2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3315-4364-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/CVPR52734.2025.00729

Visual Prompting for One-shot Controllable Video Editing without Inversion

Zhengbo Zhang[1*] Yuxi Zhou[2*] Duo Peng[1] Joo-Hwee Lim[3]
Zhigang Tu[2†] De Wen Soh[1] Lin Geng Foo[1]
[1]Singapore University of Technology and Design [2]Wuhan University
[3] Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore
https://vp4video-editing.github.io/

Abstract

One-shot controllable video editing (OCVE) is an important yet challenging task, aiming to propagate user edits that are made – using any image editing tool – on the first frame of a video to all subsequent frames, while ensuring content consistency between edited frames and source frames. To achieve this, prior methods employ DDIM inversion to transform source frames into latent noise, which is then fed into a pre-trained diffusion model, conditioned on the user-edited first frame, to generate the edited video. However, the DDIM inversion process accumulates errors, which hinder the latent noise from accurately reconstructing the source frames, ultimately compromising content consistency in the generated edited frames. To overcome it, our method eliminates the need for DDIM inversion by performing OCVE through a novel perspective based on visual prompting. Furthermore, inspired by consistency models that can perform multi-step consistency sampling to generate a sequence of content-consistent images, we propose a content consistency sampling (CCS) to ensure content consistency between the generated edited frames and the source frames. Moreover, we introduce a temporal-content consistency sampling (TCS) based on Stein Variational Gradient Descent to ensure temporal consistency across the edited frames. Extensive experiments validate the effectiveness of our approach.

1. Introduction

Video production plays a crucial role in creating compelling visuals for films, short videos, and various other media formats, its significance has rapidly increased amidst the ever-growing demand for high-quality video content. For instance, high-quality video production is often highly important for bloggers to create entertaining video vlogs on social platforms [61] or for filmmakers to generate captivating virtual scenes in films [62]. Yet, in many cases, both generated and real-world video content may fall short of meeting specific user requirements. As a result, there has been a significant increase in demand for convenient *video editing* tools that allow users to modify videos according to customized instructions.

Recently, diffusion-based video editing methods [7, 37, 39, 49, 64] have gained considerable attention. These approaches have demonstrated strong performance across various video editing tasks, such as visual style transfer [7] and character or object modification [49]. Additionally, they are highly user-friendly, often requiring only minor adjustments to the textual descriptions of the video content. Yet, although video editing via modifying textual descriptions offers convenience, this approach is a double-edged sword, as it typically facilitates global changes which often restricts the *controllability* of the editing. For instance, consider a scenario where social media users aim to make precise modifications to specific objects, structures or layouts in the video. Achieving such fine-grained control is challenging when relying solely on text-based descriptions. Therefore, controllable video editing methods are required to achieve the desired outcomes in such cases.

Nevertheless, achieving controllable video editing beyond text-based instructions is challenging, as it requires both high versatility and precision, including capabilities like accurately repositioning objects, erasing or adding specific elements, *etc*. To enable accurate and efficient controllable video editing, *one-shot controllable video editing* (OCVE) approaches [12, 20, 36] have been introduced. These approaches allow users to apply desired edits to the first frame of the source video using any off-the-shelf image editing tools (*e.g*., Photoshop, Paint, image editing diffusion models), and these approaches then propagate the edits to the remaining video frames. These OCVE methods commonly utilize *DDIM inversion* [55], which is a recursive process that converts the source video into latent noise, serving as latent representations that enable the diffusion

*Equal contribution.
†Corresponding author: tuzhigang@whu.edu.cn

2575-7075/25/$31.00 ©2025 IEEE
DOI 10.1109/CVPR52734.2025.00729

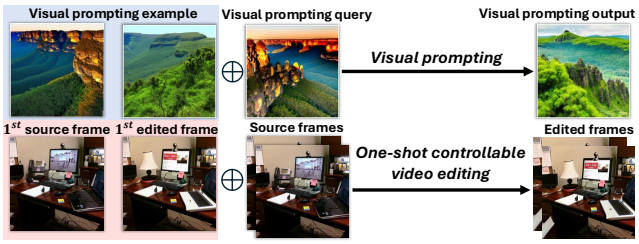

Figure 1. Visual prompting and one-shot controllable video editing share the goal of propagating certain modifications across images. In visual prompting, modifications made in the example (*e.g.*, changing the color of the mountain from golden to green) are transferred to the query, whereas in one-shot controllable video editing, modifications applied to the first edited frame are propagated to the subsequent source frames.

models to reconstruct the source video. Then, the inverted latents are combined with editing guidance (which indicates the target parts or objects to edit), and processed through the diffusion's sampling process to generate the edited video, thereby achieving controllable video editing. By leveraging the DDIM inversion in this manner, such OCVE approaches can achieve efficient editing while preserving the information of the source video.

Although the aforementioned DDIM inversion-based OCVE methods have made commendable progress, they still encounter the following two challenges. Firstly, the DDIM inversion introduces approximation errors at each timestep [55], and the accumulation of these errors often degrades the quality of the reconstructed video, which, in turn, diminishes the *content consistency* of the edited video and weakens the editing capabilities of DDIM inversion-based methods [66]. Secondly, after obtaining the inverted latents by DDIM inversion, when using image diffusion models [52] to generate edited video frames, the lack of strong temporal priors can lead to edited videos having poor *temporal consistency* [8, 68]. To tackle this, some methods [12, 36] utilize video diffusion models [5, 73] to provide temporal priors. However, the quality of these temporal priors is often insufficient, since the available open-sourced video datasets [2, 44, 67] used to train the video diffusion models tend to be of lower quality and size compared to the proprietary datasets used to train high-quality, closed-sourced models like Sora [46] and Kling [35]. Besides, video diffusion models are computationally demanding, resulting in the approaches based on these models being highly time-consuming.

Therefore, in this paper, we eschew the DDIM inversion process which can potentially introduce errors, and instead approach OCVE from a novel perspective, by treating it as a *visual prompting task* [3]. Our insight is that, both OCVE and visual prompting share the goal of propagating certain modifications across images (See Fig. 1). From this perspective, to tackle OCVE, we can consider the first source frame and the first edited frame (modified using any image editing tool), as the visual prompting example, with each remaining frame of the source video as a query. The visual prompting example and query are subsequently input into a diffusion model to obtain the edited frame, ensuring that the image pair – comprising the query and the output edited frame – remains consistent with the provided example. At the same time, we employ a pre-trained inpainting image diffusion model [1] to achieve it, leveraging its strong visual reasoning ability. Our method *bypasses the inaccurate DDIM inversion*, because the query frame that requires editing is directly input into the diffusion model in the encoded feature representation using the diffusion model's image encoder, rather than being inverted into latent noise by the DDIM inversion.

Additionally, to facilitate the edited frames to better maintain *content consistency* with the source frames, we draw inspiration from consistency models [57] which employ multi-step consistency sampling to generate a sequence of content-consistent images, and we introduce a content consistency sampling (CCS) method. Furthermore, we develop a temporal-content consistency sampling (TCS) method based on Stein Variational Gradient Descent (SVGD) [40] to ensure that the generated frames also exhibit good *temporal consistency*. Notably, compared to methods [12, 36] that depend on video diffusion models to provide temporal priors for preserving the temporal consistency of edited frames, our TCS is much faster while also providing quality improvements. Experimental results indicate that our method achieves strong performance in OCVE.

2. Related work

Video editing [4, 33] is a challenging task that aims to modify the content of a given video according to the user's intentions. Recently, inspired by the extensive knowledge contained in pre-trained diffusion models (e.g., Stable Diffusion [52]), researchers have widely explored leveraging diffusion models to facilitate video editing. Yet, most existing diffusion-based video editing approaches [9, 10, 18, 28, 30, 31, 34, 41, 47, 49, 51, 54, 63, 68, 69, 74–76] are predominantly text-driven, where videos are modified based mostly on textual instructions for tasks such as style editing [7], object editing [49], motion transfer [69], texture editing [10], *etc*. Since text-based video editing methods often face challenges in achieving fine-grained controllability, controllable video editing [11, 43, 70] methodologies have recently drawn significant attention. In particular, such controllable video editing approaches can be categorized according to their training data requirements: methods requiring abundant training data [45, 70], few-shot methods [11], and one-shot methods [12, 20, 36, 43].

In this paper, we focus on the *controllable one-shot set-*

ting (OCVE), which aims to edit the source video given only one edited frame (which shows the exact desired edits by the user). Existing one-shot methods [36, 43] often rely on DDIM inversion [55] to obtain the inverted latent for video editing, which facilitates video editing by enabling the diffusion model to first approximately reconstruct the source video. However, the inherent approximation errors in DDIM inversion limits the editing capabilities, compromising quality and content consistency in the edited video [66]. Different from previous works, we eschew DDIM inversion by approaching OCVE from a novel visual prompting perspective. Furthermore, we propose CCS, which is based on the multi-step consistency sampling of consistency models to improve content consistency, as well as TCS based on Stein Variational Gradient Descent [40] to improve temporal consistency. Overall, this results in significantly improved quality for OCVE.

Diffusion models [14, 26, 55, 56] have demonstrated remarkable success in image generation, by learning to progressively refine samples from a tractable noise distribution towards the target data distribution. Because of their impressive performance, researchers [1, 8, 15, 16, 19, 21–25, 30, 59, 66, 68] have applied diffusion models to a range of tasks, including image inpainting [1], image editing [29], pose estimation [13], video editing [30], and video generation [59], leading to significant advancements in these areas. In this paper, unlike previous works in video editing [30, 36], we leverage the strong visual reasoning ability of an image inpainting diffusion model [1] to perform OCVE through visual prompting. Besides, we modify the sampling process of the inpainting diffusion model to emulate the multi-step consistency sampling of consistency models, ensuring that the generated edited frames maintain content consistency with the source frames.

Consistency models [42, 57] are a new class of generative models designed for fast sampling, which allows for efficient one-step generation. A key characteristic of consistency models is self-consistency, which ensures that samples generated along a trajectory can be directly mapped back to their initial state. In addition to their one-step generation capabilities, consistency models also support multi-step consistency sampling [42, 57], which provides a trade-off between computational efficiency and sample quality, enabling the generation of a sequence of content-consistent images. In this work, for the first time, we modify the sampling equations of a pre-trained inpainting diffusion model to enable the multi-step consistency sampling without requiring additional training.

3. Preliminaries

OCVE is a challenging task, requiring the preservation of both the realism and consistency of the edited video with the source video, while editing the video according to the user's intentions. Given that diffusion models [52, 55] – trained on large datasets of real images and videos – exhibit a strong ability to generate highly realistic visuals, researchers [12, 36] often leverage their strong capabilities to address the challenging OCVE task. Notably, to enhance efficiency, the source video is processed in its latent representations, where the latent diffusion model is used. Below, we detail the process of mapping the source video into the latent space (*i.e.*, *inverting the video into a noise latent*) to obtain a source latent that aids the diffusion model in reconstructing the source video. We first provide an overview of how latent diffusion models typically generate samples, followed by how the inversion is often done.

Latent diffusion models produce target samples (z_0) via a progressive latent denoising process consisting of T recursive steps. Specifically, starting from $t = T$, the t-th denoising step denoises a latent at the t-th step (z_t) into a latent at the $(t-1)$-th step (z_{t-1}) as follows [26]:

$$z_{t-1} = \sqrt{\alpha_{t-1}} \cdot \underbrace{(z_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(z_t, t))/\sqrt{\alpha_t)}}_{\text{predicted } \hat{z}_0} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta(z_t, t)}_{\text{adjustment along } z_t} + \underbrace{\sigma_t \cdot \epsilon}_{\text{random noise}}, \tag{1}$$

where $\epsilon_\theta(z_t, t)$ is a noise prediction network parameterized by θ (often adopting a U-Net [53] autoencoder), $\alpha_{1:T} \in (0, 1)$ is a decreasing sequence of coefficients, $\sigma_{1:T}$ is a noise schedule, and $\epsilon \sim \mathcal{N}(0, I)$ is standard Gaussian noise independent of z_t. By applying Eq. 1 recursively for T steps, we eventually obtain z_0 as an output denoised latent.

Due to the iterative nature of the denoising diffusion process (from z_T to z_0) explained above, the straightforward inversion approach to map the source video into the latent space, *i.e.*, compute z_T from z_0, would also be an iterative process, where the t-th step is given by:

$$\begin{aligned} z_t =& (\sqrt{\alpha_t} \cdot z_{t-1} - \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta(z_t, t))/\sqrt{\alpha_{t-1}} \\ &+ \sqrt{1-\alpha_t} \cdot \epsilon_\theta(z_t, t) \end{aligned} \tag{2}$$

Here, the noise schedule σ_t is set to 0, eliminating the "random noise" item in Eq. (1) and transforming Eq. (1) into a deterministic forward process, which facilitates the inversion process. Yet, as evident from Eq. (2), directly performing the inversion process is impractical because the noise prediction network $\epsilon_\theta(\cdot, \cdot)$ requires the desired z_t as input.

DDIM inversion [55] has been proposed to solve this inversion issue, by assuming that the ordinary differential equation process can be reversed in the limit of infinitesimally small timesteps. Concretely, in the DDIM inversion process, $\epsilon_\theta(z_t, t)$ is replaced with $\epsilon_\theta(z_{t-1}, t)$ for the noise prediction in Eq. (2), which makes it tractable. However, although approximating $\epsilon_\theta(z_t, t)$ with $\epsilon_\theta(z_{t-1}, t)$ achieves inversion to some extent, this approximation introduces errors

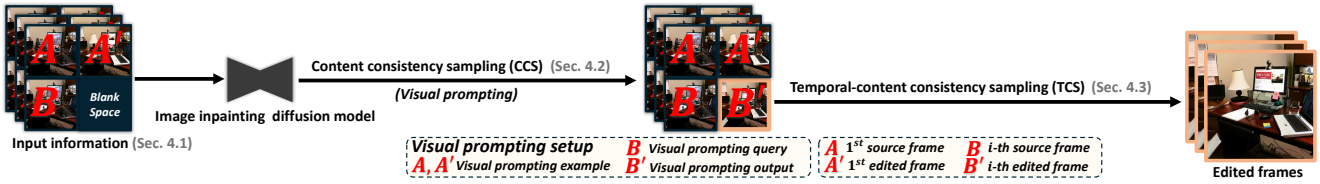

Figure 2. The overall pipeline of our method. **First**, to enable the image inpainting diffusion model to perform OCVE through visual prompting, we organize the example ($\boldsymbol{A}$ and $\boldsymbol{A'}$), query ($\boldsymbol{B}$), and output ($\boldsymbol{B'}$) from the visual prompting setup into a 2×2 square grid, which serves as the input information (Sec. 4.1) to the inpainting diffusion model. **Next**, we modify the sampling process of the inpainting diffusion model, and design a content consistency sampling (Sec. 4.2), to generate $\boldsymbol{B'}$ using the multi-step consistency sampling of the consistency models [57]. **Finally**, based on the generated $\boldsymbol{B'}$, we apply Temporal-content Consistency Sampling (Sec. 4.3) with Stein Variational Gradient Descent [40] to adjust the source frames, enhancing their temporal consistency and yielding the final edited frames in our framework.

at each timestep. Such cumulative errors degrade the reconstruction quality of latents $z_t, \ldots, z_1$, which can ultimately diminish the performance of video editing methods [66].

4. Method

In this work, to avoid relying upon DDIM inversion [55] which may introduce errors, for the first time, we approach OCVE from a visual prompting perspective (Sec. 4.1). See Fig. 2 for a summary of our full pipeline. In our method, edited frames are generated as the visual prompting output by using the source frame as the visual prompting query to prompt an image inpainting diffusion model with the visual prompting example (the first source frame and the first edited frame). To ensure the produced edited frames preserve content consistency with the source frames, we modify the sampling process of the inpainting diffusion model and propose a Content Consistency Sampling (CCS) (Sec. 4.2), leveraging the multi-step consistency sampling property of consistency models [57] which can generate a sequence of content-consistent images. Finally, to make the output edited frame of our method maintain temporal consistency, we perform a Temporal-content Consistency Sampling (TCS) (Sec. 4.3) based on Stein Variational Gradient Descent (SVGD) [40]. In TCS, the edited frames produced by CCS are adjusted to align with the source frames, which helps preserve the temporal consistency.

4.1. New perspective on OCVE

In this paper, we approach OCVE from a fresh perspective, adopting a visual prompting approach and eschewing the DDIM inversion step [55]. Our key insight is that: both OCVE and visual prompting can both be understood as tasks *focusing on the propagation of certain modifications* (See Fig. 1). In OCVE, the goal is to propagate user edits from the first frame to subsequent frames of the source video. In visual prompting, the aim is to propagate the modifications observed in an input-output image pair to the input query. With this in mind, we observe that *OCVE* can be re-casted as a type of a visual prompting task, where the given example consists of the edited and source versions of the first frame, and the query can be the subsequent frames of the source video. Notably, our approach does not require DDIM inversion because the source frames for editing are encoded as features, rather than as latent noise derived from the recursive DDIM inversion process. In this subsection, we introduce each part of our visual prompting-based pipeline in more detail.

Performing visual prompting with inpainting diffusion model. To tackle the challenging OCVE task, we follow prior methods [12, 36] to leverage the extensive knowledge encoded in a pre-trained diffusion model. The employed diffusion model is expected to have strong and robust visual reasoning capabilities, as we aim to perform visual prompting with the diffusion model, *i.e.*, the diffusion model needs to infer how to propagate the frames of the source video based only on the provided pair of example frames (first edited frame + first source frame). Here, we propose leveraging an image inpainting diffusion model for our work, as such models are well-suited for completing missing regions of an image while maintaining contextual consistency with the surrounding parts [1], demonstrating strong visual reasoning capabilities. Yet, utilizing the diffusion model, originally designed for inpainting, to perform OCVE through visual prompting is not straightforward. To achieve it, we derive inspiration from [21], and carefully *tailor the inputs* to the inpainting diffusion model. In the following, we first introduce the inputs of the inpainting diffusion model, and then detail the modifications made to these inputs.

Tailoring inputs of inpainting diffusion model. The image inpainting diffusion model is designed to achieve user-specified inpainting through a denoising process based on three input parameters. The parameters consist of: input information G to be inpainted, mask information M indicating the region in the input information G that requires inpainting, and a guiding text prompt p describing the desired inpainting result. To effectively harness the reasoning capability of the image inpainting diffusion model for

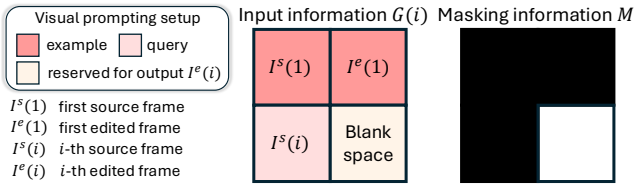

Figure 3. A visualization of the input information $G(i)$ (i denotes the i-th frame) and its corresponding mask information M.

OCVE through visual prompting, we design an inpainting strategy aligned with the model's original purpose, *i.e.*, image inpainting, to generate the desired output (edited video frames). As illustrated in Fig. 3, at the i-th source frame, the inpainting strategy organizes the input information $G(i)$ into four distinct regions: **(1)** the upper two regions hold the first frame before and after user editing, serving as the visual prompt example; **(2)** the bottom left region contains the i-th source frame, acting as the visual query; **(3)** the bottom right region is kept "blank", where we expect the diffusion model to generate the i-th edited frame via visual prompting based on the provided example and query. Additionally, Fig. 3 presents the corresponding mask information M for the input information $G(i)$. In the mask information M, white regions denote the areas in the input information $G(i)$ where inpainting by the diffusion model is needed, while black regions indicate the corresponding areas in $G(i)$ that should remain unaltered.

We next describe the design of the text prompt p, which is crucial for guiding the inpainting diffusion model toward generating the desired output, especially given the potential for multiple viable inpainting solutions. In our task, intuitively, the text prompt p may provide descriptions of the user's edits in the first frame. However, as described in Sec. 1, accurately describing the user's editing in text is often challenging. Given that vectors in the CLIP space can often effectively capture editing direction [48], we represent the user's editing as the difference between the encoded features of the first edited frame $I^e(1)$ and first source frame $I^s(1)$ in the CLIP embedding space [50]. Specifically, the user's editing, *i.e.*, "textual" prompt p is calculated as:

$$p = \lambda_1 \cdot \{E_{CLIP}(I^e(1)) - E_{CLIP}(I^s(1))\}, \quad (3)$$

where $E_{CLIP}(\cdot)$ represents the image encoder of CLIP, and λ_1 is a hyper-parameter. Since our adopted inpainting diffusion model is built on the text encoder of CLIP, we can directly use p as a "text" prompt to guide the diffusion model.

4.2. Content consistency sampling (CCS)

In the previous subsection, we tackle OCVE from a visual prompting perspective, effectively bypassing the DDIM inversion step by generating the edited frames with an inpainting diffusion model. Ideally, the content of the generated edited frame should be based directly on the source frame to preserve content consistency between them. However, in our method, the content of the generated edited frame is based on the latent noise from the previous denoising timestep, as our method builds upon the image inpainting diffusion model, where the edited frame is generated through a progressive denoising process [26] (see Eq. (1)). Moreover, since our method does not incorporate DDIM inversion, the initial noise in our progressive denoising process is not the noise obtained through DDIM inversion that approximately preserves the original image content. As a result, our method may face challenges in ensuring content consistency between the generated edited frame and the source frame.

To handle this, we get inspiration from consistency models, where their multi-step consistency sampling [57, 65] can generate content-consistent images by basing each timestep's output image on the previous timestep's generated image. Based on this property, we propose a Content Consistency Sampling (CCS), which is a multi-step consistency sampling built upon the progressive denoising sampling of the inpainting diffusion model. Specifically, to maintain content consistency between the CCS-generated edited frame and the source frame, we artificially configure CCS to generate the source frame in the first time step. A noise calibration mechanism is then applied to guide the sampling process, allowing the generated images to gradually transition from the source frame to the desired edited frame. Note that CCS is a sampling approach, and thus does not require additional training.

Below, we first introduce the special sampling process of CCS, then we describe how our introduced sampling process is utilized to generate the desired edited frame with improved content consistency.

Multi-step consistency sampling based on inpainting diffusion model. In the multi-step consistency sampling of consistency models, the content of the generated image at each timestep is based on the output image from the previous denoising timestep, thereby generating a sequence of content-consistent images [42]. Here, we aim to utilize this property to generate a sequence of content-consistent images, beginning with the source frame and progressively shifting the content along the user's intended editing direction, ultimately generating a desired edited frame while preserving content-consistency with the source frame. However, the sampling process in our inpainting diffusion model (Eq. (1)) can struggle to generate a sequence of content-consistent images, due to its Markovian denoising nature [26], which operates without reference to the source frame content. Therefore, we seek to adapt the inpainting diffusion model's denoising sampling into the multi-step consistency sampling. To this end, we first modify the inpainting diffusion model's sampling (Eq. (1)), so that it is no longer an iterative denoising Markov process. Then, we

enable the modified sampling process to generate a latent of output image ($\hat{z}_0$) at each timestep, where the content of each generated latent is based on the content of the latent produced in the previous timestep, thereby yielding a sequence of content consistent images. These steps are detailed below.

Specifically, to eliminate the Markovian denoising nature in the sampling of our inpainting diffusion model, we remove the adjustment term (the second term) in Eq. (1) by setting the noise parameter $\sigma_t = \sqrt{1-\alpha_{t-1}}$, resulting in the following sampling:

$$z_{t-1} = \sqrt{\alpha_{t-1}} \cdot \underbrace{(z_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(z_t, t))/\sqrt{\alpha_t}}_{\textbf{predicted } \hat{\mathbf{z}}_0} + \underbrace{\sqrt{1-\alpha_{t-1}} \cdot \epsilon}_{\textbf{random noise}}, \epsilon \sim \mathcal{N}(0, I). \tag{4}$$

Next, we consider the "predicted $\hat{z}_0$" as the output of the sampling process at each timestep, rather than the denoised latent (*e.g.*, z_{t-1} in Eq. (4)), enabling the sampling process to output the latent of output image ($\hat{z}_0$) at each timestep following the multi-step consistency sampling. Besides, in the multi-step consistency sampling, the generated latent ($\hat{z}_0$) across different timesteps should be consistent [57]. However, the "predicted $\hat{z}_0$" at different timesteps are generally independent of each other, which may fail to ensure consistency across timesteps [38]. To solve it, we draw inspiration from [65], and introduce a consistency noise ϵ^c to replace the parameterized noise ϵ_θ in the "predicted $\hat{z}_0$" term, ensuring that $\hat{z}_0$ generated by the new term maintains consistency across different timesteps. Specifically, the new term is defined as:

$$\hat{f}(z_t, t, \epsilon^c(t)) = (z_t - \sqrt{1-\alpha_t} \cdot \epsilon^c(t))/\sqrt{\alpha_t}, \tag{5}$$

and we have $\hat{z}_0^{(t)} = \hat{f}(z_t, t, \epsilon^c(t))$. In fact, $\hat{f}$ serves as a consistency model that can perform the multi-step consistency sampling. Hence, by substituting Eq. (5) into Eq. (4), we obtain the special multi-step consistency sampling, performed by iteratively executing the following process:

$$\begin{aligned} \hat{z}_{t-1} &= \sqrt{\alpha_{t-1}} \cdot \hat{z}_0^{(t)} + \sqrt{1-\alpha_{t-1}} \cdot \epsilon, \epsilon \sim \mathcal{N}(0, I) \\ \hat{z}_0^{(t-1)} &= \hat{f}(\hat{z}_{t-1}, t-1, \epsilon^c(t-1)). \end{aligned} \tag{6}$$

In this sampling process, the latent of the output frame generated at the current timestep (*e.g.*, $\hat{z}_0^{(t-1)}$) is based on the latent of the output frame generated at the previous timestep ($\hat{z}_0^{(t)}$), thereby allowing the generation of a sequence of video frames that maintain content consistency.

Generating desired edited frames. After deriving the special multi-step consistency sampling, we aim to to generate the desired edited frames by performing CCS. To this end, the sampling process of CCS starts by generating the source frame at the first timestep, ensuring that the generated edited frame maintains content consistency with the source frame. Next, a noise calibration mechanism is employed within the sampling process to gradually transform the generated frames from the source frame to the desired edited frame. We detail these steps below.

First, to ensure that CCS generates the source frame at the first timestep, it is necessary to determine the corresponding consistency noise ϵ^c. Since $\hat{f}$ functions as a consistency model, we have $z_0^s = \hat{f}(\hat{z}_t, t, \epsilon^c(t; z_0^s))$, where z_0^s represents the latent of the source frame. Hence, from this equation, we can obtain the corresponding noise $\epsilon^c(t; z_0^s) = (\hat{z}_t - \sqrt{\alpha_t} \cdot z_0^s)/\sqrt{1-\alpha_t}$.

Next, we aim to guide this multi-step consistency sampling to progress along the user's intended editing direction (*i.e.*, guiding the content of the images generated by CCS to evolve in line with the user's intended edits), thereby generating the desired edited frame. Notably, this progression is reflected in the sampling process of the inpainting diffusion model used to generate the edited frame (Sec. 4.1). To be specific, in this sampling process, all regions initially receive the random Gaussian noise. However, the lower-left area of the input information $G(i)$ is progressively denoised toward the source frame, while the lower-right area is denoised toward the desired edited frame. The denoising difference between these two regions at each timestep effectively captures the divergence between the source and edited frames, actually reflecting the user's intended editing direction. Hence, we utilize this denoising difference to guide CCS in generating images that evolve along the user's intended editing direction, ultimately generating the desired edited frame while maintaining content consistency with the source frame.

Specifically, the denoising difference $\Delta\epsilon_t$ is calculated as: $\Delta\epsilon_t = \epsilon_\theta(z_t(I^e), t) - \epsilon_\theta(z_t(I^s), t)$, where $z_t(I^e)$ and $z_t(I^s)$ denote the latent in the lower-right and lower-left regions of z_t at timestep t, respectively. We then add the denoising difference $\Delta\epsilon_t$ to the noise term ϵ_c of the consistency model $\hat{f}$ (Eq. (5)) in CCS at each timestep, as follows:

$$\hat{f}(\hat{z}_t, t, \epsilon^c(t; z_0^s)) = (\hat{z}_t - \sqrt{1-\alpha_t}(\epsilon^c(t; z_0^s) + \lambda_2 \Delta\epsilon_t))/\sqrt{\alpha_t} \tag{7}$$

where λ_2 denotes a hyper-parameter. Notably, CCS is performed on the lower-right subregion of the input information $G(i)$.

4.3. Temporal-content consistency sampling (TCS)

Based on the CCS (Sec. 4.2), we can perform OCVE without relying on DDIM inversion, and generate edited frames that maintain content consistency with the source frames. However, CCS does not explicitly ensure the preservation of temporal consistency between source frames during its sampling process, which may lead to *temporal inconsistency and unsmooth edited clips*. We address it by proposing a Temporal-consistent Consistency Sampling (TCS), which is performed following the completion of CCS.

To achieve this, we first explicitly model the temporal consistency of the source video by treating the video as a distribution, where each source frame is considered a sample drawn from this distribution. These source samples are constrained by their mutual relationships, *i.e.*, temporal consistency. Since we want the edited frames (produced by CCS) to emulate the temporal consistency of the source frames, we constrain the edited frames to approximate the distribution of the source frames. This process can essentially be computed via Bayesian inference [6]. Given the complexity of video data, this inference occurs in a high-dimensional space, where the curse of dimensionality presents a long-standing challenge [60]. To overcome it, we get inspiration from [34], and employ Stein Variational Gradient Descent (SVGD) [40] to enhance the updating process, as SVGD offers the steepest descent for the updating, transforming the complex high-dimensional Bayesian inference into a deterministic updating process. Below, we provide details on the SVGD-enhanced updating (TCS).

Updating progress based on SVGD. We consider the N source frames as N samples, $\{z(i)\}_{i=1}^{N}$, drawn from a distribution. We note that, as our method is based on the latent diffusion model, these samples $\{z(i)\}_{i=1}^{N}$ are in fact latents sampled from the latent space. We aim to update the CCS-generated samples $\{\hat{z}_0^{(0)}(i)\}_{i=1}^{N}$ (for simplicity, below we denote $\{\hat{z}_0^{(0)}(i)\}_{i=1}^{N}$ as $\{\hat{z}^{(0)}(i)\}_{i=1}^{N}$) based on SVGD to approximate the samples $\{z(i)\}_{i=1}^{N}$ of source frames, thereby resulting in edited frames with good temporal consistency. We refer to this updating process as TCS, which is performed after CCS is completed. Specifically, our TCS, based on SVGD, follows a deterministic process consisting of $\mathcal{L}$ recursive steps. Beginning from $\ell = \mathcal{L}$, the ℓ-th step of the TCS denoises the latent representation $\hat{z}_{\ell}^{(0)}(i)$ of the i-th sample ($i \in [1, \ldots, N]$) into the latent $\hat{z}_{\ell-1}^{(0)}(i)$ of the i-th sample at the $(\ell-1)$-th step:

$$\hat{z}_{\ell-1}^{(0)}(i) = \hat{z}_{\ell}^{(0)}(i) - \eta \cdot \hat{\phi}\big(\hat{z}_{\ell}^{(0)}(i)\big), \text{where } \hat{\phi}(z) = \frac{1}{N}\sum_{j=1}^{N}\Big[K\big(\hat{z}_{\ell}^{(0)}(j), z\big)\big(\hat{z}_{\ell}^{(0)}(j) - z(j)\big) + \nabla_{\hat{z}_{\ell}^{(0)}(j)} K\big(\hat{z}_{\ell}^{(0)}(j), z\big)\Big]. \tag{8}$$

Here, η is the step size, and $K(\cdot,\cdot)$ is a standard Radial Basis Function (RBF) kernel.

In high-dimensional space, updating each sample based solely on its own gradient may lead to unstable optimization and increase the risk of getting trapped in local optima [17]. Therefore, we follow SVGD to use an averaged gradient across all N samples (represented by the first term in $\hat{\phi}(z)$) to update each sample. The second term in $\hat{\phi}(z)$ serves as a repulsive force to prevent mode collapse among the samples, further contributing to the stability of the update process [40]. The algorithm for our method is provided in *supplementary*.

5. Experiments

5.1. Experiment setup

Implementation details. We use Stable Diffusion Inpainting 1.5 [1] as the image inpainting diffusion model. CCS operates with 30 timesteps, and TCS utilizes 50 timesteps. We set $\lambda_1 = 0.7$, $\lambda_2 = 1.2$, and $\eta = 2.0$. We follow [21] to use the self-attention cloning in our CCS. All experiments are conducted on a A100 GPU.

Datasets. Following Videoshop [12], our method is evaluated on a large-scale generated video dataset derived from MagicBrush dataset [72], which consists of 10388 tuples in the format *(source video, editing instruction, the first edited frame)*. These tuples represent a wide range of editing types, including object addition, replacement, removal, and modifications in action, color, and texture, *etc*.

Baselines. Our method is compared with 2 state-of-the-art (SOTA) OCVE methods (Videoshop [12], AnyV2V [36]). Following Videoshop, we also compare our method with 5 SOTA text-based video editing methods: Pix2Video [7], Fatezero [49], Spacetime [69], RAVE [31], and BDIA [71].

Evaluation metrics. Following Videoshop [12], we evaluate our method from 4 perspectives. **1)** Edit fidelity: We measure CLIP_{tgt} similarity [50] between each edited frame and the first edited image. We use the TIFA score [27] to assess the semantic alignment between the first edited frame and subsequent edited frames in the video. **2)** Source faithfulness: We measure CLIP_{src} similarity between the source and edited videos. Flow score [58] is employed to evaluate motion faithfulness. The FVD and SSIM scores are used to assess the overall quality of the edited videos and the quality of edited frames, respectively. **3)** Temporal consistency: We measure the average CLIP similarity between adjacent frames, referred to as CLIP_{TC}. **4)** Efficiency: We measure the average time taken by each video editing method to process a video. **5)** Human evaluation: We ask human evaluators to compare the editing quality of our method with that of the baseline. For more details of the metrics, please refer to our *supplementary*.

5.2. Quantitative results

We compare our method with SOTA video editing methods [12, 36] on the generated dataset, as shown in Tab. 1. Our proposed method achieves the best performance across multiple metrics, demonstrating its effectiveness. The time metric clearly shows that our method is significantly more efficient compared to previous OCVE methods [12, 36]. This efficiency arises from our use of a more streamlined image diffusion model, rather than the video diffusion models used in earlier OCVE approaches. Furthermore, SVGD used in our CCS method, which ensures temporal consistency across the edited frames, is also efficient [40]. We provide results of the human evaluation in *supplementary*.

Table 1. Quantitative comparisons of our method with baselines. Best results are highlighted. "+" indicates that the metric is used to evaluate the edited region, whereas "−" indicates that the metric is used to evaluate the unedited region. Following Videoshop [12], we use Cotracker [32] to identify the edited and unedited regions. (T.C. = Temporal Consistency; E. = Efficiency)

Method	Edit Fidelity			Source Faithfulness						T.C.	E.
	$\text{CLIP}_{\text{tar}}\uparrow$ $(\times10^{-2})$	$\text{CLIP}^{+}_{\text{tar}}\uparrow$ $(\times10^{-2})$	TIFA $\uparrow$ $(\times10^{-2})$	$\text{CLIP}_{\text{src}}\uparrow$ $(\times10^{-2})$	$\text{CLIP}^{+}_{\text{src}}\uparrow$ $(\times10^{-2})$	Flow $\downarrow$ $(\times10^{-1})$	$\text{Flow}^{-}\downarrow$ $(\times10^{-1})$	FVD $\downarrow$ $(\times10^{2})$	SSIM $\uparrow$ $(\times10^{-2})$	$\text{CLIP}_{\text{TC}}\uparrow$ $(\times10^{-2})$	time (s) $\downarrow$ $(\times10^{0})$
BDIA [71]	82.1	82.2	57.7	82.5	87.1	28.3	14.3	34.8	49.7	94.4	35
Pix2Video [7]	71.2	76.5	52.0	74.6	79.0	35.9	25.8	29.9	59.1	94.5	157
Fatezero [49]	84.9	79.1	55.4	92.4	86.9	44.2	31.1	22.1	48.6	95.7	25
Spacetime [69]	63.9	75.2	46.3	65.7	71.9	82.4	56.2	48.2	41.6	96.6	135
RAVE [31]	74.7	78.6	51.1	76.0	80.2	33.5	24.2	23.5	62.2	96.6	45
AnyV2V [5]	87.1	85.9	67.0	91.3	94.2	24.6	14.1	17.1	65.5	93.9	149
Videoshop [12]	88.8	85.6	64.4	91.0	94.8	19.0	7.8	14.8	71.9	95.2	32
Ours w/o CCS	80.3	77.6	55.8	81.3	82.7	23.7	10.9	25.1	59.8	95.8	19
Ours w/o TCS	89.8	86.1	68.3	92.8	95.1	33.1	20.5	23.7	69.5	89.8	18
Ours	90.1	88.2	69.1	93.2	96.6	21.9	9.2	15.2	69.2	97.1	19

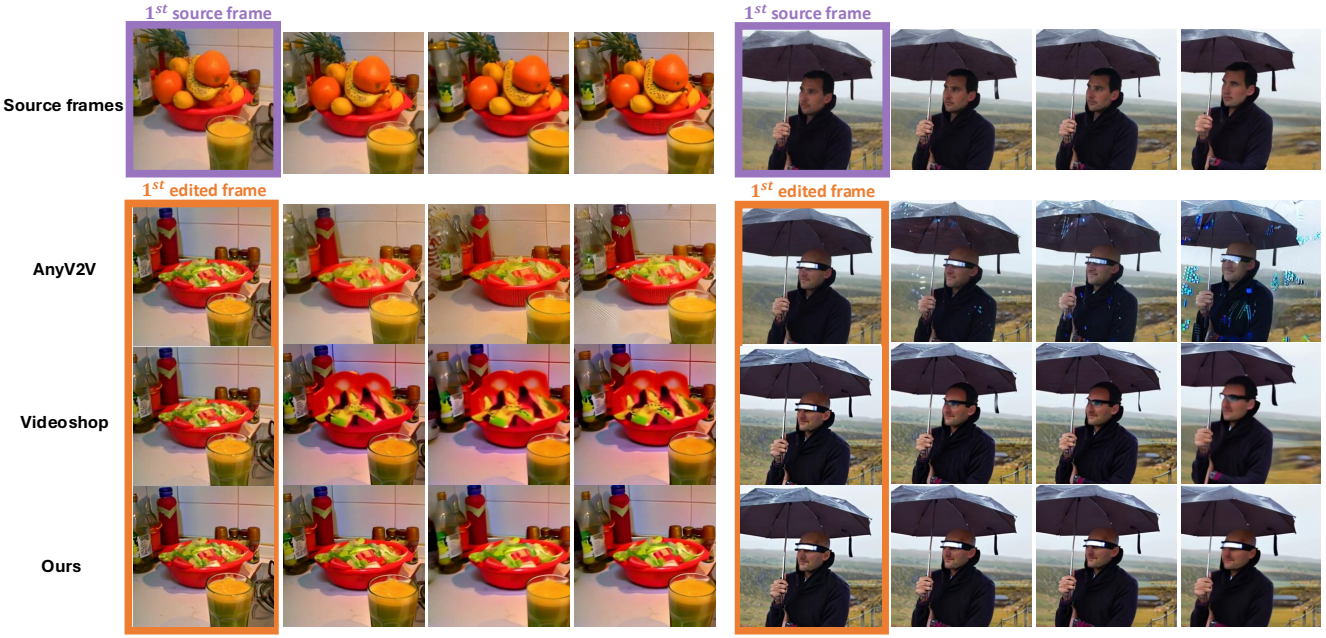

Figure 4. The visual comparison includes our method alongside two SOTA OCVE methods (AnyV2V [36] and Videoshop [12]), evaluated across two distinct types of editing. On the left, user modifications consist of replacing the fruit in the basin with vegetables. On the right, the user edits involve: 1) removing the individual's hair and 2) adding glasses.

5.3. Qualitative results

As shown in Fig. 4, we evaluate our method with 2 SOTA OCVE methods [12, 36] on two types of edits: object replacement and object removal/addition. Our method achieves the best performance in both cases. For instance, in the edit where fruit is replaced with vegetables, Videoshop struggles to maintain the fruit's appearance. In contrast, our method, leveraging CCS, consistently preserves the appearance of the fruit across the edited frames.

5.4. Ablation studies

We evaluates two variants of our method: "Ours w/o CCS" and "Ours w/o TCS" (see Tab. 1). By comparing the source faithfulness metrics between "Ours w/o CCS" and "Ours", we observe a performance drop in "Ours w/o CCS", indicating that our designed CCS effectively enhances content consistency between the edited frames and the source frames. The temporal consistency metric for "Ours w/o TCS" shows a significant decrease, demonstrating that TCS effectively improves temporal consistency across the edited frames.

6. Conclusion

In this paper, we perform OCVE via visual prompting, eschewing the DDIM inversion process which can potentially introduce errors. Our method comprises CCS and TCS, which ensure content consistency between the edited and source frames, as well as maintain temporal consistency throughout the edited frames. Both quantitative and qualitative experimental results validate the efficacy of our method.

References

[1] Stability AI. Stable diffusion inpainting. https://github.com/Stability-AI/stablediffusion, 2022. 2, 3, 4, 7

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 2

[3] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 2

[4] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022. 2

[5] 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. 2, 8

[6] George EP Box and George C Tiao. *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011. 7

[7] 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. 1, 2, 7, 8

[8] 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. 2, 3

[9] 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. 2

[10] Paul Couairon, Clément Rambour, Jean-Emmanuel Haugeard, and Nicolas Thome. Videdit: Zero-shot and spatially aware text-driven video editing. *Transactions on Machine Learning Research*, 2023. 2

[11] 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. 2

[12] Xiang Fan, Anand Bhattad, and Ranjay Krishna. Videoshop: Localized semantic video editing with noise-extrapolated diffusion inversion. In *European conference on computer vision*. Springer, 2024. 1, 2, 3, 4, 7, 8

[13] Lin Geng Foo, Jia Gong, Hossein Rahmani, and Jun Liu. Distribution-aligned diffusion for human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9221–9232, 2023. 3

[14] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey. *arXiv preprint arXiv:2308.14177*, 2023. 3

[15] Lin Geng Foo, Tianjiao Li, Hossein Rahmani, and Jun Liu. Action detection via an image diffusion process. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18351–18361, 2024. 3

[16] Lin Geng Foo, Yixuan He, Ajmal Saeed Mian, Hossein Rahmani, Jun Liu, and Christian Theobalt. Avatar concept slider: Controllable editing of concepts in 3d human avatars, 2025. 3

[17] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015. 7

[18] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 2

[19] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 3

[20] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. Videoswap: Customized video subject swapping with interactive semantic point correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7621–7630, 2024. 1, 2

[21] Zheng Gu, Shiyuan Yang, Jing Liao, Jing Huo, and Yang Gao. Analogist: Out-of-the-box visual in-context learning with image diffusion model. *arXiv preprint arxiv:2405.10316*, 2024. 3, 4, 7

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

[23] Ziyan Guo, Zeyu Hu, Na Zhao, and De Wen Soh. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. *arXiv preprint arXiv:2502.02358*, 2025.

[24] Ziyan Guo, Haoxuan Qu, Hossein Rahmani, Dewen Soh, Ping Hu, Qiuhong Ke, and Jun Liu. Tstmotion: Training-free scene-aware text-to-motion generation. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2025.

[25] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3, 5

[27] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417, 2023. 7

[28] 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. 2

[29] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[30] Kumara Kahatapitiya, Adil Karjauv, Davide Abati, Fatih Porikli, Yuki M Asano, and Amirhossein Habibian. Object-centric diffusion for efficient video editing. *arXiv preprint arXiv:2401.05735*, 2024. 2, 3

[31] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6507–6516, 2024. 2, 7, 8

[32] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 8

[33] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 2

[34] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative score distillation for consistent visual editing. *Advances in Neural Information Processing Systems*, 36:73232–73257, 2023. 2, 7

[35] KlingAI. Kling. https://klingai.com/, 2024. 2

[36] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A plug-and-play framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. 1, 2, 3, 4, 7, 8

[37] Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14317–14326, 2023. 1

[38] Tong Li, Hansen Feng, Lizhi Wang, Zhiwei Xiong, and Hua Huang. Stimulating the diffusion model for image denoising via adaptive embedding and ensembling. *arXiv preprint arXiv:2307.03992*, 2023. 6

[39] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. Magicedit: High-fidelity and temporally coherent video editing. *arXiv preprint arXiv:2308.14749*, 2023. 1

[40] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016. 2, 3, 4, 7

[41] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8599–8608, 2024. 2

[42] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3, 5

[43] Yue Ma, Xiaodong Cun, Yingqing He, Chenyang Qi, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Magicstick: Controllable video editing via control handle transformations. *arXiv preprint arXiv:2312.03047*, 2023. 2, 3

[44] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 2

[45] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. *arXiv preprint arXiv:2405.13865*, 2024. 2

[46] OpenAI. Sora. https://openai.com/index/sora/, 2024. 2

[47] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8089–8099, 2024. 2

[48] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 5

[49] 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*, pages 15932–15942, 2023. 1, 2, 7, 8

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5, 7

[51] Yixuan Ren, Yang Zhou, Jimei Yang, Jing Shi, Difan Liu, Feng Liu, Mingi Kwon, and Abhinav Shrivastava. Customize-a-video: One-shot motion customization of text-to-video diffusion models. *arXiv preprint arXiv:2402.14780*, 2024. 2

[52] 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. 2, 3

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

[54] Chaehun Shin, Heeseung Kim, Che Hyun Lee, Sang-gil Lee, and Sungroh Yoon. Edit-a-video: Single video editing with object-aware consistency. In *Asian Conference on Machine Learning*, pages 1215–1230. PMLR, 2024. 2

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2, 3, 4

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3

[57] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 2, 3, 4, 5, 6

[58] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 7

[59] 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. 3

[60] Yifei Wang, Peng Chen, and Wuchen Li. Projected wasserstein gradient descent for high-dimensional bayesian inference. *SIAM/ASA Journal on Uncertainty Quantification*, 10 (4):1513–1532, 2022. 7

[61] Tony Wibowo and Lisanto Lisanto. Cinematic sequence for video blog using multimedia development life cycle. *Journal of Information System and Technology (JOINT)*, 2(2):16–48, 2021. 1

[62] Hui-Yin Wu, Quentin Galvane, Christophe Lino, and Marc Christie. Analyzing elements of style in annotated film clips. In *WICED 2017-Eurographics Workshop on Intelligent Cinematography and Editing*, pages 29–35. The Eurographics Association, 2017. 1

[63] 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. 2

[64] Zhen Xing, Qi Dai, Han Hu, Zuxuan Wu, and Yu-Gang Jiang. Simda: Simple diffusion adapter for efficient video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7827–7839, 2024. 1

[65] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. *arXiv preprint arXiv:2312.04965*, 2023. 5, 6

[66] Yangyang Xu, Wenqi Shao, Yong Du, Haiming Zhu, Yang Zhou, Ping Luo, and Shengfeng He. Task-oriented diffusion inversion for high-fidelity text-based editing. *arXiv preprint arXiv:2408.13395*, 2024. 2, 3, 4

[67] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022. 2

[68] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 2, 3

[69] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8476, 2024. 2, 7, 8

[70] Jaehong Yoon, Shoubin Yu, and Mohit Bansal. Raccoon: Remove, add, and change video content with auto-generated narratives. *arXiv preprint arXiv:2405.18406*, 2024. 2

[71] G. Zhang, J. P. Lewis, and W. B. Kleijn. Exact diffusion inversion via bi-directional integration approximation. In *European conference on computer vision*. Springer, 2024. 7, 8

[72] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[73] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 2

[74] Youyuan Zhang, Xuan Ju, and James J Clark. Fastvideoedit: Leveraging consistency models for efficient text-to-video editing. *arXiv preprint arXiv:2403.06269*, 2024. 2

[75] Lianghan Zhu, Yanqi Bao, Jing Huo, Jing Wu, Yu-Kun Lai, Wenbin Li, and Yang Gao. Zero-shot video editing through adaptive sliding score distillation. *arXiv preprint arXiv:2406.04888*, 2024.

[76] Zhichao Zuo, Zhao Zhang, Yan Luo, Yang Zhao, Haijun Zhang, Yi Yang, and Meng Wang. Cut-and-paste: Subject-driven video editing with attention control. *arXiv preprint arXiv:2311.11697*, 2023. 2