Videoshop: Localized Semantic Video Editing with Noise-Extrapolated Diffusion Inversion

Xiang Fan[1], Anand Bhattad[⋆2], and Ranjay Krishna[⋆1]

[1] University of Washington
[2] Toyota Technological Institute at Chicago
https://videoshop-editing.github.io/

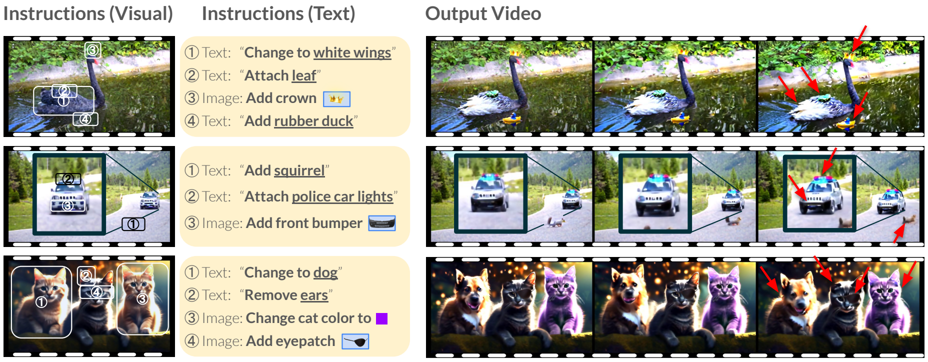


Fig. 1: Videoshop is a training-free method for precise video editing. Given an original video and user edits to the first frame, Videoshop automatically propagates the changes to all the frames of the video while maintaining semantic, geometric, and temporal consistency. To edit the first frame, users can leverage image editing tools, including text-based inpainting and professional editing software like Photoshop. As such, Videoshop supports video edits congruent with possible image edits that Photoshop enables: users can add new objects, remove objects or their parts, modify attributes, etc.

Abstract. We introduce Videoshop, a training-free video editing algorithm for localized semantic edits. Videoshop allows users to use any editing software, including Photoshop and generative inpainting, to modify the first frame; it automatically propagates those changes, with semantic, spatial, and temporally consistent motion, to the remaining frames. Unlike existing methods that enable edits only through imprecise textual instructions, Videoshop allows users to add or remove objects, semantically change objects, insert stock photos into videos, etc. with fine-grained control over locations and appearance. We achieve this through image-based video editing by inverting latents with noise extrapolation, from which we generate videos conditioned on the edited image. Videoshop produces higher quality edits against 6 baselines on 2 editing benchmarks using 10 evaluation metrics.

Keywords: Video Editing · Training-free · Diffusion Models

[⋆] equal advising

1 Introduction

Traditional video editing requires sophisticated direct manipulation and is a manually exhaustive process [34, 44]. Software tools like Adobe Premiere and Apple Final Cut have a steep learning curve and are still limited in their ease of use [23, 24]. While they support operations like stitching together video clips, they provide little to no support for *propagating edit changes* from one frame to another. Consider wanting to change the wings of a swan to have white features, add a crown on top of the swan's head, or even add a rubber duck wading along with the swan (Fig. 1). Such edits are only possible with painstaking per-frame manual curation.

Current video models fall short in facilitating precise, localized semantic editing. Open-source video diffusion models [4] have spurred new algorithmic developments [6, 28, 40, 58], yet they do not support the finesse required for localized edits. Models typically require extensive fine-tuning for individual videos or rely on coarse textual instructions that lack specificity [16, 47, 65]. Moreover, they struggle to introduce new objects with independent motions [26, 58].

Image editing has improved control over still frames [64], but ensuring temporal consistency across video frames remains unresolved. Frame-by-frame adjustments do not guarantee the continuity essential for coherent video sequences, revealing a gap between static image manipulation and dynamic video editing.

To address this limitation, we introduce Videoshop, a training-free video editing algorithm that enables users to make localized semantic edits. Videoshop allows users to make any modification they want to the first frame of the video; it propagates those changes, with temporally consistent motion, to all the remaining frames. The changes to the first frame can be made using any image editing tool [1, 3, 10, 17, 20, 37, 59, 63]. Users can use image editing models like ControlNet [64] or Text Inversion [15]; they can even load the frame onto Adobe Photoshop to edit pixels manually, add clip art, use content-aware fill, or any of Photoshop's features. Videoshop does not require finetuning and leverages Stable Video Diffusion [4] to make the edits. In other words, Videoshop can edit 14-frame videos within an average of 2 minutes, and supports edits within the large domain of videos. As video diffusion models themselves improve to support longer videos, Videoshop will be able to edit even longer videos.

Two technical insights enable Videoshop: (1) noticing that video latents follow a near-linear trajectory during the denoising process, and that (2) the VAE encoder is unnormalized, resulting in a high variance in the magnitude of the latents. To contextualize these insights, let's review how traditional image inversion enables image editing. Input images are inverted using a denoising diffusion implicit model (DDIM) to generate intermediate latents across time steps deterministically. This process approximates the noise at time step $t + 1$ using the latent from time step t. We find that this approximation reconstructs only the first frame accurately in image-to-video diffusion, resulting in unstable latents due to cumulative approximation errors. (1) Our investigations reveal that the latents are near-linear during the denoising process. We capitalize on this observation and introduce **inversion with noise extrapolation**, a mechanism for achieving faithful reconstruction for any video. (2) Our investigations also reveal that the video latents are unnormalized, leading to further instability. We introduce a **latent normalization** technique to ensure consistency and quality.

Our extensive experiments show that Videoshop produces higher quality edits against 6 baselines on 2 editing benchmarks using 10 evaluation metrics. This method empowers users to make direct pixel modifications, enabling a spectrum of semantic edits (refer to Fig. 1). Examples include "transforming a specific cat into a dog", "removing the ears of a cat", "changing a cat's fur color to purple", "adding dynamic objects such as a squirrel on the road" or "police car lights" that respond naturally to the car's movements. Videoshop not only ensures geometric fidelity where the car lights appropriately turn with the vehicle but also captures realistic motion, as exemplified by a rubber duck floating in unison with the swan and a squirrel that scurries away as a police car draws near.

On a broader scale, Videoshop equips users with video manipulation capabilities akin to those provided by image editing software like Photoshop, enabling new potential applications that previously would have been prohibitively challenging.

2 Related Work

Image and video generation. Generative Adversarial Networks (GANs) [18] initially advanced the quality of image generation, with notable architectures such as StyleGAN [31]. Despite the community's efforts, GANs remain difficult to train, limiting their ability to generate videos [27,46]. Diffusion models, including Denoising Score Matching [51] and Noise-Contrastive Estimation [19], are much easier to scale [21,30,43]. Today's video generation models are generally diffusion models [4,5,9,33,36,39,48,54,55,61]. We develop our technique using Stable Video Diffusion (SVD) [4], which is based on the EDM [30] framework and generates high-quality videos conditioned on a first frame.

Text-based video editing. Current video editing methods are commonly text-based, which modify video based on textual instructions. Such tasks include motion transfer [7,58,60], object editing [26,28,40,42,47,65], attribute editing [25, 28], texture editing [11,32], and style editing [6,16,26,40,57]. A common approach is to use a text-to-video model [54] and control the denoising process such that the generated result satisfies the editing conditions. However, textual instructions alone provide minimal specification, limiting most practical applications beyond modifying object classes, attributes, and textures.

DDIM inversion. Denoising Diffusion Implicit Models (DDIMs) [49] are a class of diffusion models that deterministically generate samples from a random noise. Due to its determinism, DDIM models can be inverted to find the corresponding noise given a sample. In the context of image editing, several methods have been proposed to improve the quality the edited images, including null-text inversion [38] and mathematically-exact inversion methods [35,52,62]. EDICT [52] and BDIM [62] propose an alternative denoising process that tracks more than one latent variables to derive an exact solution to the inversion formula. Another proposes a fixed-point iteration method to solve the inversion formula [35]. Unfortunately, applying diffusion inversion to text-to-image models can lead to a loss of temporal consistency in the resulting video [40].

Layered video editing. Another approach to video editing is through layered atlases, which decompose video frames into several layers (often foregrounds and backgrounds) and edit the layers corresponding to a target (*e.g.* NLA [32] and DiffusionAtlas [8]). However, atlases do not generate new motions or support edits beyond changes to object class, attribute, and texture [8].

3 Method

Given an input video $\mathcal{V}$ and an edited first frame $\mathcal{I}$, our goal is to generate a new video $\mathcal{J}$ that preserves the overall motion and semantics of the original video $\mathcal{V}$, while propagating the changes made to the first frame $\mathcal{I}$. Before we explain our method, we revisit diffusion models, DDIM inversion, and the EDM framework.

3.1 Background on latent diffusion models

Denoising Diffusion Probabilistic Models (DDPM) [21] are a class of models that generate an image or video from a random Gaussian noise through a sequence of T denoising steps. The denoising process is defined as a sequence of timesteps $T, T-1, ..., 0$. At timestep T, a $F \times H \times W \times 3$-dimensional random noise is sampled from a multivariate normal distribution, denoted as the initial noise x_T. DDPM applies a denoising neural network to iteratively de-noise latents $x_{t+1} \rightarrow x_t$ until x_0. The final x_0 is a generated $F \times H \times W \times 3$-dimensional video with F frames, and spatial size of $H \times W$.

Denoising in pixel space is computationally expensive as each denoising step needs to produce a $F \times H \times W \times 3$-dimensional noise estimation. **Latent Diffusion Models** [43] instead encode the sample image into a much smaller latent space using a pretrained variational autoencoder (VAE) with a $F \times h \times w \times c$-dimensional hidden representation, such that $F \times h \times w \times c \ll F \times H \times W \times 3$. Latent models sample x_T in the latent space and iteratively denoise until x_0. Finally, x_0 is decoded using the pretrained VAE decoder into the pixel space.

Latent models are trained by first sampling a video encoded as x_0, a timestep t, and a random noise ϵ. From these, a noise ϵ_t corresponding to the timestep t is calculated and added to x_0 to produce x_t. A denoising, timestep-conditioned U-Net ϵ_θ, is trained to estimate ϵ_t given x_t:

$$\mathbb{E}_{x,t,\epsilon \sim \mathcal{N}(0,1)} \|\epsilon_t - \epsilon_\theta(x_t; t)\|^2 \tag{1}$$

3.2 Background on diffusion inversion

Denoising Diffusion Implicit Models (DDIMs) [49] are a class of latent diffusion models with a deterministic denoising process. DDIMs deterministically generate samples from a random noise when its denoising step noise parameter is set to zero. This determinism allows for the diffusion inversion process.

The **inversion process** is a technique used to generate the deterministic latents x_T given a video $\mathcal{V}$. From $\mathcal{V}$, we first extract x_0 from the pretrained variational autoencoder's encoder. This process calculates the latents $\hat{x}_t$ for $t = 1 \dots T$, given x_0. If the inversion process is accurate, denoising $\epsilon_\theta(x_t; t)$ iteratively starting from $\hat{x}_T$ should yield $\hat{x}_0$, such that $\hat{x}_0 \approx x_0$.

In **diffusion-based image editing**, it is common to first invert the image latents x_0 into its corresponding noise $\hat{x}_T$, and apply the denoising process with a modified conditioning text to obtain the edited image [38]. Common image diffusion models are conditioned on a text prompt [43].

With deterministic sampling, the denoising step from $x_{t+1} \to x_t$ is defined as:

$$x_t = \sqrt{\alpha_t} \underbrace{\left(\frac{x_{t+1} - \sqrt{1-\alpha_{t+1}}\epsilon_\theta(x_{t+1}; t+1)}{\sqrt{\alpha_{t+1}}} \right)}_{\text{predicted } x_0} + \underbrace{\sqrt{1-\alpha_t} \cdot \epsilon_\theta(x_{t+1}; t+1)}_{\text{direction pointing to } x_{t+1}} \tag{2}$$

where ϵ_θ is the noise-predicting U-Net and α_t is the noise scheduling parameter.

Stable Video Diffusion [4] is a video diffusion model that conditions on the first-frame image instead of text. Ideally, one would invert the video latents similar to the inversion process mentioned above and apply edits by conditioning on an edited first frame. However, we observe that naively inverting the video latents often results in an incoherent video (Fig. 6), necessitating a new method.

3.3 Background on the EDM framework

Stable Video Diffusion [4] employs the **EDM framework** [30], which improves upon DDIM with a reparameterization to the denoising process. A denoising step from $x_{t+1} \to x_t$ of deterministic sampling in the EDM framework is:

$$x_t = x_{t+1} + \underbrace{\frac{\sigma_t - \sigma_{t+1}}{\sigma_{t+1}} \left(x_{t+1} - \overbrace{\left(c_{\text{skip}}^{t+1} x_{t+1} + c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} x_{t+1}; c_{\text{noise}}^{t+1} \right) \right)}^{\text{predicted } x_0} \right)}_{\text{noise removed at step } t} \tag{3}$$

where σ_t is the scheduled noise level at step t. c_{skip}^t, c_{in}^t, c_{out}^t, and c_{noise}^t are coefficients dependent on the noise schedule and the current step t. F_θ is a neural network parametrized by θ.

To perform **inversion in the EDM Framework,** we can rewrite Eq. (3) as an inversion step $\hat{x}_t \to \hat{x}_{t+1}$:

$$\hat{x}_{t+1} = \frac{\sigma_{t+1}\hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right)}{(\sigma_t - \sigma_{t+1}) \left(1 - c_{\text{skip}}^{t+1} \right) + \sigma_{t+1}} \tag{4}$$

However, because the input to F_θ is dependent on the next inverted latent $\hat{x}_{t+1}$, Eq. (4) is not directly solvable. Naive inversion methods approximate $\hat{x}_{t+1}$ with $\hat{x}_t$ such that:

$$F_\theta \left(c_{\text{in}}^{t+1} \hat{x}_{t+1}; c_{\text{noise}}^{t+1} \right) \approx F_\theta \left(c_{\text{in}}^{t} \hat{x}_t; c_{\text{noise}}^{t+1} \right) \tag{5}$$

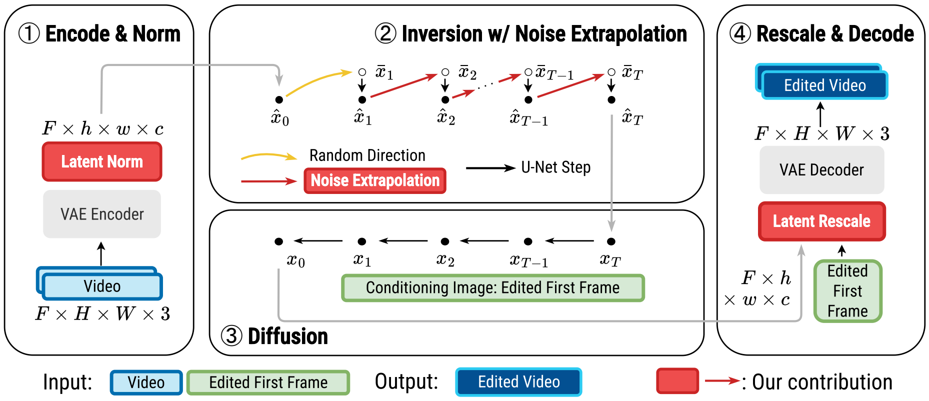


Fig. 2: Overview of VIDEOSHOP for localized semantic video editing. Our contributions are highlighted with red boxes and arrows. Our method includes four primary stages: (1) **Encode & Norm**, where the input video is encoded into a latent space using a VAE encoder, followed by normalization to ensure stability throughout inversion. (2) In the **Inversion w/ Noise Extrapolation** phase, noise extrapolation is systematically applied at each step to provide a corrective term that guides the inversion trajectory, ensuring the video is mapped to correct latent noise. This step is key for aligning the latent space trajectory at every timestep. (3) **Diffusion** then ensures user edits are seamlessly integrated across the video sequence, enforcing consistency while diffusing the initial modifications through time. (4) The last step is **Rescale & Decode**, where the now-edited latent sequence is rescaled to align with the original data's statistical distribution and decoded back into the video, resulting in an output video that reflects the desired semantic edits while maintaining the natural flow of the original sequence.

3.4 Limitations with naive video inversion to EDM

While common in inverting image diffusion models [6,38], naive inversion (Eq. (5)) leads to latents that only correctly reconstruct the first video frame, when directly applied to Stable Video Diffusion. We find that naive inversion (Eq. (5)) introduces a compounding approximation error that we show quantitatively and qualitatively in Sec. 4.1.

Several methods have been proposed to address this in image diffusion models [35,52,62]. Amongst them, EDICT [52] and FPI [35] require additional passes through the model at each step, increasing computation cost. While BDIA [62] improves upon EDICT [52] by eliminating additional passes, both of them modify the denoising process. We find that BDIA destabilizes the latents in the inversion process and results in undesirable artifacts in the resulting video (Fig. 5).

3.5 Our contribution: Inverting with noise extrapolation

Our goal is to find a better approximation for $F_\theta\left(c_{\text{in}}^{t+1}\hat{x}_{t+1}; c_{\text{noise}}^{t+1}\right)$ in Eq. (4). First, we observe that the latents in the denoising process maintain a near-linear trajectory. We then propose noise extrapolation to exploit this observation.

Near-linearity of x_t trajectory. It has been observed that the denoising trajectory of x_t in image diffusion models is approximately linear [30] at low and high noise levels. To investigate the latent trajectory of video diffusion models, we measure the average cosine similarity between vectors $x_t \to x_0$ and $x_{t'} \to x_0$ for pairs of t, t' in the denoising process of 100 random videos and show the results in Fig. 3. We observe high cosine similarities throughout the denoising process after the initial steps, suggesting that the trajectory of x_t is approximately linear after the low noise levels, which we exploit in our noise extrapolation method to invert $\hat{x}_t$.

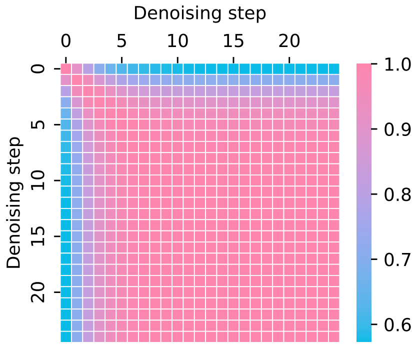


Fig. 3: Cosine similarity matrix for pairs of latent vectors throughout the denoising process. The latent vectors are approximately collinear, which supports our linear noise extrapolation.

Quantifying the near-linear trajectory. We measure the cosine similarities throughout the denoising process and report our results. The cosine similarity of $x_t - x_0$ and $x_{t'} - x_0$, averaged over all pairs of timesteps t, t' and 100 video samples, is **0.9282**. The cosine similarity of $x_t - x_0$ and $x_{t+1} - x_0$, averaged over timesteps t and 100 video samples, is **0.9919**. The minimum cosine similarity of $x_t - x_0$ and $x_{t+1} - x_0$, over all timesteps t and 100 video samples, is **0.9107**.

Noise extrapolation. With this insight, we linearly extrapolate the noise at $\hat{x}_t$ to obtain an approximation of $\bar{x}_{t+1}$ to provide to F_θ, as follows:

$$\bar{x}_{t+1} \approx \begin{cases} \underbrace{\frac{\sigma_{t+1}}{\sigma_t} \overbrace{(\hat{x}_t - x_0)}^{\sim\mathcal{N}(0,\sigma_t)}}_{\sim\mathcal{N}(0,\sigma_{t+1})} + x_0 & (\sigma_t > \varSigma) \\ \mathcal{N}(0, \sigma_{t+1}) + x_0 & (\sigma_t \leq \varSigma) \end{cases} \tag{6}$$

where $\varSigma$ is a threshold noise level. The threshold $\varSigma$ is necessary because at low σ_t, dividing by a small number results in a large noise, which can destabilize the latents. We ablate noise threshold in Sec. 4.1.

Putting it together, we modify Eq. (4) to estimate the inversion using $\bar{x}_{t+1}$:

$$\hat{x}_{t+1} = \frac{\sigma_{t+1}\hat{x}_t + (\sigma_t - \sigma_{t+1})\, c_{\text{out}}^{t+1} F_\theta \left(c_{\text{in}}^{t+1}\bar{x}_{t+1}; c_{\text{noise}}^{t+1}\right)}{(\sigma_t - \sigma_{t+1})\left(1 - c_{\text{skip}}^{t+1}\right) + \sigma_{t+1}} \tag{7}$$

After obtaining the final $\hat{x}_T$ from the inversion process, we apply the denoising process to $\hat{x}_T$ conditioned on the edited image $\mathcal{I}$ to obtain the edited latents x_0.

3.6 Our contribution: Latent normalization and rescaling

Direct output from the VAE encoder is unnormalized, resulting in a large variance in the magnitude of the final latent from the inversion process. We observe that this leads to poor quality in generated videos. To address this, we propose to normalize the latents before the start of the inversion process to unit standard deviation to stabilize the latents. After denoising, we rescale the latents with the mean and standard deviation of the latents of the target image:

$$\hat{x}_0 = \frac{x_{\text{in}}}{\sigma_{\text{in}}} \qquad x_{\text{out}} = \frac{\sigma_{\text{img}}}{\sigma_0}\left(x_0 - \mu_0 + \mu_{\text{img}}\right) \tag{8}$$

where x_{in} is the VAE-encoded latent sample, $\hat{x}_0$ is the input to the inversion process, x_0 is the output of the denoising process, and x_{out} is the final output to be decoded by the VAE into video $\mathcal{J}$. μ_{img} and σ_{img} are the mean and standard deviation of the VAE-encoded latents of the target image. All normalization are done per-channel. σ_{in} is calculated across all frames; μ_0, σ_0, μ_{img}, and σ_{img} are calculated for the first frame (the only available frame in the target image).

4 Experiments and Results

The key takeaways from our experiments are as follows: (1) Videoshop successfully performs localized semantic video editing among a diverse set of edit types. (2) Compared to Videoshop, existing methods demonstrate clear limitations in maintaining visual fidelity to the source video and target edit. (3) Videoshop achieves SOTA performance in localized editing, as evaluated by edit fidelity and source faithfulness, while maintaining high temporal consistency. (4) In our user study, Videoshop consistently outperforms text-based video editing methods, while maintaining high video generation quality. (5) Videoshop is efficient, with an average speedup of 2.23x compared to the baselines.

Datasets. We utilize two datasets for our experiments: a large-scale generated video editing dataset from the MagicBrush [63] image-editing dataset and an expert-curated video editing dataset with source videos from HD-VILA-100M [56].

The MagicBrush dataset is a manually annotated image editing dataset that contains over 10,000 tuples of "`(source image, instruction, edit mask, edited image)`". These tuples cover a wide range of edit types, including object addition, replacement, removal, and changes in action, color, texture, and counting. To convert MagicBrush into a video dataset, we generate videos conditioned on the source images using a video generation model [4]. The first frame of each generated video is conditioned to match the corresponding source image. The resulting dataset consists of "`(source video, instruction, edit mask, edited image)`" tuples.

The HD-VILA-100M dataset is a large-scale high-resolution video dataset collected from YouTube, encompassing diverse open domains. From this dataset, we sample 45 videos and ask editing experts to provide edits on the first frame of each video, along with the corresponding edit instructions. The resulting expert dataset consists of "`(source video, instruction, edit mask, edited image)`" tuples. A summary of the edit types from the expert dataset is shown in Tab. 1. All videos are resized to 14 frames with an aspect ratio of 16:9.

Table 1: Types of edit in the expert dataset.

Type of Edit	Percentage %
Add object	36%
Change appearance	20%
Remove object	18%
Replace object	16%
Change action	6%
Change color	4%

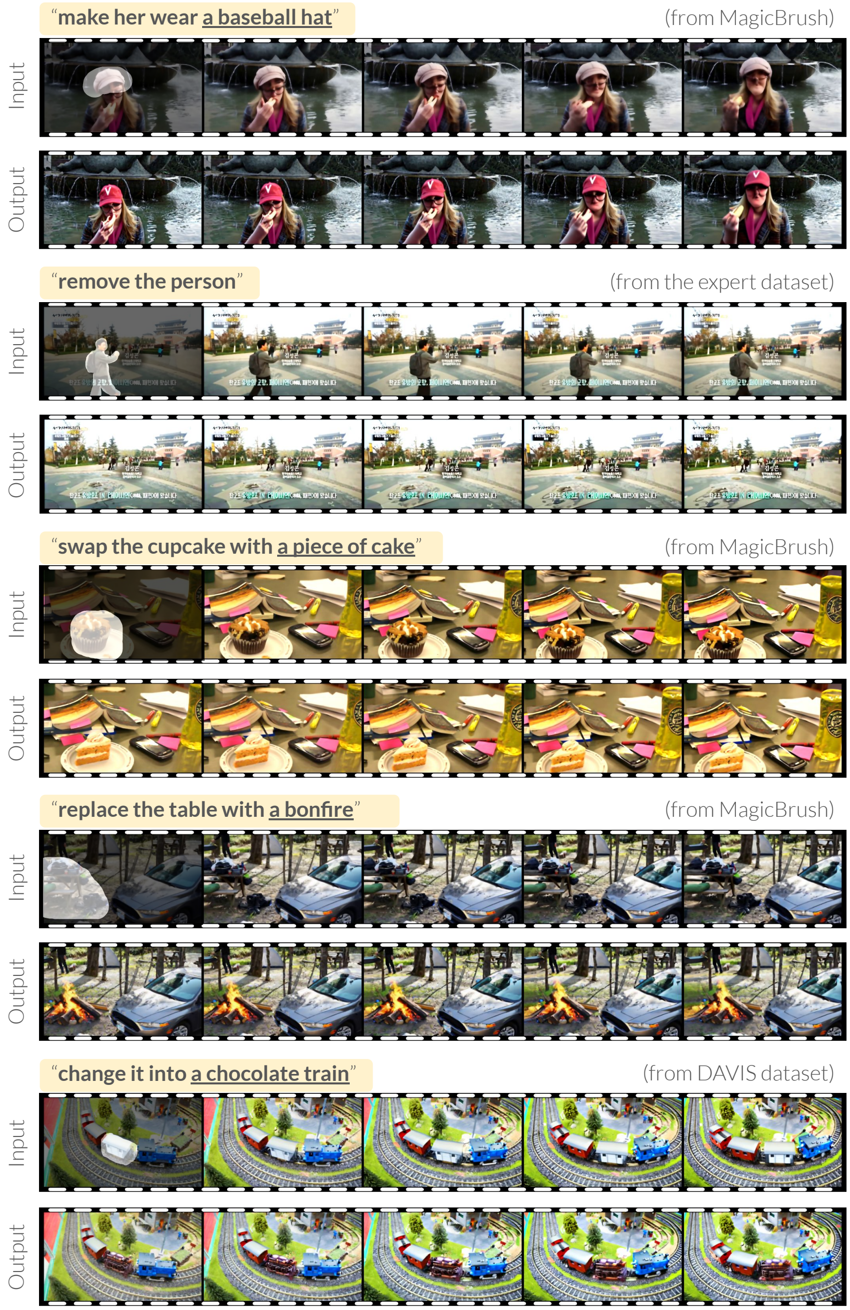

"make her wear a baseball hat"
(from MagicBrush)
Input
Output
"remove the person"
(from the expert dataset)
Input
Output
"swap the cupcake with a piece of cake"
(from MagicBrush)
Input
Output
"replace the table with a bonfire"
(from MagicBrush)
Input
Output
"change it into a chocolate train"
(from DAVIS dataset)
Input
Output

Fig. 4: Examples of edited videos. Our method handles a diverse set of edit types; examples shown include appearance editing, object removal, semantic editing, and shape/texture editing. VIDEOSHOP successfully performs precise local edits while maintaining high visual fidelity to the source video.

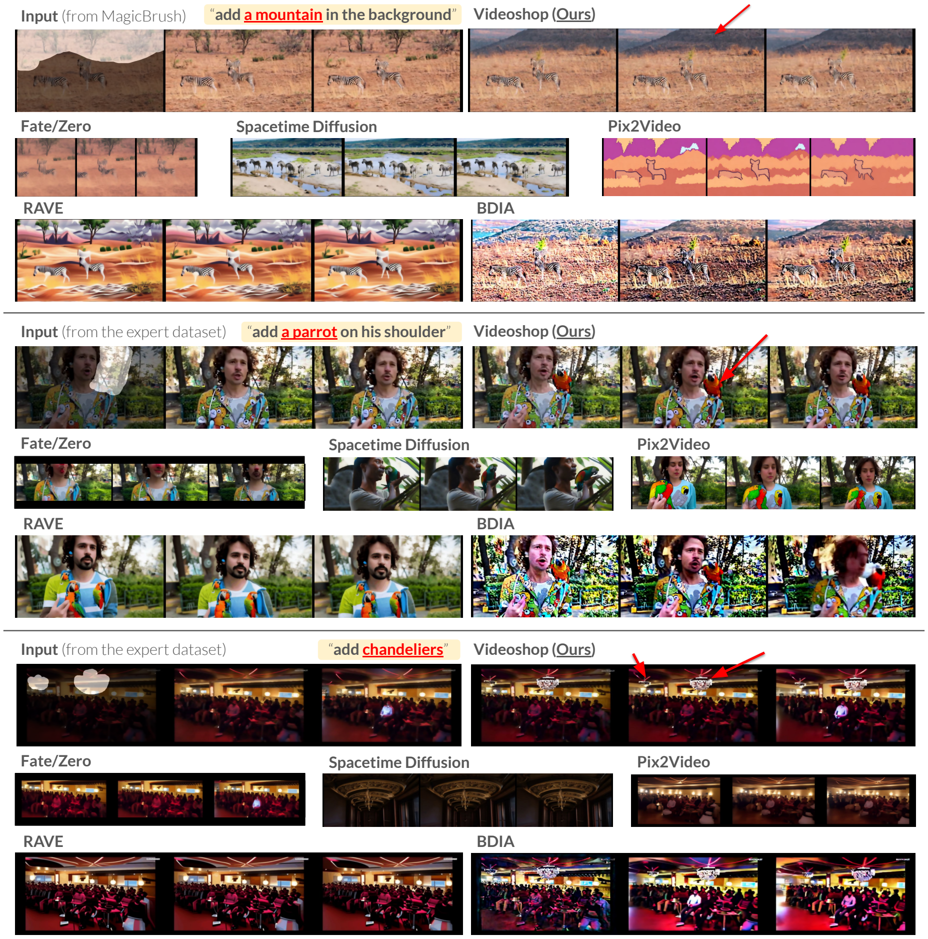


Fig. 5: Qualitative comparison against baselines. VIDEOSHOP successfully maintains visual fidelity to the source video and target edit, while existing methods fail to do so.

Experimental setup. We use the Stable Video Diffusion model [4] as the video generation model (F_θ in Eq. (7)). We compare our method against text-based video editing models including Pix2Video [6], Fate/Zero [40], Spacetime Diffusion [58], and RAVE [28], as well as the exact inversion method BDIA [62]. Pix2Video and RAVE are based on text-to-image models, while Fate/Zero and Spacetime Diffusion are based on text-to-video models. All baselines are implemented using their official codebases with videos resized to match the maximum supported video size. The baseline text-based video editing models either take a source and target prompt, or a single target prompt. To produce such source-target prompt pairs, we caption the first frame of the source video and update the modified concept to the target one (*e.g.*, "`a dog running`" → "`a cat running`").

Video editing capabilities of VIDEOSHOP. We demonstrate the video editing capabilities of VIDEOSHOP in Fig. 1 and Fig. 4. We show a diverse example of video edits, including object addition (Fig. 1(a)③④), object removal (Fig. 1(c)②, Fig. 4(b)), color edits (Fig. 1(c)③), semantic edits (Fig. 1(c)①, Fig. 4(c)(d)(e)), object attachment (Fig. 1(b)②) and appearance edits (Fig. 1(a)①, Fig. 4(a)). We also show examples of multiple edits at once (Fig. 1). We observe that Videoshop successfully performs precise local edits, appearance control, and independent object addition, while maintaining high visual fidelity to the source video.

Limitations of existing methods. We compare Videoshop with existing methods in Fig. 5. We observe that existing methods often fail to maintain visual fidelity to the source video and target edit. For example, Fate/Zero does not correctly modify the source video based on the edit, Spacetime Diffusion generates videos that are structurally inconsistent with the source video, Pix2Video and RAVE can both undesirably change the style of the source video (Fig. 5(a)), and BDIA demonstrates large visual inconsistencies from the source video.

Automated Evaluation. To assess generation quality, we adopt a comprehensive set of metrics. We use the terms "source video frames," "target image," and "edited video frames" to refer to the frames from the source video, the reference edited first frame, and the model-generated edited video frames.

1. *Edit Fidelity*: Building on prior text-based video editing work [6,28,58], we use the CLIP similarity [41] metric. Our $\mathbf{CLIP_{tgt}}$ metric measures the similarity of CLIP embeddings between each edited frame and the target image. The $\mathbf{CLIP^{+}_{tgt}}$ score, utilizing CoTracker [29], focuses only on the edited region. Furthermore, the **TIFA** score [22] evaluates semantic alignment between the target image and the edited video frames.
2. *Source Faithfulness*: We measure the $\mathbf{CLIP_{src}}$ similarity between the source and edited videos. To refine this, $\mathbf{CLIP^{+}_{src}}$ masks the edited region using CoTracker [29] and measures only the unedited region. Motion faithfulness is assessed via the end-point error (EPE) from optical flow comparisons using RAFT [50], which we denote as **Flow**. We also report the EPE within only unedited regions, denoted as $\mathbf{Flow^{+}}$. **FVD** and **SSIM** scores provide additional quality measures.
3. *Temporal Consistency*: The average CLIP similarity between consecutive frames ($\mathbf{CLIP_{TC}}$) is calculated following protocols from [6, 16, 28, 57].

We report these metrics for both Videoshop and baseline methods, detailed in Tab. 2 for MagicBrush and Tab. 3 for the expert dataset. In Tab. 2, Videoshop demonstrates superior performance over the baseline methods across the majority of metrics related to edit fidelity and source faithfulness, while showing competitive results in other evaluated areas, with marginal differences from the leading method. The findings in Tab. 3 echo this pattern, as Videoshop consistently ranks above the baselines in most metrics and holds competitive in the rest. Notably, the expert dataset exhibits an overall increase in optical flow errors, likely due to the more complex motion dynamics in real-world videos. Despite this, Videoshop maintains competitive scores on flow metrics and stands out particularly in preserving the source video flow in regions unoccupied by the edit.

Table 2: Quantitative results on MagicBrush. (**T.C.** = Temporal Consistency.)

Method	Edit Fidelity			Source Faithfulness						T.C.
	CLIP$_{\text{tgt}}$↑ ($\times 10^{-2}$)	**CLIP$^{+}_{\text{tgt}}$↑** ($\times 10^{-2}$)	**TIFA↑** ($\times 10^{-2}$)	**CLIP$_{\text{src}}$↑** ($\times 10^{-2}$)	**CLIP$^{+}_{\text{src}}$↑** ($\times 10^{-2}$)	**Flow↓** ($\times 1$)	**Flow^{+}↓** ($\times 1$)	**FVD↓** ($\times 1$)	**SSIM↑** ($\times 10^{-2}$)	**CLIP$_{\text{TC}}$↑** ($\times 10^{-2}$)
BDIA [62]	82.12	82.19	57.67	82.48	87.10	2.83	1.43	3482.79	49.67	94.36
Pix2Video [6]	71.19	76.47	51.98	74.55	79.03	3.59	2.58	2993.95	59.08	94.48
Fate/Zero [40]	84.87	79.10	55.41	**92.41**	86.94	4.42	3.11	2205.03	48.59	95.71
Spacetime [58]	63.85	75.20	46.33	65.74	71.91	8.24	5.62	4815.63	41.61	96.58
RAVE [28]	74.70	78.58	51.12	75.99	80.19	3.35	2.42	2354.09	62.21	**96.59**
SVD (no src. vid.)	87.63	84.73	64.16	90.64	94.47	9.74	6.89	1894.16	47.50	95.07
Videoshop (Ours)	**88.80**	**85.58**	**64.40**	$\underline{90.95}$	**94.77**	**1.90**	**0.78**	**1478.76**	**71.92**	95.16

Table 3: Quantitative results on the expert dataset. (**T.C.** = Temporal Consistency.)

Method	Edit Fidelity			Source Faithfulness						T.C.
	CLIP$_{\text{tgt}}$↑ ($\times 10^{-2}$)	**CLIP$^{+}_{\text{tgt}}$↑** ($\times 10^{-2}$)	**TIFA↑** ($\times 10^{-2}$)	**CLIP$_{\text{src}}$↑** ($\times 10^{-2}$)	**CLIP$^{+}_{\text{src}}$↑** ($\times 10^{-2}$)	**Flow↓** ($\times 1$)	**Flow^{+}↓** ($\times 1$)	**FVD↓** ($\times 1$)	**SSIM↑** ($\times 10^{-2}$)	**CLIP$_{\text{TC}}$↑** ($\times 10^{-2}$)
BDIA [62]	81.90	79.97	63.63	80.38	82.94	12.18	10.10	3048.53	36.42	92.86
Pix2Video [6]	68.28	71.18	49.66	72.35	76.81	7.92	6.46	1876.43	61.23	93.90
Fate/Zero [40]	71.51	72.53	48.65	74.59	82.68	9.70	8.30	2103.63	52.75	95.77
Spacetime [58]	58.61	68.45	43.06	56.51	65.86	7.93	6.31	4497.18	40.12	**97.50**
RAVE [28]	71.59	71.39	57.78	74.37	76.35	**5.59**	4.54	1890.95	63.76	96.05
Videoshop (Ours)	**87.96**	**83.54**	**66.14**	**84.89**	**91.50**	$\underline{5.85}$	**4.47**	**1718.31**	**65.63**	94.71

Human Evaluation. We conduct a human evaluation study on the dev set of the MagicBrush dataset. For each baseline, we ask evaluators to compare the editing and video generation quality of our method with the baseline. The results are in the first two columns of Tab. 4. We observe that our method outperforms all baselines in both editing quality and video generation quality.

Efficiency. We assess the efficiency of Videoshop against baseline methods by measuring the average execution time per video, last column in Tab. 4. Videoshop aligns closely with the execution time of BDIA, which is known for its low overhead due to not requiring extra U-Net steps. Additionally, Videoshop provides a considerable speed advantage, operating at more than twice the speed (2.23x faster) of the average baseline method.

4.1 Ablation Study

We ablate latent normalization (Sec. 3.6), latent rescaling (Sec. 3.6), noise extrapolation (Sec. 3.5), and noise threshold (Sec. 3.5) qualitatively in Fig. 6 and quantitatively in Tab. 5. In Fig. 6, the yellow vertical line tracks the background movement and the yellow color picker shows the RGB value of a background pixel at the same location in the first frame. As we can see, the lack of noise extrapolation results in incoherent videos, the lack of latent normalization results in a slow, incorrect background movement that disregards motion in the source video, and removing latent rescaling results in shifted colors. In Tab. 5, we observe that our final method outperforms all ablations except for a few scores without latent normalization. However, it is important to note that the lack of latent normalization tends to result in static or slow movements that disregard motion

Table 4: Human evaluation and execution time. For human evaluation, evaluators are asked to compare the editing quality and video generation quality of our method against each baseline. VIDEOSHOP outperforms all baselines in both editing quality and video generation quality and has a competitive runtime compared to the baseline methods.

VIDEOSHOP (Ours) vs. ...	Editing Quality (preference in our favor %)	Video Generation Quality (preference in our favor %)	Execution Time (as multiples of ours)
BDIA [62]	94.89%	90.53%	1.03x
Pix2Video [6]	98.30%	96.97%	4.70x
Fate/Zero [40]	98.67%	92.23%	0.71x
Spacetime [58]	99.81%	69.32%	3.41x
RAVE [28]	92.99%	74.05%	1.31x
Average	96.93%	84.62%	2.23x

Table 5: Ablations on a 10-video subset. Our method outperforms all ablations except for a few scores when removing latent norm (second row). However, it is important to note that the lack of latent norm tends to result in static or slow movements that disregard the movement in the source video, as evidenced by the low Flow scores. Furthermore, removing the noise threshold (last row) leads to division by a very small number, resulting in NaN values in the latents. (**T.C.** = Temporal Consistency.)

Method	Edit Fidelity			Source Faithfulness						T.C.
	CLIP$_{\text{tgt}}$↑ ($\times 10^{-2}$)	CLIP$^{+}_{\text{tgt}}$↑ ($\times 10^{-2}$)	TIFA↑ ($\times 10^{-2}$)	CLIP$_{\text{src}}$↑ ($\times 10^{-2}$)	CLIP$^{+}_{\text{src}}$↑ ($\times 10^{-2}$)	Flow↓ ($\times 1$)	Flow^{+}↓ ($\times 1$)	FVD↓ ($\times 1$)	SSIM↑ ($\times 10^{-2}$)	CLIP$_{\text{TC}}$↑ ($\times 10^{-2}$)
VIDEOSHOP (Ours)	**92.12**	88.89	**100**	88.82	**97.92**	**4.57**	**2.42**	1410.70	**57.25**	95.18
– Norm.	91.36	**91.54**	**100**	**90.22**	**97.34**	**19.57**	**13.00**	**1324.45**	41.74	**96.58**
– Rescaling	89.74	88.45	90	88.72	97.34	6.37	3.64	1784.10	52.52	94.60
– Extrapolation	73.36	75.42	70	87.19	95.12	17.92	10.47	5316.65	16.76	84.55
– Threshold	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

in the source video, and therefore are unsuitable for video editing. Furthermore, removing the threshold in noise extrapolation leads to division by a very small number, resulting in numerical instability and NaN values in the latents.

4.2 Extension to Other Models and Longer Videos

Our approach can be applied to any image-to-video diffusion model with an invertible denoising step. We show results on the AnimateLCM [53] model, a recent image-to-video model, in Fig. 7a. The number of frames in our method can be naturally extended by recurrently generating blocks of frames, each block conditioned on the last frame of the previous block as shown in our Fig. 7b.

5 Discussion and Limitations

VIDEOSHOP is a novel video editing approach that lets users make localized semantic edits without any need for re-training. Other video editing methods currently edit whole videos with sparse textual instructions. Instead, we simplify the problem by reducing it to image editing, a well-studied and widely applied task in the image domain. We exploited the linear relationship in the latents during the inversion process with our noise extrapolation and showed our latent normalization and rescaling enables realistic localized semantic editing.

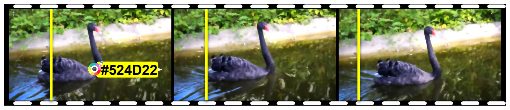
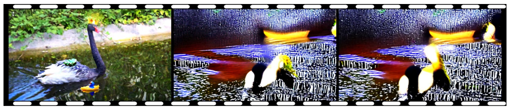
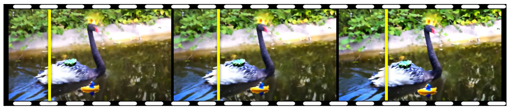
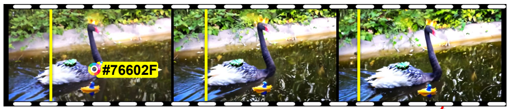
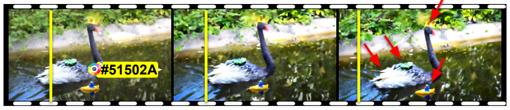
Method	Output	Comments
Input Video		
w/o Extrapolation (Naive Inversion)		Incoherent video.
w/o Latent Norm		Incorrect background movement (see yellow guide line).
w/o Latent Rescaling		Large color shift (see color picker value).
Full Method		

Fig. 6: Examples from ablations. The yellow vertical line tracks the background movement and the yellow color picker shows the RGB value of a background pixel at the same location in the first frame. As we can see from the examples, the lack of noise extrapolation results in incoherent videos, the lack of latent normalization results in a slow, incorrect background movement that disregards the source video's movement, and the lack of latent rescaling results in shifted colors.

(a) Results on AnimateLCM [53]. **(b)** Expanded frames (50 total).

Fig. 7: Our method can extend to other video diffusion models and longer videos.

Despite the strengths of our method, there are limitations: (a) The VAE used in our method may lead to loss of information during video encoding, which could obscure some fine details such as small text. (b) In cases where the source video contains large movements or flickering, the temporal consistency of our method may be compromised. (c) Our method is training-free, which means that it only uses the knowledge contained in the base model. This limits the introduction of new information (such as new motion). We anticipate that our approach will become even more effective as image-to-video models improve. Finally, we can leverage video models to extend our method for editing 3D meshes [12, 13] and extracting visual knowledge by analyzing the latent [2, 14, 45]. Another line of work includes combining image editing with motion and trajectory controls to ensure seamless video results. In summary, Videoshop reimagines video editing and could unlock possibilities for new and creative applications. With our method, users can edit videos with the same ease as they would edit images in Photoshop.

Acknowledgement. This work was supported by NSF award IIS-2211133.

References

1. Bhattad, A., Forsyth, D.A.: Cut-and-paste object insertion by enabling deep image prior for reshading. In: 2022 International Conference on 3D Vision (3DV). pp. 332–341. IEEE (2022)
2. Bhattad, A., McKee, D., Hoiem, D., Forsyth, D.: Stylegan knows normal, depth, albedo, and more. Advances in Neural Information Processing Systems **36** (2024)
3. Bhattad, A., Soole, J., Forsyth, D.: Stylitgan: Image-based relighting via latent control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4231–4240 (2024)
4. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., Rombach, R.: Stable video diffusion: Scaling latent video diffusion models to large datasets (2023)
5. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22563–22575 (2023)
6. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2video: Video editing using image diffusion (2023)
7. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now (2019)
8. Chang, S.Y., Chen, H.T., Liu, T.L.: Diffusionatlas: High-fidelity consistent diffusion video editing (2023)
9. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models (2024)
10. Chen, X., Huang, L., Liu, Y., Shen, Y., Zhao, D., Zhao, H.: Anydoor: Zero-shot object-level image customization. arXiv preprint arXiv:2307.09481 (2023)
11. Couairon, P., Rambour, C., Haugeard, J.E., Thome, N.: Videdit: Zero-shot and spatially aware text-driven video editing (2023)
12. Decatur, D., Lang, I., Aberman, K., Hanocka, R.: 3d paintbrush: Local stylization of 3d shapes with cascaded score distillation. arXiv preprint arXiv:2311.09571 (2023)
13. Decatur, D., Lang, I., Hanocka, R.: 3d highlighter: Localizing regions on 3d shapes via text descriptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20930–20939 (2023)
14. Du, X., Kolkin, N., Shakhnarovich, G., Bhattad, A.: Generative models: What do they know? do they know things? let's find out! arXiv preprint arXiv:2311.17137 (2023)
15. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)
16. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: Tokenflow: Consistent diffusion features for consistent video editing (2023)
17. Goel, V., Peruzzo, E., Jiang, Y., Xu, D., Sebe, N., Darrell, T., Wang, Z., Shi, H.: Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models. arXiv preprint arXiv:2303.17546 (2023)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
19. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 297–304. JMLR Workshop and Conference Proceedings (2010)

20. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020)
22. Hu, Y., Liu, B., Kasai, J., Wang, Y., Ostendorf, M., Krishna, R., Smith, N.A.: Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering (2023)
23. Inc., A.: Final cut pro. `https://www.apple.com/final-cut-pro/` (2023), accessed: 2024-03-03
24. Incorporated, A.S.: Adobe premiere pro. `https://www.adobe.com/products/premiere.html` (2023), accessed: 2024-03-03
25. Jeong, H., Ye, J.C.: Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models (2024)
26. Kahatapitiya, K., Karjauv, A., Abati, D., Porikli, F., Asano, Y.M., Habibian, A.: Object-centric diffusion for efficient video editing (2024)
27. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10124–10134 (2023)
28. Kara, O., Kurtkaya, B., Yesiltepe, H., Rehg, J.M., Yanardag, P.: Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models (2023)
29. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together (2023)
30. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models (2022)
31. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
32. Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing (2021)
33. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., et al.: Videopoet: A large language model for zero-shot video generation. arXiv preprint arXiv:2312.14125 (2023)
34. Mackay, W., Pagani, D.: Video mosaic: Laying out time in a physical space. In: Proceedings of the second ACM international conference on Multimedia. pp. 165–172 (1994)
35. Meiri, B., Samuel, D., Darshan, N., Chechik, G., Avidan, S., Ben-Ari, R.: Fixed-point inversion for text-to-image diffusion models (2023)
36. Menapace, W., Siarohin, A., Skorokhodov, I., Deyneka, E., Chen, T.S., Kag, A., Fang, Y., Stoliar, A., Ricci, E., Ren, J., et al.: Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. arXiv preprint arXiv:2402.14797 (2024)
37. Michel, O., Bhattad, A., VanderBilt, E., Krishna, R., Kembhavi, A., Gupta, T.: Object 3dit: Language-guided 3d-aware image editing. Advances in Neural Information Processing Systems **36** (2024)
38. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models (2022)
39. Mullan, J., Crawbuck, D., Sastry, A.: Hotshot-XL (Oct 2023), `https://github.com/hotshotco/hotshot-xl`
40. Qi, C., Cun, X., Zhang, Y., Lei, C., Wang, X., Shan, Y., Chen, Q.: Fatezero: Fusing attentions for zero-shot text-based video editing (2023)

41. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)
42. Ren, Y., Zhou, Y., Yang, J., Shi, J., Liu, D., Liu, F., Kwon, M., Shrivastava, A.: Customize-a-video: One-shot motion customization of text-to-video diffusion models (2024)
43. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022)
44. Santosa, S., Chevalier, F., Balakrishnan, R., Singh, K.: Direct space-time trajectory control for visual media editing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1149–1158 (2013)
45. Sarkar, A., Mai, H., Mahapatra, A., Lazebnik, S., Forsyth, D.A., Bhattad, A.: Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28140–28149 (2024)
46. Sauer, A., Karras, T., Laine, S., Geiger, A., Aila, T.: Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. arXiv preprint arXiv:2301.09515 (2023)
47. Shin, C., Kim, H., Lee, C.H., gil Lee, S., Yoon, S.: Edit-a-video: Single video editing with object-aware consistency (2023)
48. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
49. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2022)
50. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow (2020)
51. Vincent, P.: A connection between score matching and denoising autoencoders. Neural computation **23**(7), 1661–1674 (2011)
52. Wallace, B., Gokul, A., Naik, N.: Edict: Exact diffusion inversion via coupled transformations (2022)
53. Wang, F.Y., Huang, Z., Shi, X., Bian, W., Song, G., Liu, Y., Li, H.: Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning (2024)
54. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: Modelscope text-to-video technical report (2023)
55. Wang, X., Yuan, H., Zhang, S., Chen, D., Wang, J., Zhang, Y., Shen, Y., Zhao, D., Zhou, J.: Videocomposer: Compositional video synthesis with motion controllability (2023)
56. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., Fu, J., Guo, B.: Advancing high-resolution video-language representation with large-scale video transcriptions (2022)
57. Yang, S., Zhou, Y., Liu, Z., Loy, C.C.: Rerender a video: Zero-shot text-guided video-to-video translation (2023)
58. Yatim, D., Fridman, R., Bar-Tal, O., Kasten, Y., Dekel, T.: Space-time diffusion features for zero-shot text-driven motion transfer (2023)
59. Yenphraphai, J., Pan, X., Liu, S., Panozzo, D., Xie, S.: Image sculpting: Precise object editing with 3d geometry control. arXiv preprint arXiv:2401.01702 (2024)
60. Yin, W., Yin, H., Baraka, K., Kragic, D., Björkman, M.: Dance style transfer with cross-modal transformer (2023)
61. Zhang, D.J., Wu, J.Z., Liu, J.W., Zhao, R., Ran, L., Gu, Y., Gao, D., Shou, M.Z.: Show-1: Marrying pixel and latent diffusion models for text-to-video generation (2023)

62. Zhang, G., Lewis, J.P., Kleijn, W.B.: Exact diffusion inversion via bi-directional integration approximation (2023)
63. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing (2023)
64. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
65. Zuo, Z., Zhang, Z., Luo, Y., Zhao, Y., Zhang, H., Yang, Y., Wang, M.: Cut-and-paste: Subject-driven video editing with attention control (2023)