baseline methods in terms of inference speed.

Theoretically, our method primarily involves feature replacement operations in attention modules, achieved through forward hook registration and introducing minimal additional computation. Therefore, the main difference between synthesizing a video from random noise and editing a video with UniEdit lies in the batch size of the denoising process (i.e., vanilla generation: batchsize=1, appearance editing: batchsize=2, motion editing: batchsize=3), and this process could be further accelerated through multi-GPU parallel processing techniques. Additionally, we utilize LaVie [63] as the base T2V model in the paper, which takes approximately 45 seconds to synthesize a 16-frame video. Our method can be even faster when adapted to more efficient base models.

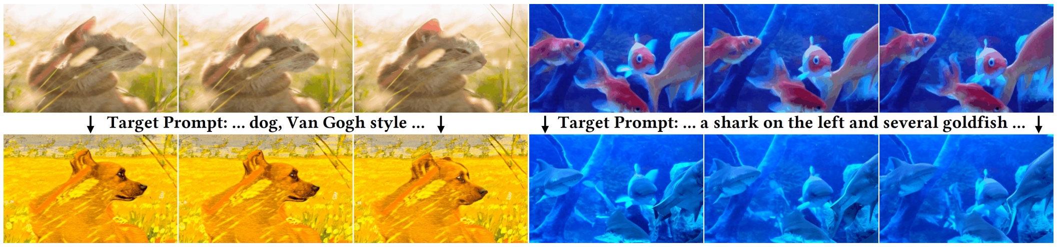
Empirically, UniEdit demonstrates comparable speed with baseline methods. The comparison of inference time on a single 16-frame source video clip with a resolution of 320x512 on 1 NVIDIA A100 GPU is as follows:

Table 6: Quantitative comparison on inference time of editing a single 16-frame video clip.

Method	TAV	MasaCtrl*	FateZero	Rerender	TokenFlow	UniEdit (appearance editing)	UniEdit (motion editing)
Inference time	∼10min	∼90s	∼130s	∼110s	∼100s	∼95s	∼125s

B.6 Failure Cases Visualization

We exhibit failure cases in Fig. 12. Fig. 12a showcase when editing multiple elements simultaneously, and we observe a relatively large inconsistency with the source video. A naive solution is to perform editing with UniEdit multiple times. Fig. 12b visualizes the results when editing video with complex scenes, and the model sometimes could not understand the semantics in the target prompt, resulting in incorrect editing. This may be caused by the base model's limited text understanding power, as discussed in [30]. It could be alleviated by leveraging the reasoning power of MLLM [30], or adapting approaches in complex scenario editing [40].


↓ Target Prompt: ... dog, Van Gogh style ... ↓
↓ Target Prompt: ... a shark on the left and several goldfish ... ↓

(a) Edit multiple elements simultaneously.

(b) Complex scene editing.

Figure 12: Visualization of failure cases.

B.7 More Comparison with State-of-the-Art Methods

Please refer to Fig. 13 and Fig. 14 for more comparison with the state-of-the-art methods. For a fair comparison, we also migrated all baselines to LaVie [63], using the same base model as our method. The results are presented in Fig. 15, and they are found to be inferior compared to those in Fig. 5 (based on Stable Diffusion).

B.8 More Results of UniEdit

More edited results of UniEdit are provided in Fig. 16-21. Examples of TI2V generation are provided in Fig. 22.

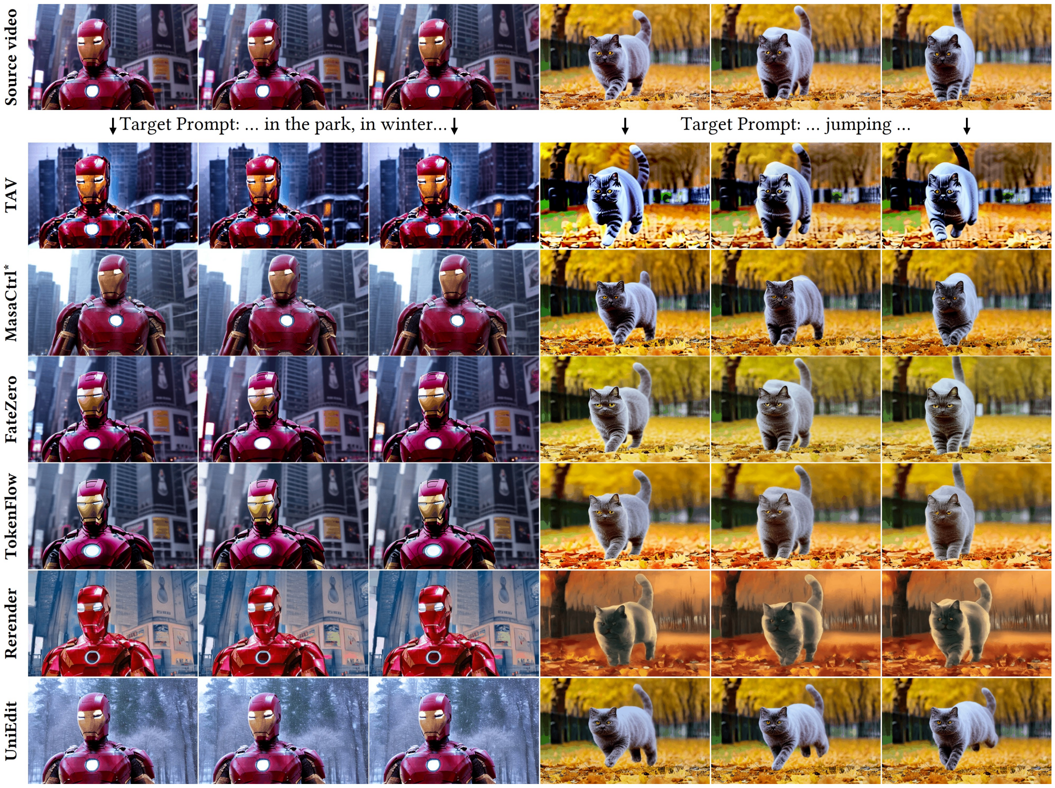

Source video
Target Prompt: ... in the park, in winter...
Target Prompt: ... jumping ...
TAV
MasaCtrl*
FateZero
TokenFlow
Rerender
UniEdit

Figure 13: More comparison with state-of-the-art methods.

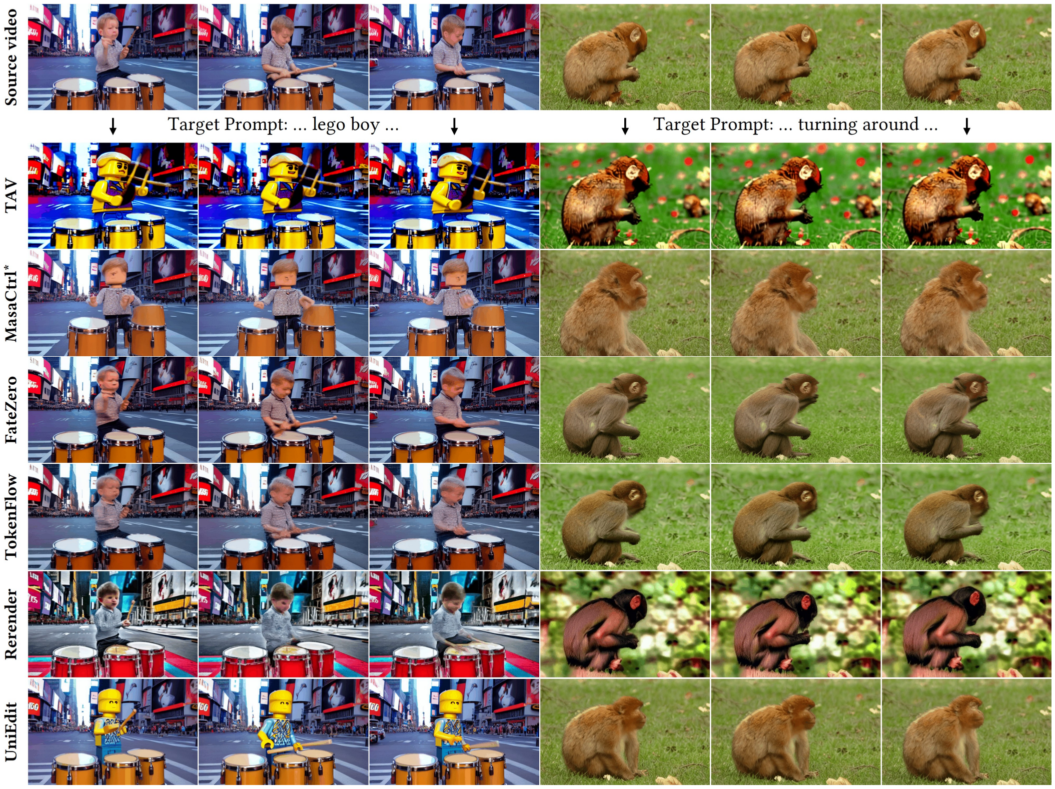

Source video
Target Prompt: ... lego boy ...
Target Prompt: ... turning around ...
TAV
MasaCtrl*
FateZero
TokenFlow
Rerender
UniEdit

Figure 14: More comparison with state-of-the-art methods.

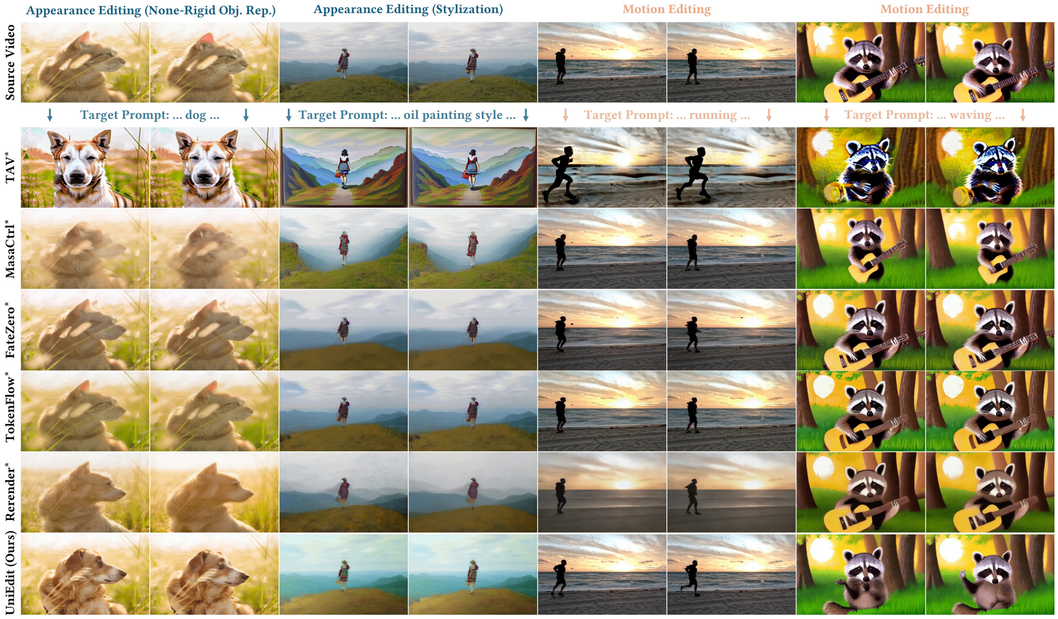
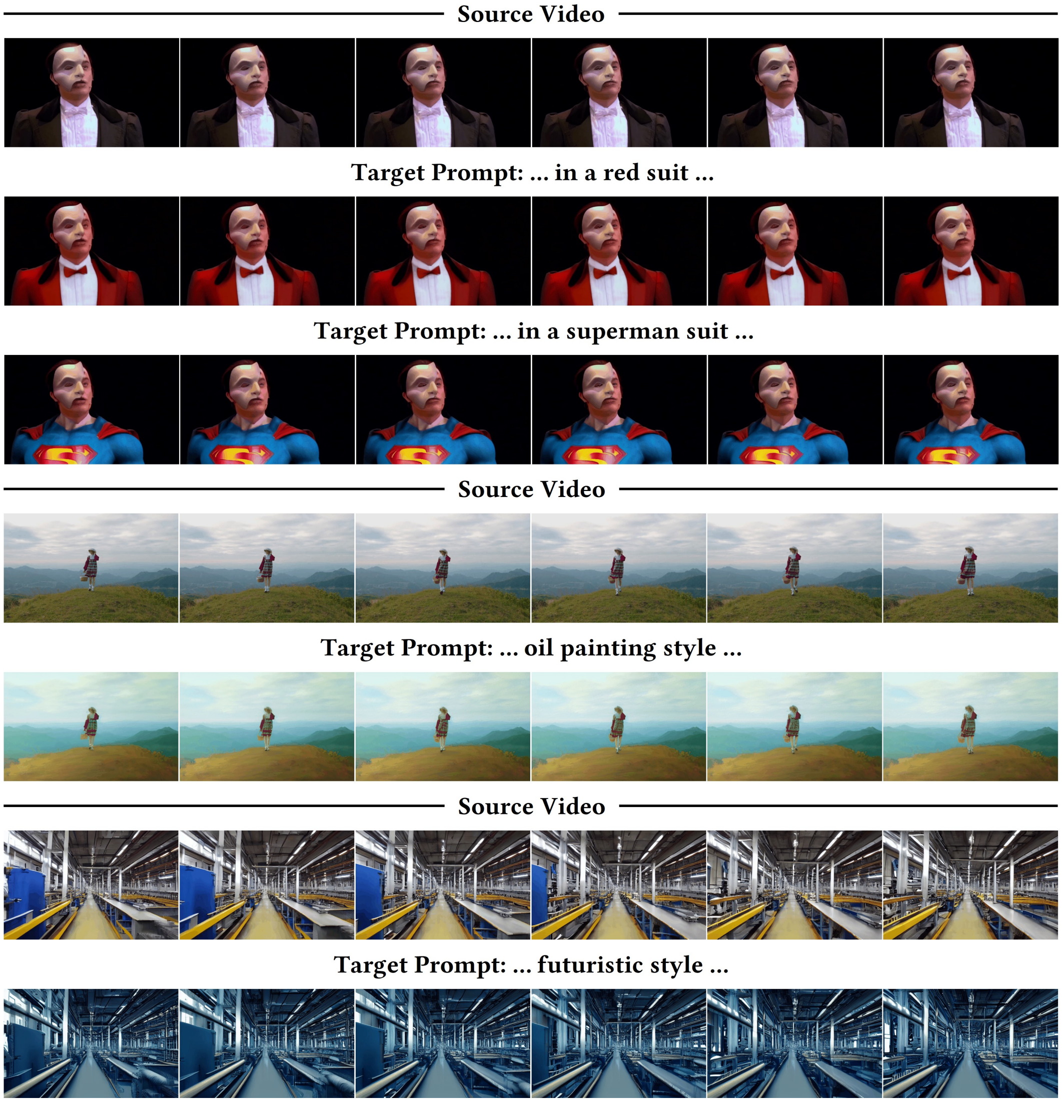

Appearance Editing (None-Rigid Obj. Rep.)
Appearance Editing (Stylization)
Motion Editing
Motion Editing
Source Video
Target Prompt: ... dog ...
Target Prompt: ... oil painting style ...
Target Prompt: ... running ...
Target Prompt: ... waving ...
TAV*
MasaCtrl*
FateZero*
TokenFlow*
Rerender*
UniEdit (Ours)

Figure 15: More comparison with state-of-the-art methods. We adapt the baseline methods to the text-to-video model LaVie [63] and compare with our method (also based on LaVie).



Source Video

Target Prompt: ... in a red suit ...

Target Prompt: ... in a superman suit ...

Source Video

Target Prompt: ... oil painting style ...

Source Video

Target Prompt: ... futuristic style ...

Figure 16: More appearance editing results of UniEdit.

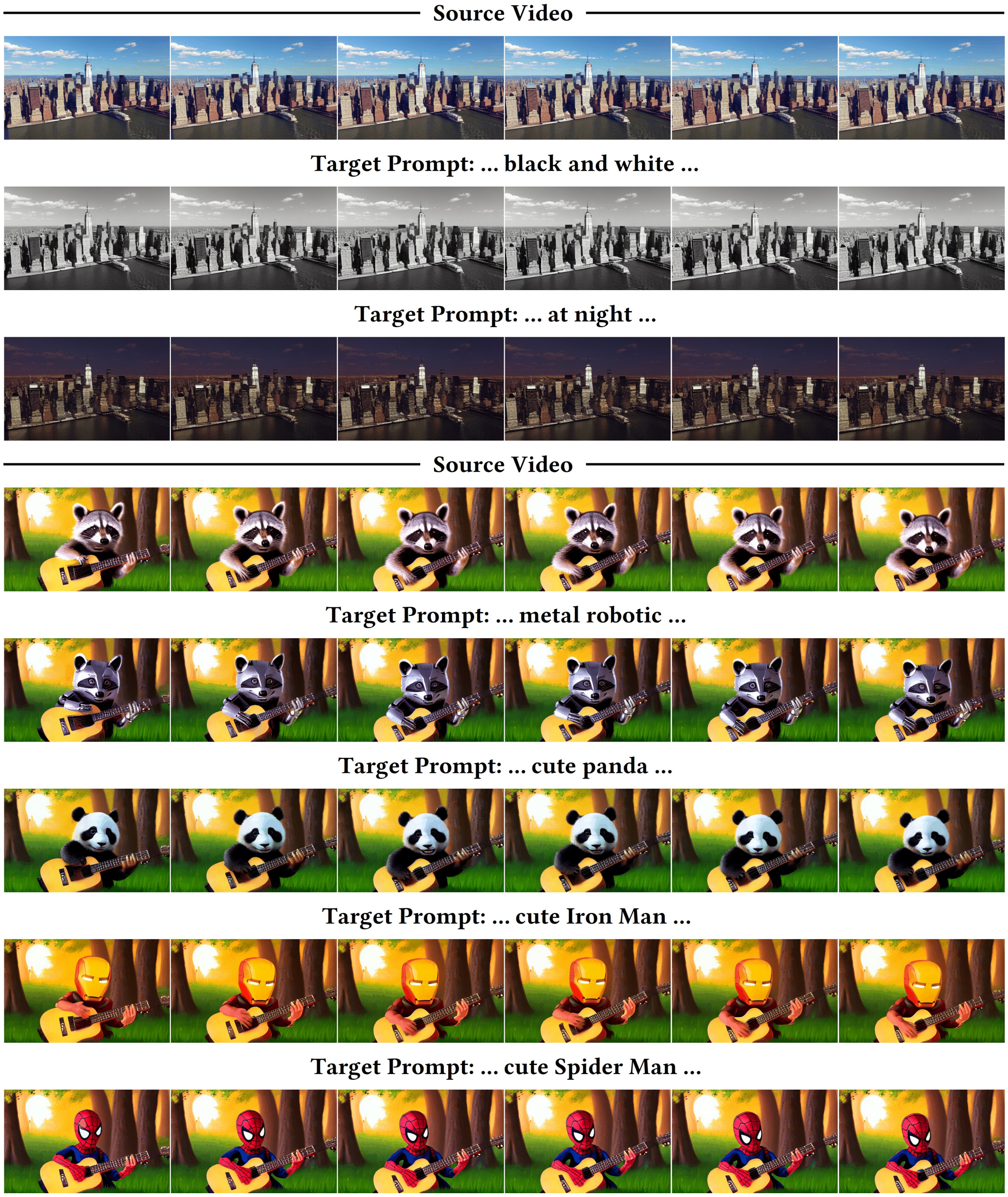
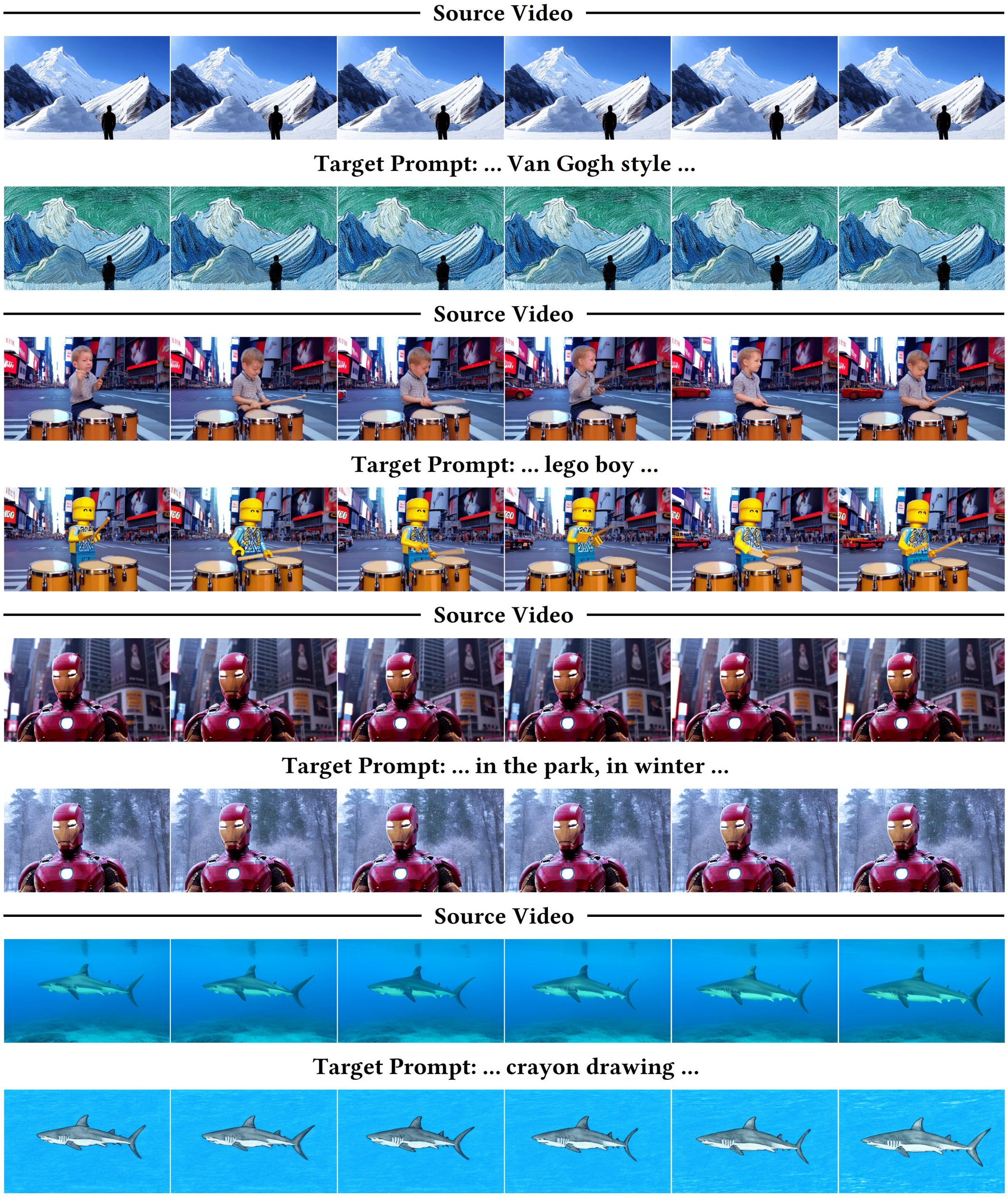

Source Video
Target Prompt: ... black and white ...
Target Prompt: ... at night ...
Source Video
Target Prompt: ... metal robotic ...
Target Prompt: ... cute panda ...
Target Prompt: ... cute Iron Man ...
Target Prompt: ... cute Spider Man ...

Figure 17: More appearance editing results of UniEdit.



Source Video

Target Prompt: ... Van Gogh style ...

Source Video

Target Prompt: ... lego boy ...

Source Video

Target Prompt: ... in the park, in winter ...

Source Video

Target Prompt: ... crayon drawing ...

Figure 18: More appearance editing results of UniEdit.

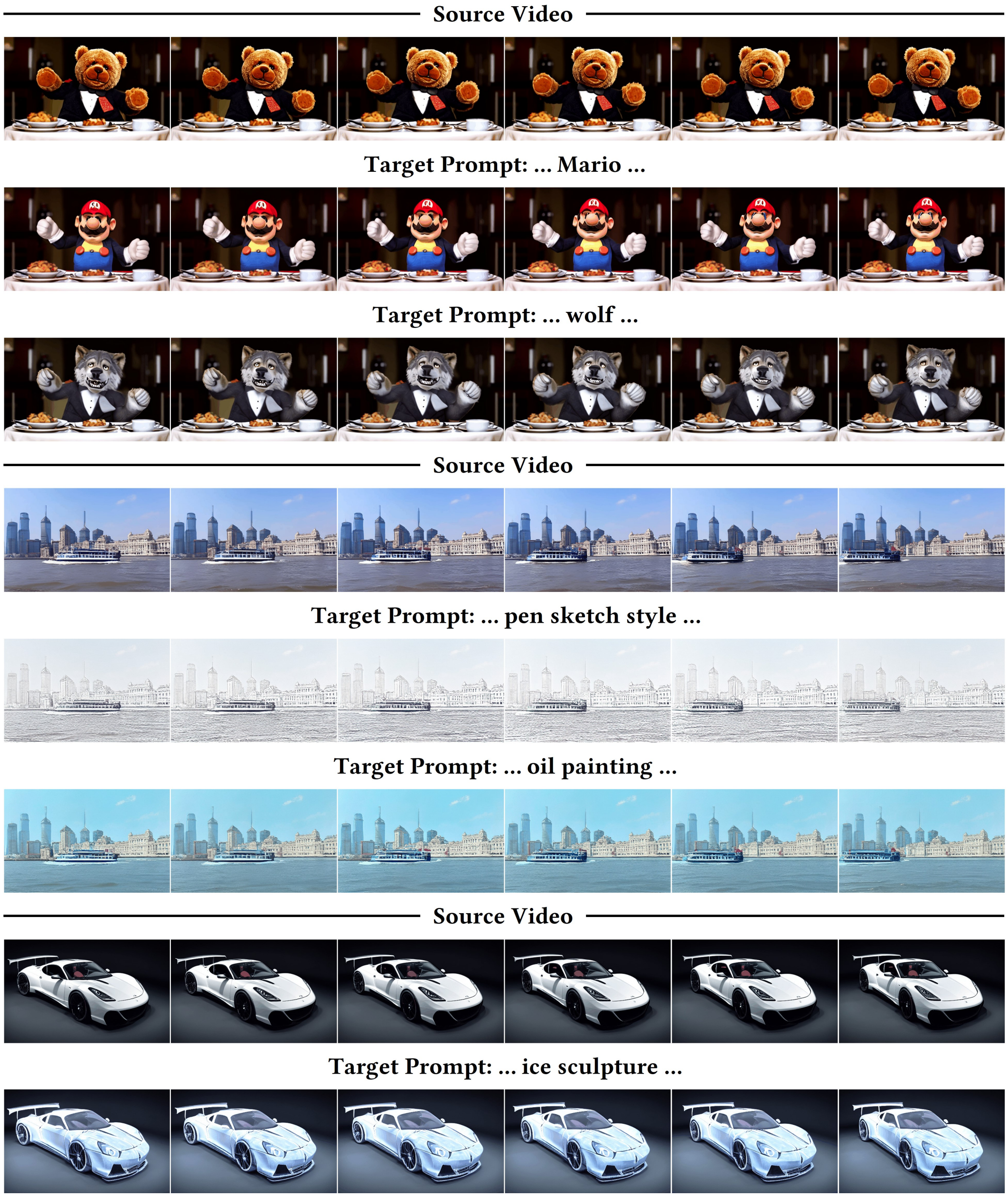
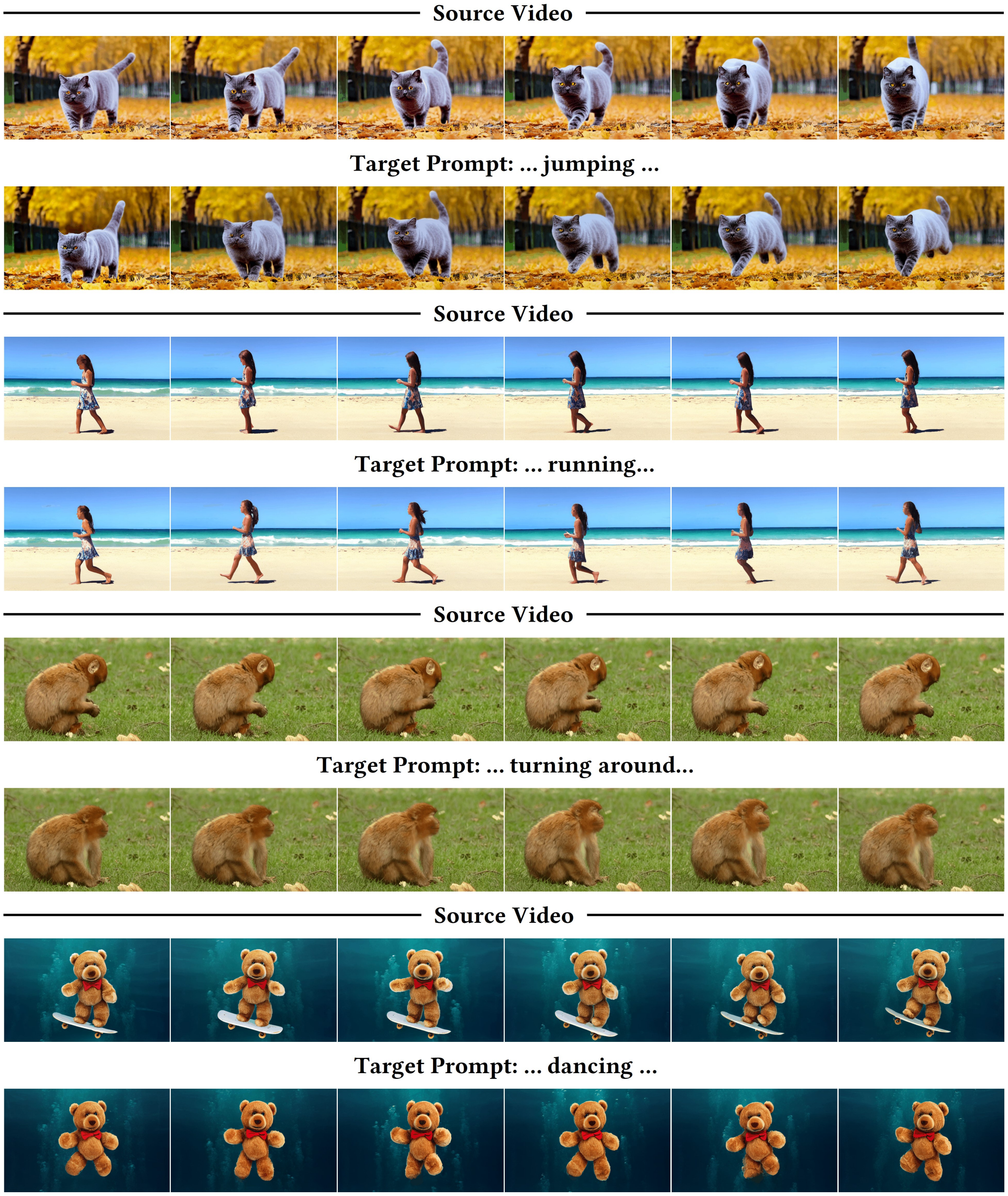

Source Video
Target Prompt: ... Mario ...
Target Prompt: ... wolf ...
Source Video
Target Prompt: ... pen sketch style ...
Target Prompt: ... oil painting ...
Source Video
Target Prompt: ... ice sculpture ...

Figure 19: More appearance editing results of UniEdit.



Source Video

Target Prompt: ... jumping ...

Source Video

Target Prompt: ... running...

Source Video

Target Prompt: ... turning around...

Source Video

Target Prompt: ... dancing ...

Figure 20: More motion editing results of UniEdit.

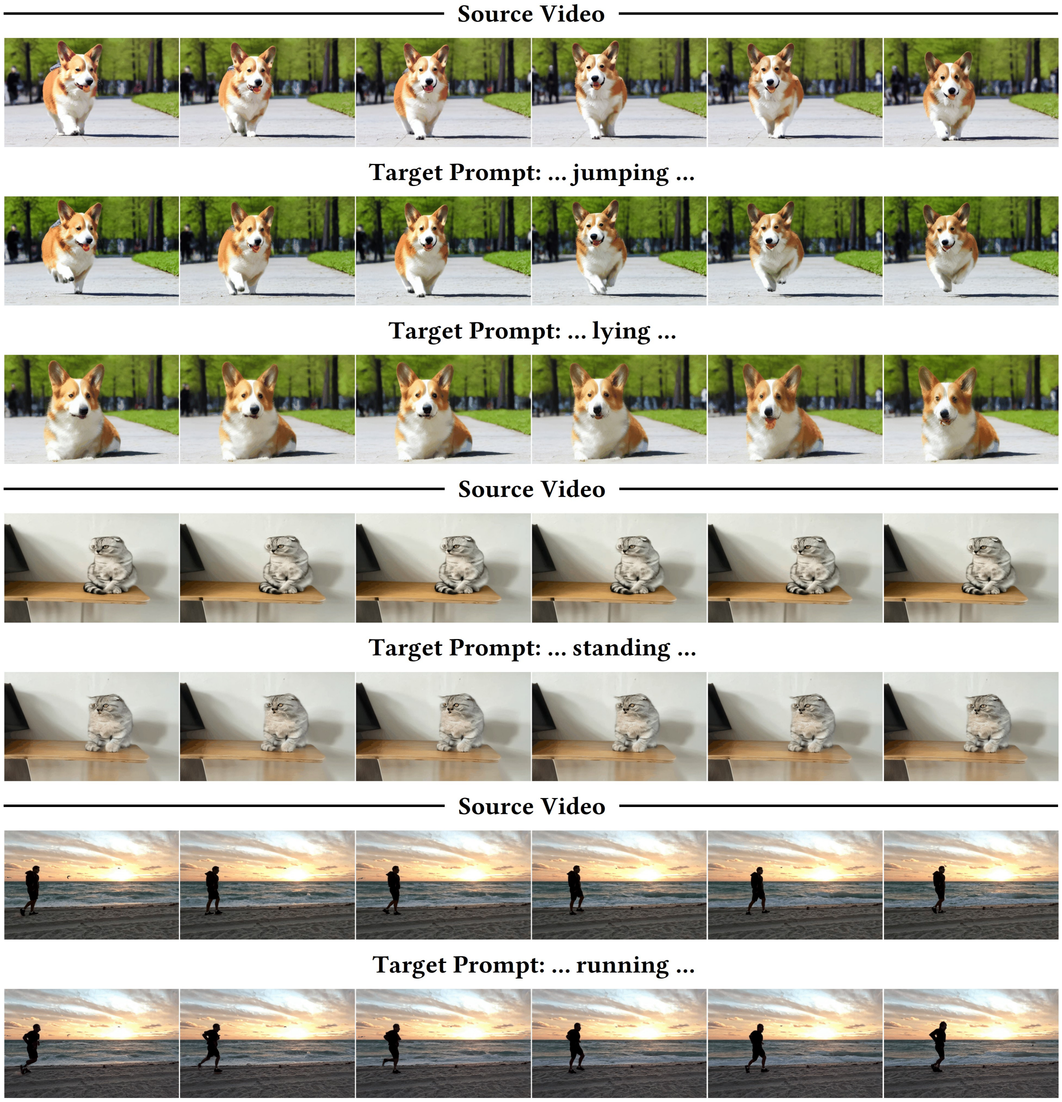

Source Video
Target Prompt: ... jumping ...
Target Prompt: ... lying ...
Source Video
Target Prompt: ... standing ...
Source Video
Target Prompt: ... running ...

Figure 21: More motion editing results of UniEdit.

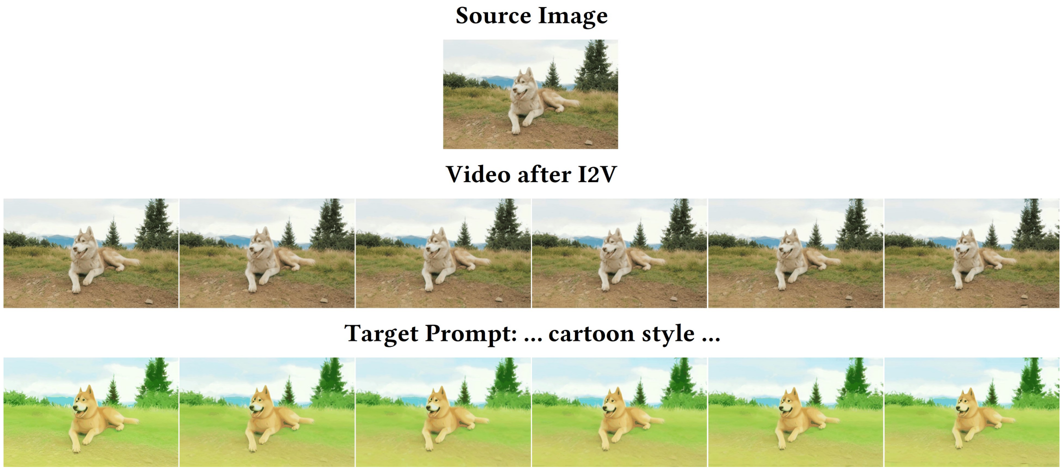


Figure 22: Results of text-image-to-video synthesis in Sec. 4.4.

C Broader Impacts

UniEdit is a tuning-free approach and is intended for advancing AI/ML research on video editing. We encourage users to use the model responsibly. We discourage users from using the codes to generate intentionally deceptive or untrue content or for inauthentic activities. It is suggested to add watermarks to prevent misuse.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In this work, we present UniEdit, a tuning-free framework that supports both video motion and appearance editing by harnessing the power of a pre-trained text-to-video generator within an inversion-then-generation framework.Extensive experiments demonstrate that UniEdit covers video motion editing and various appearance editing scenarios, and surpasses the state-of-the-art method.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the potential limitations of the method in Sec. 6 and presented failed cases in Appendix B.6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper aims to design a simple-and-effective video editing method named UniEdit, without focusing on theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper provides detailed information on the models, parameters, hyperparameter selection, computational resources in Sec. 5 and Appendix A to ensure reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Due to company policy reasons, we are currently unable to upload the code. **The code will be publicly available after the paper is published.**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper provides detailed information on the models, parameters, hyperparameter selection, computational resources in Sec. 5 and Appendix A to ensure reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The common practice in video editing does not including error bars, and we follow the previous papers.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provides detailed information on the computational resources in Sec. 5 and Appendix A and inference time comparison in Tab. 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research strictly adheres to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts are discussed in Appendix C.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, the creators or original owners of assets used in the paper are properly credited, and the license and terms of use are explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have uploaded the code of this paper to an anonymous repository and provided the corresponding link in Appendix. The code will be made publicly available after the paper is published.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.