TokenFlow: Consistent Diffusion Features for Consistent Video Editing

Michal Geyer* Omer Bar-Tal* Shai Bagon Tali Dekel
Weizmann Institute of Science
*Indicates equal contribution.

Project webpage: https://diffusion-tokenflow.github.io

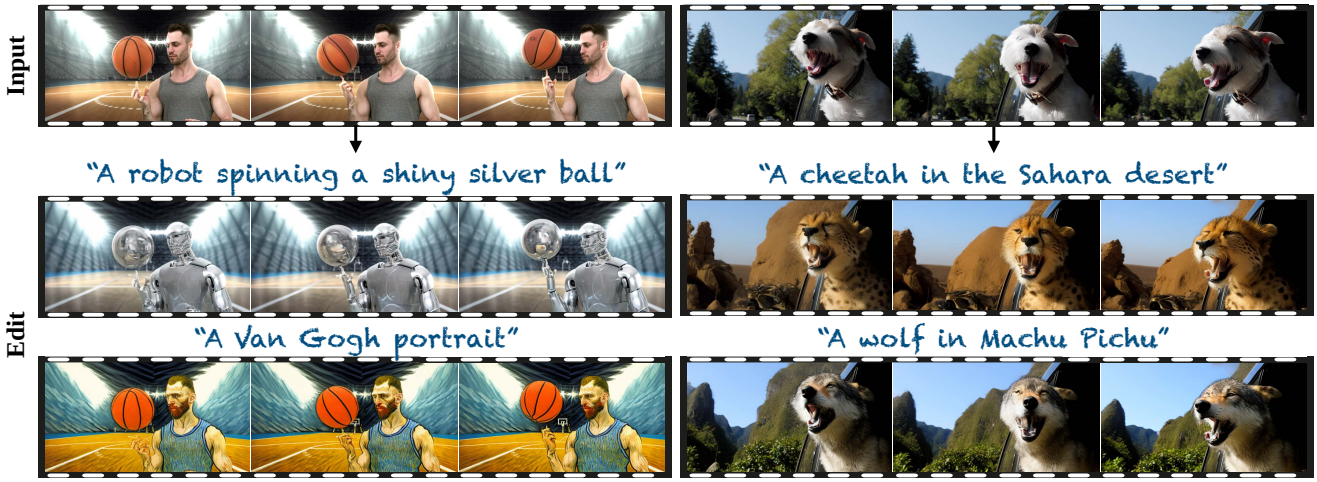

Figure 1. TokenFlow enables consistent, high-quality semantic edits of real-world videos. Given an input video (top row), our method edits it according to a target text prompt (middle and bottom rows), while preserving the semantic layout and motion in the original scene.

Abstract

The generative AI revolution has recently expanded to videos. Nevertheless, current state-of-the-art video models are still lagging behind image models in terms of visual quality and user control over the generated content. In this work, we present a framework that harnesses the power of a text-to-image diffusion model for the task of text-driven video editing. Specifically, given a source video and a target text-prompt, our method generates a high-quality video that adheres to the target text, while preserving the spatial layout and motion of the input video. Our method is based on a key observation that consistency in the edited video can be obtained by enforcing consistency in the diffusion feature space. We achieve this by explicitly propagating diffusion features based on inter-frame correspondences, readily available in the model. Thus, our framework does not require any training or fine-tuning, and can work in conjunction with any off-the-shelf text-to-image editing method. We demonstrate state-of-the-art editing results on a variety of real-world videos.

1. Introduction

The evolution of text-to-image models has recently facilitated advances in image editing and content creation, allowing users to control various proprieties of both generated and real images. Nevertheless, expanding this exciting progress to video is still lagging behind. A surge of large-scale text-to-video generative models has emerged, demonstrating impressive results in generating clips solely from textual descriptions. However, despite the progress made in this area, existing video models are still in their infancy, being limited in resolution, video length, or the complexity of video dynamics they can represent. In this paper, we harness the power of a state-of-the-art pre-trained text-to-*image* model for the task of text-driven editing of natural videos. Specifically, our goal is to generate high-quality videos that adhere to the target edit expressed by an input text prompt, while preserving the spatial layout and motion of the original video.

The main challenge in leveraging an image diffusion model for video editing is to ensure that the edited content is consistent across all video frames – ideally, each physical point in the 3D world undergoes coherent modifications across time. Existing and concurrent video editing methods that are based on image diffusion models have demonstrated that global appearance coherency across the edited frames can be achieved by extending the self-attention module to include multiple frames (e.g., [53, 19, 5, 34]). Nevertheless, this approach is insufficient for achieving the desired level of temporal consistency, as motion in the video is only

implicitly preserved through the attention module. Consequently, professionals or semi-professionals users often resort to elaborate video editing pipelines that entail additional manual work.

In this work, we propose a framework to tackle this challenge by *explicitly* enforcing the original inter-frame video correspondences on the edit.

Intuitively, natural videos contain redundant information across frames, e.g., depict similar appearance and shared visual elements. Our key observation is that the internal representation of the video in the diffusion model exhibits similar properties. That is, the level of redundancy and temporal consistency of the frames in the RGB space and in the diffusion feature space are tightly correlated. Based on this observation, the pillar of our approach is to achieve consistent edit by ensuring that the features of the edited video are consistent across frames. Specifically, we enforce that the edited features convey the same inter-frame correspondences and redundancy as the original video features. To do so, we leverage the original inter-frame feature correspondences, which are readily available by the model. This leads to an effective method that directly propagates the *edited* diffusion features based on the *original* video dynamics. This approach allows us to harness the generative prior of state-of-the-art image diffusion model without additional training or fine-tuning, and can work in conjunction with an off-the-shelf diffusion-based image editing method (e.g., [29, 56, 12]).

To summarize, we make the following key contributions:

- A technique, dubbed *TokenFlow*, that enforces semantic correspondences of diffusion features across frames, allowing to significantly increase temporal consistency in videos generated by a text-to-image diffusion model.
- Novel empirical analysis studying the proprieties of diffusion features across a video.
- State-of-the-art editing results on diverse videos, depicting complex motions.

2. Related Work

Text-driven image & video synthesis Seminal works designed GAN architectures to synthesize images conditioned on text embeddings [37, 54]. With the ever-growing scale of vision-language datasets and pretraining strategies [35, 42], there has been a remarkable progress in text-driven image generation capabilities. Users can sytnesize high-quality visual content using simple text prompts. Much of this progress is also attributed to diffusion models [47, 8, 9, 14, 31] which have been established as state-of-the-art text-to-image generators [30, 41, 36, 38, 44, 2]. Such models have been extended for text-to-video generation, by extending 2D architectures to the temporal dimension (e.g., using temporal attention [15]) and performing large-scale training on video datasets [13, 46]. Recently, Gen-1 [10] tailored a diffusion model architecture for the task of video editing, by conditioning the network on structure/appearance representations. Nevertheless, due to their extensive computation and memory requirements, existing video diffusion models are still in infancy and are largely restricted to short clips, or exhibit lower visual quality compared to image models. On the other side of the spectrum, a promising recent trend of works leverage a pre-trained image diffusion model for video synthesis tasks, without additional training [11, 53, 23, 34]. Our work falls into this category, employing a pretrained text-to-image diffusion model for the task of video editing, without any training or fine-tuning.

Consistent video stylization A common approach for video stylization involves applying image editing techniques (e.g., style transfer) on a frame-by-frame basis, followed by post-processing to address temporal inconsistencies in the resulting video [21, 25, 24]. Although these methods effectively reduce high-frequency temporal flickering, they are not designed to handle frames that exhibit substantial variations in content, which often occur when applying text-based image editing techniques [34]. Kasten et al. [18] propose to decompose a video into a set of 2D atlases, each provides a unified representation of the background or of a foreground object throughout the video. Edits applied to the 2D atlases are automatically mapped back to the video, thus achieving temporal consistency with minimal effort. However, this method is limited in representation capabilities and requires long training, both limiting the applicability of this technique. Our work is also related to classical works that demonstrated that small patches in a natural video extensively repeat across frames [43, 7], and thus consistent editing can by simplified by editing a subset of keyframes and propagating the edit across the video by establishing patch correspondences using handcrafted features and optical flow [40] or by training a patch-based GAN [50].

Nevertheless, such propagation methods struggle to handle videos with illumination changes, or with complex dynamics, and can only function as post-processing. Our work shares a similar motivation as this approach that benefits from the temporal redundancies in natural videos. We show that such redundancies are also present in the feature space of a text-to-image diffusion model, and leverage this property to achieve consistency.

Controlled generation via diffusion features manipulation Recently, a surge of works demonstrated how text-to-image diffusion models can be readily adapted to various editing and generation tasks, by performing simple operations on the intermediate feature representation of the diffusion network [6, 16, 28, 51, 12, 32, 4]. Concurrent works demonstrated semantic appearance swapping using diffusion feature correspondences ([27, 55]). Prompt-to-Prompt [12] observed that by manipulating the cross-attention layers, it is possible to control the relation between the spatial layout of the image to each word in the text. Plug-and-

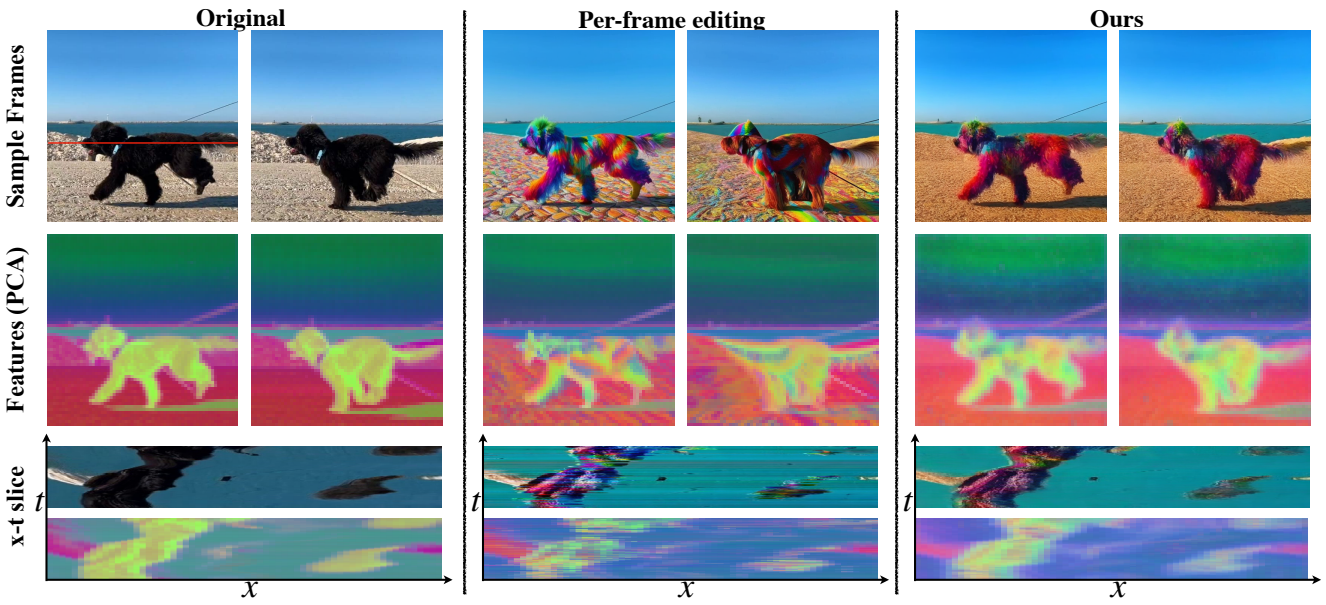

Figure 2. **Diffusion features across time.** *Left:* Given an input video (top row), we apply DDIM inversion on each frame and extract features from the highest resolution decoder layer in ϵ_θ. We apply PCA on the features (i.e., output tokens from the self-attention module) extracted from all frames and visualize the first three components (second row). We further visualize an x-t slice (marked in red on the original frame) for both RGB and features (bottom row). The feature representation is consistent across time – corresponding regions are encoded with similar features across the video. *Middle:* Frames and feature visualization for an edited video obtained by applying an image editing method ([51]) on each frame; inconsistent patterns in RGB are also evident in the feature space (e.g., on the dog's body). *Right:* Our method enforces the edited video to convey the same level of feature consistency as the original video, which translates into a coherent and high-quality edit in RGB space.

Play Diffusion (PnP) [51] analyzed the spatial features and the self-attention maps and found that they capture semantic information at high spatial granularity. Tune-A-Video [53] observed that by extending the self-attention module to operate on more then a single frame, it is possible to generate frames that share a common global appearance. Concurrent works [34, 5, 20, 45, 26] leverage this property to achieve globally-coherent video edits. Nevertheless, as demonstrated in Sec. 5, inflating the self-attention module is insufficient for achieving fine-grained temporal consistency. Prior and concurrent works either compromise visual quality, or exhibit limited temporal consistency. In this work, we also perform video editing via simple operations in the feature space of a pre-trained text-to-image model. However, rather than solely relying on self-attention inflation, we explicitly encourage the features of the model to be temporally consistent through *TokenFlow*.

3. Preliminaries

Diffusion Models Diffusion probabalistic models (DPM) [47, 8, 9, 14, 31] are a class of generative models that aim to approximate a data distribution q through a progressive denosing process. Starting from a Gaussian i.i.d noisy image $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$, the diffusion model gradually denoises it, until reaching a clean image $\boldsymbol{x}_0$ drawn from the target distribution q. DPM can learn a conditional distribution by incorporating additional guiding signals, such as text conditioning.

Song et al. [48] derived DDIM, a deterministic sampling algorithm given an initial noise $\boldsymbol{x}_T$. By applying this algorithm in the reverse order (a.k.a. DDIM inversion) starting from the clean $\boldsymbol{x}_0$, it allows to obtain the intermediate noisy images $\{\boldsymbol{x}_i\}_{t=1}^{T}$ used to generate it.

Stable Diffusion Stable Diffusion (SD) [38] is a prominent text-to-image diffusion model that operates in a latent image space. A pretrained encoder maps RGB images to this space, and a decoder decodes latents back to high-resolution images. In more detail, SD is based on a U-Net architecture [39], which comprises of residual, self-attention, and cross-attention blocks. The residual block convolves the activations from a previous layer, while cross-attention manipulates features according to the text prompt.

In the self-attention block, features are projected into queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$. The output of the block is given by:

$$\boldsymbol{A} \cdot \boldsymbol{V} \quad \text{where} \quad \boldsymbol{A} = \texttt{Attention}(\boldsymbol{Q}; \boldsymbol{K}) \tag{1}$$

The `Attention` operation [52] computes the affinities be-

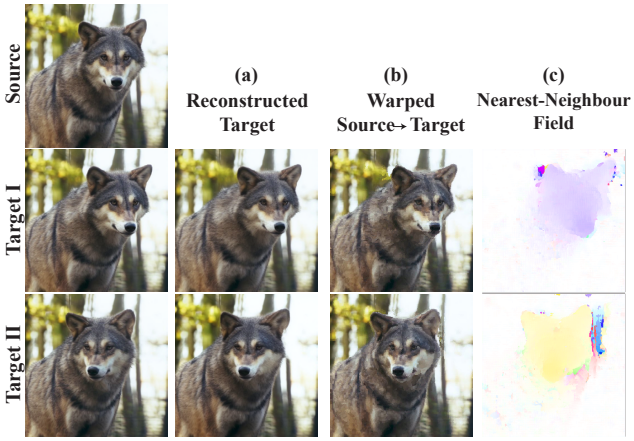

Figure 3. **Fine-grained feature correspondences.** Features (i.e., output tokens from the self-attention modules) extracted from of a source frame are used to reconstruct nearby frames. This is done by: (a) swapping each feature in the target by its nearest feature in the source, in all layers and all generation time steps, and (b) simple warping in RGB space, using a nearest neighbour field (c), computed between the source and target features extracted from the highest resolution decoder layer. The target is faithfully reconstructed, demonstrating the high level of spatial granularity and shared content between the features.

tween the d-dimensional projections Q, V. Formally,

$$\texttt{Attention}(\boldsymbol{Q};\boldsymbol{K}) = \texttt{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right) \quad (2)$$

4. Method

Given an input video $\boldsymbol{\mathcal{I}} = [\boldsymbol{I}^1, ..., \boldsymbol{I}^n]$, and a text prompt $\boldsymbol{\mathcal{P}}$ describing the target edit, our goal is to generate an edited video $\boldsymbol{\mathcal{J}} = [\boldsymbol{J}^1, ..., \boldsymbol{J}^n]$ that adheres to the text $\boldsymbol{\mathcal{P}}$, while preserving the original motion and semantic layout of $\boldsymbol{\mathcal{I}}$. To achieve this, our framework leverages a pretrained and fixed text-to-*image* diffusion model ϵ_θ.

Naïvely leveraging ϵ_θ for *video* editing, by applying an image editing method on each frame independently (e.g., [12, 51, 29, 56]), results in content inconsistencies across frames (e.g., Fig. 2 middle column). Our key finding is that these inconsistencies can be alleviated by enforcing consistency among the internal diffusion features across frames, during the editing process.

Natural videos typically depict coherent and shared content across time. We observe that the internal representation of natural videos in $\epsilon_\theta$has similar properties. This is illustrated in Fig. 2, where we visualize the features extracted from a given video (first column). As seen, the features depict a shared and consistent representation across frames, i.e., corresponding regions exhibit similar representation. We further observe that the original video features provide fine-grained correspondences between frames, using a simple nearest neighbour search (Fig 3). Moreover, we show that these *corresponding features are interchangeable for the diffusion model* – we can faithfully synthesize one frame by swapping its features by their corresponding ones in a nearby frame (Fig 3(a)).

Nevertheless, when an edit is applied to each frame individually, the consistency of the features breaks (Fig. 2 middle column). This implies that the level of consistency of in RGB space is correlated with the consistency of the internal features of the frames. Hence, our key idea is to manipulate the features of the edited video to preserve the level of consistency and inter-frame correspondences of the original video features.

As illustrated in Fig. 4, our framework, dubbed *Token-Flow*, alternates at each generation timestep between two main components: (i) sampling a set of keyframes and jointly editing them according to $\boldsymbol{\mathcal{P}}$; this stage results in shared global appearance across the keyframes, and (ii) propagating the features from the keyframes to all of the frames based on the correspondences provided by the original video features; this stage explicitly preserves the consistency and fine-grained shared representation of the original video features. Both stages are done in combination with an image editing technique $\hat{\epsilon_\theta}$(e.g, [51]). Intuitively, the benefit of alternating between keyframe editing and propagation is twofold: first, sampling random keyframes at each generation step increases the robustness to a particular selection. Second, since each generation step results in more consistent features, the sampled keyframes in the next step will be edited more consistently.

Pre-processing: extracting diffusion features. Given an input video $\boldsymbol{\mathcal{I}}$, we apply DDIM inversion (see Sec. 3) on each frame $\boldsymbol{I}^i$, which yields a sequence of latents $[\boldsymbol{x}_1^i, ..., \boldsymbol{x}_T^i]$. For each generation timestep t, we feed the latent $\boldsymbol{x}_t^i$ of each frame $i \in [n]$ to the model and extract the tokens $\boldsymbol{\phi}(x_t^i)$ from the self-attention module of every layer in the network ϵ_θ(fig. 4, top). We will later use these tokens to establish inter-frame correspondences between diffusion features.

4.1. Keyframe Sampling and Joint Editing

Our observations imply that given the features of a single edited frame, we can generate the next frames by propagating its features to their corresponding locations.

Most videos, however, can not be represented by a single keyframe. To account for that, we consider multiple keyframes, from which we obtain a set of features (tokens), $\boldsymbol{T}_{base}$, that will later be propagated to the entire video.

Specifically, at each generation step, we randomize a set of keyframes $\{\boldsymbol{J}^i\}_{i\in\kappa}$ in fixed frame intervals (see SM for details).

We joinly edit the keyframes by extending the self-attention block to simultaneously process them [53], and thus encourage that they share a global appearance. In more detail, the input to the modified block are the self-attention features from all keyframes $\{\boldsymbol{Q}^i\}_{i\in\kappa}, \{\boldsymbol{K}^i\}_{i\in\kappa}, \{\boldsymbol{V}^i\}_{i\in\kappa}$

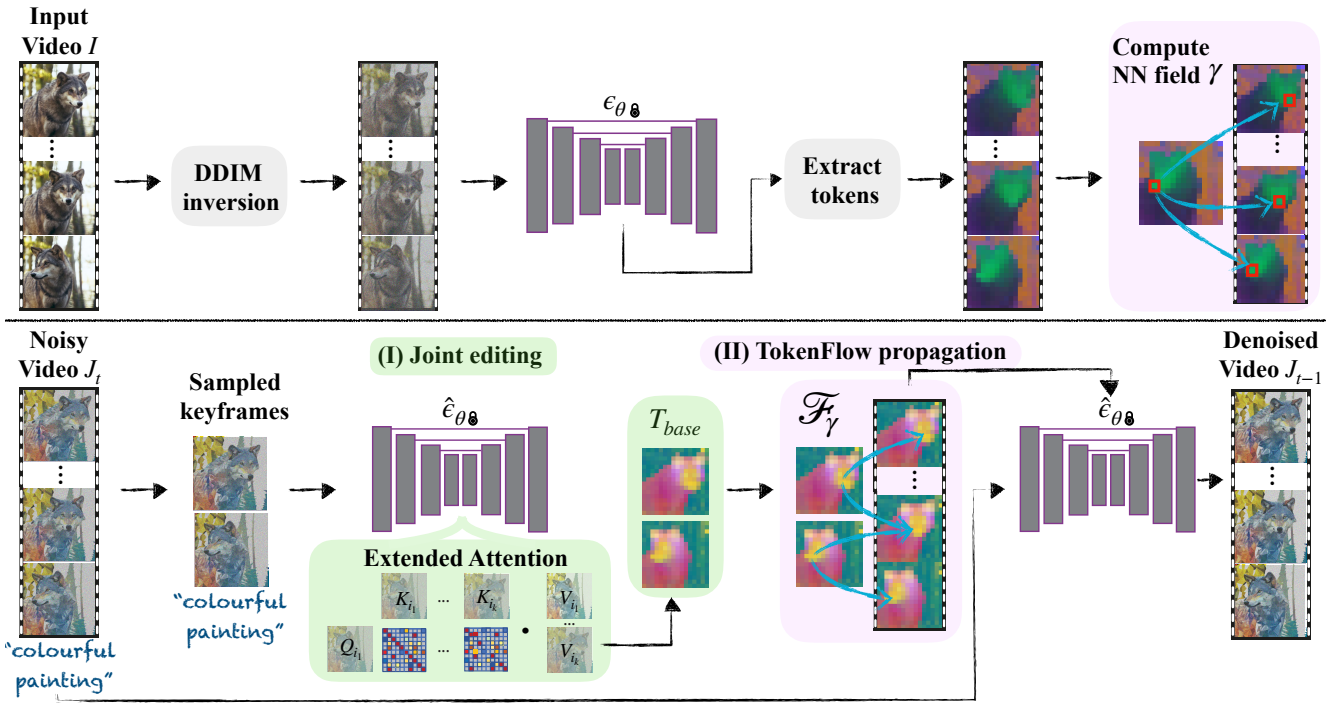

Figure 4. **TokenFlow pipeline**. Top: Given an input video $\mathcal{I}$, we DDIM invert each frame, extract its tokens, i.e., output features from the self-attention modules, from each timestep and layer, and compute inter-frame features correspondences using a nearest-neighbor (NN) search. Bottom: The edited video is generated as follows: at each denoising step t, (I) we sample keyframes from the noisy video J_t and jointly edit them using an extended-attention block; the set of resulting edited tokens is $\mathbf{T}_{base}$. (II) We propagate the edited tokens across the video according to the pre-computed correspondences of the original video features. To denoise J_t, we feed each frame to the network, and replace the generated tokens with the tokens obtained from the propagation step (II).

where $\boldsymbol{Q}^i, \boldsymbol{K}^i, \boldsymbol{V}^i$ are the queries, keys, and values of frame $i \in \kappa, \kappa = \{i_1, ...i_k\}$. The keys of all frames are concatenated, and an extended-attention is calculated:

$$\texttt{ExtAttn}\Big(\boldsymbol{Q}^i; [\boldsymbol{K}^{i_1}, \dots \boldsymbol{K}^{i_k}]\Big) = \text{Softmax}\left(\frac{\boldsymbol{Q}^i \left[\boldsymbol{K}^{i_1}, \dots \boldsymbol{K}^{i_k}\right]^T}{\sqrt{d}}\right) \tag{3}$$

The output of the block for frame i is given by:

$$\begin{aligned} &\phi(\boldsymbol{J}^i) = \hat{\boldsymbol{A}} \cdot [\boldsymbol{V}^{i_1}, \dots \boldsymbol{V}^{i_k}] \\ &\text{where} \quad \hat{\boldsymbol{A}} = \texttt{ExtAttn}\Big(\boldsymbol{Q}^i; [\boldsymbol{K}^{i_1}, \dots \boldsymbol{K}^{i_k}]\Big) \end{aligned} \tag{4}$$

And we define $\mathbf{T}_{base} = \{\phi(\boldsymbol{J}^i)\}_{i \in \kappa}$, for each layer in the network (Fig. 4 bottom left). Intuitively, each keyframe queries all other keyframes, and aggregates information from them. This results in a roughly unified appearance in the edited frames [53, 19, 5, 34].

4.2. Edit Propagation via TokenFlow

Given $\mathbf{T}_{base}$, we propagate it across the video based on the token correspondences extracted from the original video. At each generation step t, we compute the nearest neighbor (NN) of each original frame's tokens, $\boldsymbol{\phi}(x_t^i)$, and its two adjacent keyframess' tokens, $\boldsymbol{\phi}(x_t^{i+})$, $\boldsymbol{\phi}(x_t^{i-})$ where $i+$ is the index of the closest future keyframe, and $i-$ the index of the closest past keyframe. Denote the resulting NN fields γ^{i+}, γ^{i-}:

$$\gamma^{i\pm}[p] = \arg\min_q \mathcal{D}\left(\phi(\boldsymbol{x}^i)[p], \phi(\boldsymbol{x}^{i\pm})[q]\right) \tag{5}$$

Where p, q are spatial locations in the token feature map, and $\mathcal{D}$ is the cosine distance. For simplicity, we omit the notations of the generation timestep t; our method is applied in all time-steps and self-attention layers. Once we obtain $\gamma^{\pm}$, we use it to propagate the *edited* frames' tokens $\mathbf{T}_{base}$ to the rest of the video, by linearly combining the tokens in $\mathbf{T}_{base}$ corresponding to each spatial location p and frame i:

$$\begin{aligned} \mathcal{F}_{\gamma}(\mathbf{T}_{base}, i, p) = w_i \cdot \phi(\boldsymbol{J}^{i+})[\gamma^{i+}[p]] + \\ (1 - w_i) \cdot \phi(\boldsymbol{J}^{i-})[\gamma^{i-}[p]] \end{aligned} \tag{6}$$

Where $\phi(\boldsymbol{J}^{i\pm}) \in \mathbf{T}_{base}$ and $w_i \in (0, 1)$ is a scalar proportional to the distance between frame i and its adjacent keyframes (see SM), ensuring a smooth transition. Note that $\mathcal{F}$ also modifies the tokens of the sampled keyframes. That is, we modify the self-attention blocks to output a linear combination of the tokens in $\mathbf{T}_{base}$.

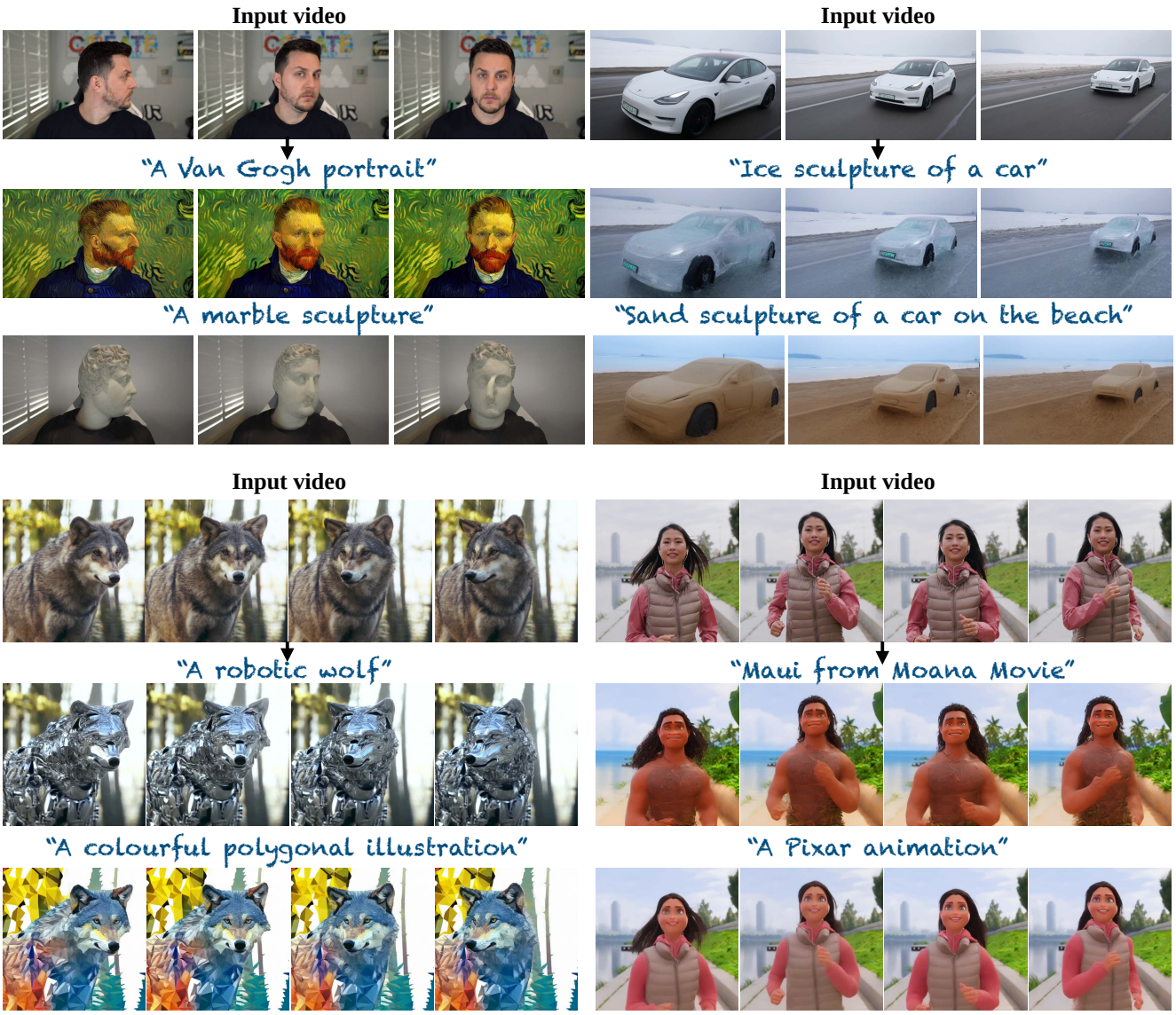

Figure 5. **Results.** Sample results of our method. We refer the reader to our webpage and SM for more examples and full-video results.

Overall algorithm We summarize our video editing algorithm in Alg. 1: We first perform DDIM inversion on the input video $\mathcal{I}$ and extract the sequence of noisy latents $\{\boldsymbol{x_t^i}\}_{t=1}^{T}$ for all frames $i \in [n]$ (fig 4, top). We then denoise the video, alternating between keyframes editing and *TokenFlow* propagation: At each generation step t, we randomize $k < n$ keyframe indices, and denoise them using an image editing technique (e.g., [51, 29]) combined with extended-attention (Eq. 4, Fig. 4 (I)). We then denoise the entire video $\mathcal{J}_t$ by combining the image-editing technique with *TokenFlow* (Eq. 6, Fig. 4 (II)) at every self-attention block in every layer of the network. Note that each layer includes a residual connection between the input and output of the self-attention block, thus performing *TokenFlow* at each layer is necessary.

5. Results

We evaluate our method on DAVIS videos [33] and on Internet videos depicting animals, food, humans, and various objects in motion.

The spatial resolution of the videos is 384×672 or 512×512 pixels, and they consist of 40 to 200 frames. We use various text prompts on each video to obtain diverse editing results. Our evaluation dataset comprises of 61 text-video pairs. We utilize PnP-Diffusion [51] as the frame editing method, and we use the same hyper-parameters for all our results. PnP-Diffusion may fail to accurately preserve the structure of each frame due to inaccurate DDIM inversion (see Fig. 2, middle column, right frame: the dog's head is distorted). Our method improves robustness to this, as

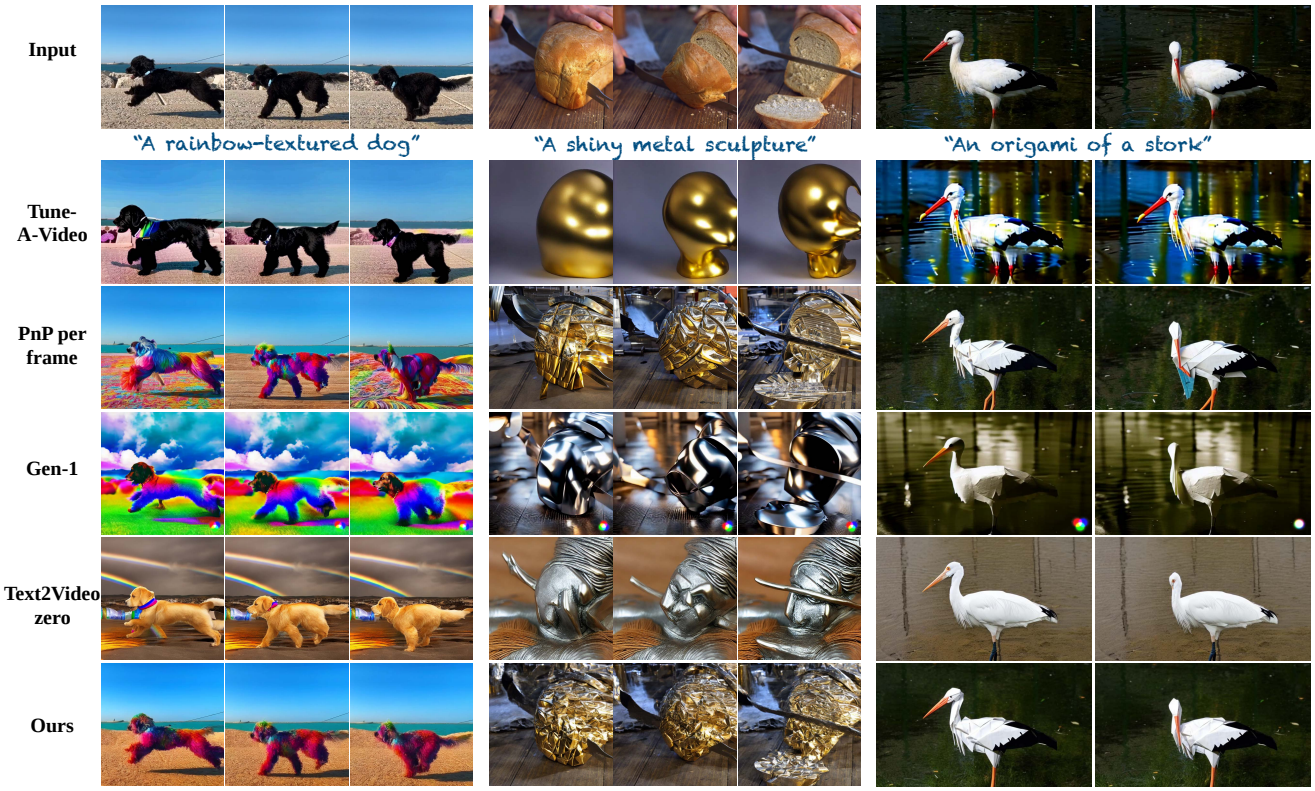

Figure 6. **Comparison.** We compare our method against Tune-A-Video [53], PnP-Diffusion [51] applied per frame, Gen-1 [10], and Text2Video-Zero[20]. We refer the reader to our webpage for full-video comparisons.

Algorithm 1 TokenFlow editing

Input:

$\mathcal{I} = [\mathbf{I}^1, ..., \mathbf{I}^n]$ ▷ Input Video

$\boldsymbol{P}$ ▷ Target text prompt

$\hat{\boldsymbol{\Psi}}$ ▷ Diffusion-based image editing technique

$\{\mathbf{x}_t^i\}_{t=1}^T, \{\phi(x_i)\}_{i=1}^n \leftarrow$ DDIM-Inv$[\mathbf{I}^i] \quad \forall i \in [n],\ t \in [T]$

$\mathbf{J}_T^1, \ldots, \mathbf{J}_T^n \leftarrow \mathbf{x}_T^1, \ldots, \mathbf{x}_T^n$

For $t = T, \ldots, 1$ **do**

$\mathcal{K} = \{i_1, \ldots, i_k\} \leftarrow$ sample keyframe indices

$\mathcal{F}_\gamma \leftarrow \gamma^{i\pm} \ \forall i \in [n]$ compute NN field

$\{\mathbf{J}_{t-1}^j\}_{j\in\mathcal{K}} \leftarrow \hat{\boldsymbol{\Psi}}[\{\mathbf{J}_t^j\}_{j\in\mathcal{K}}; \texttt{ExtAttn}]$

$\mathbf{T}_{\text{base}} \leftarrow \phi(\{\mathbf{J}_{t-1}^j\}_{j\in\mathcal{K}})$ extract keyframes' tokens

$\mathbf{J}_{t-1} \leftarrow \hat{\boldsymbol{\Psi}}[\mathbf{J}_t; \text{TokenFlow}(\mathcal{F}_\gamma(\mathbf{T}_{\text{base}}))]$

Output: $\mathcal{J} = [\mathbf{J}_0^1, \ldots, \mathbf{J}_0^n]$

multiple frames contribute to the generation of each frame in the video. Our framework can be combined with any diffusion-based image editing technique that accurately preserves the structure of the images; results with different image editing techniques (e.g. [29]) are available in the SM. We refer the reader to the SM for implementation details and video results.

Figures 5, 1 show sample frames from the edited videos. Our edits are temporally consistent and adhere to the edit prompt. The man's head is changed to Van-Gogh or marble (top left); importantly, the man's identity and the scene's background are consistent throughout the video. The patterns of the polygonal wolf (bottom left) are the same across time: the body is *consistently* orange while the chest is blue. More video examples can be found in the SM.

Baselines. We compare our method to state-of-the-art, and concurrent works: (i) Text2Video-Zero [19] that utilizes ControlNet [56] for video editing using self-attention inflation. (ii) Tune-a-Video [53] that fine-tunes the text-to-image model on the given test video. (iii) Gen-1 [10], a video diffusion model that was trained on a large-scale image and video dataset. (iv) Text2LIVE [1] which utilize a layered video representation (NLA) [18] and perform test-time training using CLIP losses. Note that NLA requires foreground/background separation masks and takes $\sim$ 10 hours to train. We therefore compare to them on DAVIS videos for which an NLA model is available. We additionaly consider the two following baselines: (i) Per-frame diffusion-based image editing baseline, PnP-Diffusion [51]. (ii) Applying PnP-Diffusion on a single keyframe and propagate the edit to the entire video using [17].

5.1. Qualitative evaluation

Fig. 6 provides a qualitative comparison of our method to four prominent baselines; please refer to SM for the full videos. Our method (bottom row) outputs videos that better adhere to the edit prompt while maintaining temporal consistency of the resulting edited video, while other methods struggle to meet both these goals. Tune-A-Video [53] (second row) inflates the 2D image model into a video model, and fine-tunes it to overfit the motion of the video; thus, it is suitable for short clips. For long videos it struggles to capture the motion resulting with meaningless edits, e.g., the shiny metal sculpture. Applying PnP for each frame independently (third row) results in exquisite edits adhering to the edit prompt but, as expected, lack any temporal consistency. The results of Gen-1 [10] (fourth row) also suffer from some temporal inconsistencies (the beak of the origami stork changes color); Moreover, their frame quality is significantly worst than that of a text-to-image diffusion model. The edits of Text2Video-Zero [19] (fifth row) suffer from severe jittering as this method relies heavily on the extended attention mechanism to *implicitly* encourage consistency.

Fig 7 shows an additional qualitative comparison of our method to Text2LIVE [1] and to propagating an edit from a single keyframe (edited using [51]) to the rest of the video using [17]. Text2LIVE lacks a strong generative prior, thus, as seen in row 3, has limited visual quality. Additionally, this method relies on a layered representation of the video ([18]), which takes around 10 hours to train and is limited to videos with simple motion. Using [17] to propagate the edit produces propagation artifacts on frames that are not near the edited keyframe (row 2).

5.2. Quantitative evaluation

We evaluate our method in terms of: (i) *edit fidelity* measured by computing the average similarity between the CLIP embedding [35] of each edited frame and the target text prompt; (ii) *temporal consistency.* Following [5, 22], temporal consistency is measured by computing the optical flow of the original video using [49], warping the edited frames according to it, and measuring the warping error.

Table 1 compares our method to baselines. Our method achieves the highest CLIP score, showing a good fit between the edited video and the input guidance prompt. Furthermore, our method has the lowest warp error, indicating temporally consistent results.

Additionally, we consider the reference baseline of passing the original video through the LDM auto-encoder without performing editing (*LDM recon.*). This baseline provides an upper bound on the temporal consistency achievable by LDM auto-encoder. As expected, the CLIP similarity of this baseline is poor as it does not involve any editing. However, this baseline does not achieve zero warp error either due to the imperfect reconstruction of the LDM auto-encoder, which hallucinates high-frequency information.

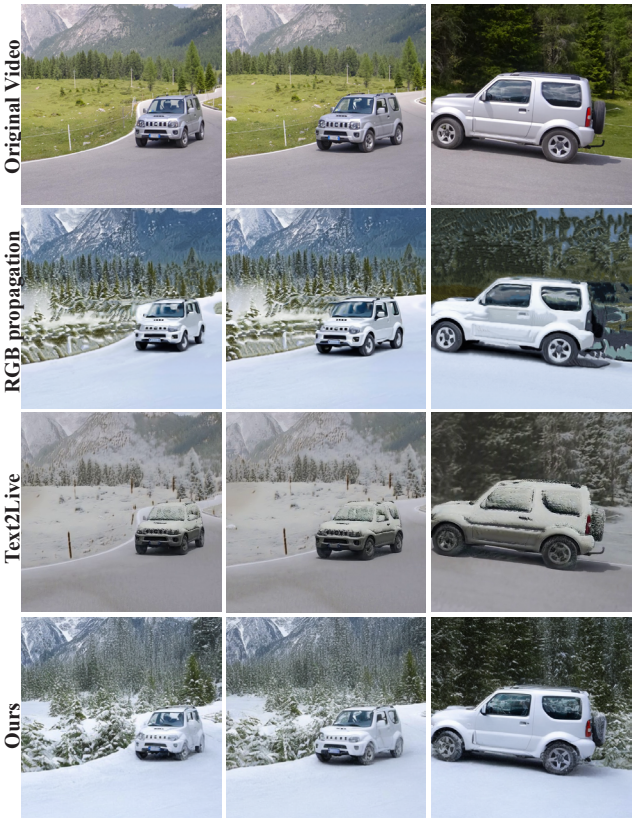

Figure 7. **Additional qualitative comparison.** RGB propagation ([17]) only has access to low-level cues such as optical flow, resulting with visual artifacts in videos with content reveal or complex dynamics. Text2LIVE ([1]) is CLIP-based and does not utilize the generative prior of a diffusion model, and is thus more limited in terms of visual quality.

	Warp-error ($\times 10^{-3}$) ↓	CLIP-score↑
LDM recon.	2.0	0.23
PnP-Diffusion	11.3	0.33
Text2Video-Zero	12.5	0.33
Tune-a-Video	30.0	0.31
Ours *w/o TokenFlow*	5.9	0.33
Ours *w/o random keyframes*	3.7	0.33
Ours	**3.0**	**0.33**

Table 1. We evaluate our method in temporal consistency by computing warp-error and in fidelity to the target text prompt using CLIP similarity. See Sec. 5 for more details.

5.3. Ablation study

We ablate the main design choice in our framework.

First, we ablate the use of *TokenFlow*, Sec. 4.2, for enforcing temporal consistency (*w/o TokenFlow*). In this experiment, we replace *TokenFlow* with extended attention (Eq. 4) and compute it between all frames of the edited video. Note that this operation is computationally demanding and does not scale well with the duration of the video.

Figure 8. **Limitations.** Our method edits the video according to the feature correspondences of the original video, hence it cannot handle edits that requires structure deviations.

Second, we ablate the randomizing of the keyframe selection at each generation step (*w/o random keyframes*). In this experiment, we use the same keyframe indices (evenly spaced in time) across the generation.

Table 1 (bottom) shows the quantitative results of our ablations, the resulting videos can be found in the SM. As seen, *TokenFlow* ensures higher degree of temporal consistency, indicating that solely relying on the extension of self-attention to multiple frames is insufficient for achieving fine-grained temporal consistency.

Additionally, fixing the keyframes creates an artificial partition of the video into short clips between the fixed keyframes. This partition reflects poorly on the temporal consistency of the result, as seen by the higher warp error. This effect can be visually seen in the ablation videos in the SM.

6. Discussion

We presented a new framework for text-driven video editing using an image diffusion model. We study the internal representation of a video in the diffusion feature space, and demonstrate that consistent video editing can be achieved via consistent diffusion feature representation during the generation. Our method outperforms existing baselines, demonstrating a significant improvement in temporal consistency. As for limitations, our method is tailored to preserve the motion of the original video, and as such, it cannot handle edits that require structural changes (Fig 8.) Moreover, our method is built upon a diffusion-based image editing technique to allow the structure preservation of the original frames. When the image-editing technique fails to preserve the structure, our method enforces correspondences that are meaningless in the edited frames, resulting in visual artifacts. Lastly, the LDM decoder introduces some high frequency flickering [3]. A possible solution for this would be to combine our framework with an improved decoder (e.g., [3], [57])

We note that this minor level of flickering can be easily eliminated with exiting post-process deflickering (see SM). Our work shed new light on the internal representation of natural videos in the space of diffusion models (e.g., temporal redundancies), and how they can be leveraged for enhancing video synthesis. We believe it can inspire future research in harnessing image models for video tasks, and for the design of text-to-video models.

7. Acknowledgement

We thank Narek Tumanyan for his valuable comments and discussion. We thank Hila Chefer for proofreading the paper. This project received funding from the Israeli Science Foundation (grant 2303/20), the Carolito Stiftung, and the NVIDIA Applied Research Accelerator Program. Dr. Bagon is a Robin Chemers Neustein AI Fellow.

References

[1] 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. 7, 8

[2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 2

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing, 2023. 2

[5] Duygu Ceylan, Chun-Hao Paul Huang, and Niloy Jyoti Mitra. Pix2video: Video editing using image diffusion. *ArXiv*, abs/2303.12688, 2023. 1, 3, 5, 8

[6] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023. 2

[7] V. Cheung, B.J. Frey, and N. Jojic. Video epitomes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. 2

[8] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022. 2, 3

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 2021. 2, 3

[10] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023. 2, 7, 8, 11

[11] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133*, 2023. 2

[12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2, 4

[13] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben

Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. 2, 3

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2

[16] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. *arXiv preprint arXiv:2210.00939*, 2022. 2

[17] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Stylizing video by example. *ACM Transactions on Graphics*, 2019. 7, 8

[18] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 2021. 2, 7, 8

[19] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023. 1, 5, 7, 8, 11

[20] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *ArXiv*, abs/2303.13439, 2023. 3, 7

[21] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 2

[22] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, 2018. 8

[23] Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing. *arXiv preprint arXiv:2301.13173*, 2023. 2

[24] Chenyang Lei, Xuanchi Ren, Zhaoxiang Zhang, and Qifeng Chen. Blind video deflickering by neural filtering with a flawed atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 2

[25] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. In *Advances in Neural Information Processing Systems*, 2020. 2

[26] Shaoteng Liu, Yuecheng Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *ArXiv*, abs/2303.04761, 2023. 3

[27] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *arXiv*, 2023. 2

[28] Wan-Duo Kurt Ma, JP Lewis, W Bastiaan Kleijn, and Thomas Leung. Directed diffusion: Direct control of object placement through attention guidance. *arXiv preprint arXiv:2302.13153*, 2023. 2

[29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 2, 4, 6, 7

[30] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[31] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2, 3

[32] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. *arXiv preprint arXiv:2303.11306*, 2023. 2

[33] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[34] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv:2303.09535*, 2023. 1, 2, 3, 5, 11

[35] 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*. PMLR, 2021. 2, 8

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

[37] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 2

[38] 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

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[40] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *Pattern Recognition - 38th German Conference (GCPR)*, 2016. 2

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine

Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. *ArXiv*, abs/2210.08402, 2022. 2

[43] Oded Shahar, Alon Faktor, and Michal Irani. Space-time super-resolution from a single video. In *CVPR 2011*, 2011. 2

[44] Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022. 2

[45] Chaehun Shin, Heeseung Kim, Che Hyun Lee, Sang gil Lee, and Sung-Hoon Yoon. Edit-a-video: Single video editing with object-aware consistency. *ArXiv*, abs/2303.07945, 2023. 3

[46] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 2

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2, 3

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 3

[49] 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*. Springer, 2020. 8

[50] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics*, 2020. 2

[51] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to image translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, June 2023. 2, 3, 4, 6, 7, 8, 11

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[53] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. 1, 2, 3, 4, 5, 7, 8, 11

[54] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2016. 2

[55] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *arXiv preprint arxiv:2305.15347*, 2023. 2

[56] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 4, 7

[57] Zixin Zhu, Xuelu Feng, Dongdong Chen, Jianmin Bao, Le Wang, Yinpeng Chen, Lu Yuan, and Gang Hua. Designing a better asymmetric vqgan for stablediffusion, 2023. 9

A. Implementation Details

StableDiffusion. We use Stable Diffusion as our pre-trained text-to-image model; we use the *StableDiffusion-v-2-1* checkpoint provided via official HuggingFace webpage.

DDIM inversion. In all of our experiments, we use DDIM deterministic sampling with 50 steps. For inverting the video, we follow [51] and use DDIM inversion with classifier-free guidance scale of 1 and 1000 forward steps; and extract the self-attention input tokens from this process similarly to [34].

Runtime. Since we don't compute the attention module on most video frames (i.e., we only compute the self-attention output on the keyframes) our method is efficient in run-time, and the sampling of the video reduces the time of per-frame editing by 20%. The inversion process with 1000 steps is the main bottleneck of our method in terms of run-time.

Hyper-parameters. In equation 6 we set w_i to be:

$$\begin{aligned} w_i &= \sigma(d_-/(d_+ + d_-)) \\ \text{where } d_+ &= ||i - i^+||, d_- = ||i - i^-|| \end{aligned} \tag{7}$$

where σ is a sigmoid function, i^+ and i^- are the future and past neighboring keyframes of i, respectively.

For sampling the edited video we set the classifier-free guidance scale to 7.5. At each timestep, we sample random keyframes in frame intervals of 8.

Baselines. For running the baseline of Tune-a-video [53] we used their official repository. For Gen-1 [10] we used their platform on Runaway website. This platform outputs a video that is not in the same length and frame-rate as the input video; therefore, we could not compute the warping error on their results. For text-to-video-zero [19] we used their official repository, with their depth conditioning configuration.