Zero-shot Customized Video Editing with Diffusion Feature Transfer

Wei Chen[1], Huidong Liu[2†], Yang Liu[2], Chien-Chih Wang[2], Moyan Li[2], Hongdong Li[2,3], Bryan Wang[2]
[1] Purdue University [2] Amazon.com, Inc. [3] Australian National University
chen2732@purdue.edu, {liuhuido, yliuu, ccwang, moyanli, hongdli, brywan}@amazon.com

Figure 1. We present FreeMix, a zero-shot approach for customized video editing. Given a source video, reference images, their text prompts, and a target text prompt where the source video's object name is replaced with that of the reference images, FreeMix generates a synthesized video by replacing the object in the source video with the object in the reference images. The key innovation is to transfer the low-frequency information from the source video and high-frequency information from the reference images to the synthesized video.

Abstract

Customized video editing aims at substituting the object in a given source video with a target object from reference images (Fig. 1). Existing approaches often rely on fine-tuning pre-trained models by learning the appearance of the objects in the reference images, as well as the temporal information from the source video. These methods are however not scalable as fine-tuning is required for each source video and each object to be customized, incurring computational overhead. More importantly, such individual customization often leads to overfitting to a few given reference images. In this paper, by leveraging the pre-trained Stable Diffusion model, we propose FreeMix, a zero-shot customized video editing approach. From careful empirical analysis of the diffusion features, we observe that the motion information of the moving object in the source video are captured by the low-frequency high-level diffusion features, while high-frequency low-level diffusion features encode more the object's appearance in the reference images. By exploiting this observation, we achieve effective motion transfer from the source video and appearance transfer from reference images to synthesize the output video via simple feature transfer in the diffusion model. To enhance temporal consistency of the synthesized video, we apply optimal transport to low-level diffusion features of consecutive source video frames, establishing feature correspondences to guide video generation. We conduct comprehensive experiments, demonstrating that our approach surpasses both state-of-the-art customized video editing methods that require fine-tuning and general text-based video editing methods.

1. Introduction

Given a source video and a few reference images, customized video editing (aka. subject-driven video editing, or personalized video editing) aims to create a novel synthesized video by modifying the main object in the source video with the foreground object depicted in the reference images. There are three key requirements (and challenges) in implementing a satisfactory customized video editing method: (i) accurate motion transfer, which means accurately transfer the motion of the object from the source

†Corresponding author.

video to the synthesized video, (ii) precise concept alignment, which aims to faithfully transfer the appearance of the object in the reference images to the synthesized video, and (iii) temporal consistency, which ensures smooth transitions across neighboring video frames.

A myriad of methods have been proposed to address the motion transfer and temporal consistency challenges. Tune-A-Video [46] adds temporal layers to the Stable Diffusion (SD) model [34] and learns temporal information from each source video, albeit incurring additional training efforts. AnimateDiff [14] trains the temporal layers to capture video motion by training on large datasets. MotionDirector [54] and VideoSwap [13], utilize temporal weights pre-trained on large datasets and fine-tune them for each source video [8, 19, 21]. Flow-based methods, such as FlowVid [24, 47] and TokenFlow [11], utilize optical flow, or nearest-neighbor feature correspondences from the source video to regularize the temporal information of the synthesized video. The concept alignment challenge has received increasing attention in recent years. Most existing methods [10, 12, 22, 35, 44, 54] typically train a customized checkpoint or embedding for each subject. However, these approaches tend to overfit to the given reference images [49], compromising the generalizability of the foundational models. Additionally, they are not scalable, as they demand extra computational resources and time for fine-tuning each subject. The recent method VideoSwap [13] leverages pre-trained personalization weights [12] for each subject/concept but requires manual adjustment of key-points to modify the shape of the target concept.

In this paper, we propose a novel method to address customized video editing tasks without fine-tuning, leveraging the diffusion features from the pre-trained SD model [34]. The example result in Fig. 1 shows the effectiveness of our method. We have the following contributions:

- We perform a comprehensive analysis of the diffusion features and observe that the low-frequency high-level diffusion features capture video motion, while high-frequency low-level diffusion features capture object appearance in images.
- We propose FreeMix, a zero-shot customized video editing approach that can take multiple reference images for customized video editing.
- To enhance temporal consistency in the synthesized video, we propose using optimal transport to establish feature correspondences between consecutive frames in the source video. We apply the feature correspondences in generating the synthesized video to improve its temporal consistency.
- Our method outperforms State-Of-The-Art (SOTA) customized video editing approaches that require fine-tuning and surpasses general text-based video editing methods as well.

2. Related Work

Diffusion-Based Video Editing. Recent advancements in diffusion models [6, 17, 36, 37] have made it possible for users to generate high-quality visual contents using just simple text prompts [6, 34]. These models have been adapted for text-to-video generation by expanding spatial attention to spatio-temporal attention [18] and conducting large-scale training on video datasets, leading to the development of several foundational models for video generation [8, 19, 21]. Recently, several approaches have been introduced to enable video editing using the Stable Diffusion [7] model , such as Tune-A-Video [46], FateZero [32], and so on. FateZero [32] and Video-P2P [26] extract cross- and self-attention from the source video to control spatial layout during video editing. Rerender-A-Video [47], ControlVideo [52], and TokenFlow [11] extract and align optical flow, depth/edge maps, and nearest-neighbor feature correspondences from the source video, respectively, encouraging temporal consistency of the synthesized video. StableVideo [2] and CoDEF [30] learn the canonical space for editing following the Layered Neural Atlas or the deformation field. Other approaches benefit from the pre-trained temporal weights of video diffusion models and achieve video editing with improved temporal consistency [48, 54].

Concept Customization. Concept customization [10, 35] refers to generating personalized or specific concepts, enabling the creation of visual contents that closely match the desired objects. Current methods can be categorized into tuning-based approaches and tuning-free solutions. Tuning-based approaches learn from multiple reference images to capture variations in concept appearance under different conditions, such as pose, viewing angle, and so on. Typical methods are DreamBooth, Text Inversion, and LoRA-based methods used in VideoSwap [13], ModionDirector [54], etc. The Splice [39, 40] trains a generator to disentangle structure and appearance for customized image editing. However, these methods tend to overfit to the given reference images [49]. Moreover, they require additional computational resources to learn each concept, limiting their scalabilities [3, 10, 12, 22, 35]. In contrast, tuning-free solutions are computationally efficient but typically only accept one reference image to stick closely to the spatial layout of the provided reference image, making them inflexible to change to other spatial layouts. Among these methods, PnP [41] and MasaCtrl [1] extract features from the self-attention layers in diffusion models, use them as plug-and-play components, and inject them into specific layers of the Diffusion UNet for image generation. Similarly, P2P [15] and P+ [44] leverage cross-attention features to transfer the appearance from a reference image to a target scene.

Temporal Consistency in Video Editing. Maintaining proper temporal control is essential in video editing to en-

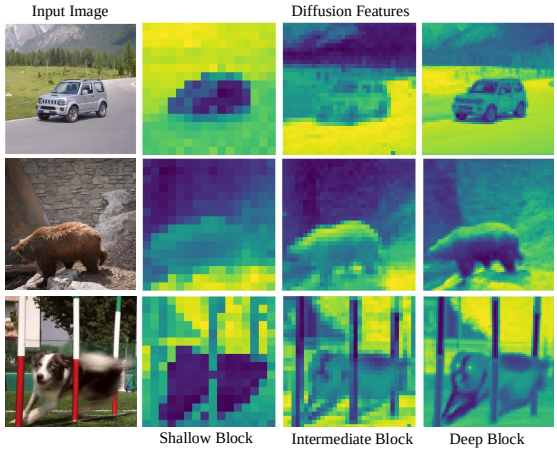


Figure 2. Visualization of features of input images from different Stable Diffusion UNet blocks. Shallow Block, Intermediate Block, and Deep Block are UNet decoder blocks corresponding to $4\times$, $2\times$, and $1\times$ downsampling of the latent features, respectively. The Shallow Block encodes localized semantic information, *e.g.*, the location of the car. The Deep Block captures fine-grained details, including car tire texture, bear shape and dog eyes.

sure consistency across frames. Some approaches use the depth maps, canny edges, or poses as used in ControlNet or ControlVideo to ensure the temporal consistency across frames [9, 28, 47, 50, 52, 53]. However, a strong control from a depth map or canny edge prevents shape changes between different objects. TokenFlow [11] propagates edited features from key frames to other frames through a nearest-neighbor approach to ensure smooth video frame transitions. The FLATTEN [4] uses the optical flow to guide the attention operation across frames to improve the temporal consistency. However, relying on nearest-neighbor feature similarities can easily cause misalignment, leading to flickering between frames.

3. Method

3.1. Analysis of Stable Diffusion Features

Inspired by PnP [41], we conducted experiments to analyze the information encoded in different layers of the SD model. A key observation reveals that the attention layers in the SD model encode significant information related to the overall structure, layout, and content of the images. As illustrated in Fig. 2, the diffusion features in the UNet decoder blocks exhibit distinct characteristics: low-frequency information is primarily contained in the shallow decoder blocks, while high-frequency information is predominantly contained in the deep decoder blocks of the UNet.

Attention. Each UNet block contains a self-attention layer. In the self-attention layer, the input feature $\mathbf{f}$ is projected into queries $\mathbf{Q} = \mathbf{f}\mathbf{W}_q$, keys $\mathbf{K} = \mathbf{f}\mathbf{W}_k$, and values $\mathbf{V} = \mathbf{f}\mathbf{W}_v$. The Attention [42] operation computes the affinities between the d-dimensional projections $\mathbf{Q}$, $\mathbf{K}$, and then multiplied by $\mathbf{V}$ to yield the output of the layer:

$$\mathbf{O} = \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d}}\right)\mathbf{V}. \quad (1)$$

The self-attention layer is illustrated in Fig. 3 (a).

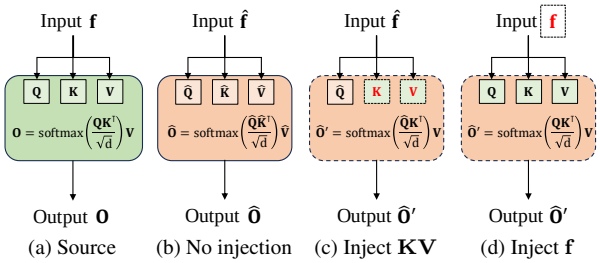


Figure 3. Illustration of the attention layer in the Stable Diffusion UNet decoder for different image synthesize processes.

Diffusion Features. In this work, we analyze diffusion features

$$g(\mathbf{X}) = \{\mathbf{f}, \mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{O}\}, \quad (2)$$

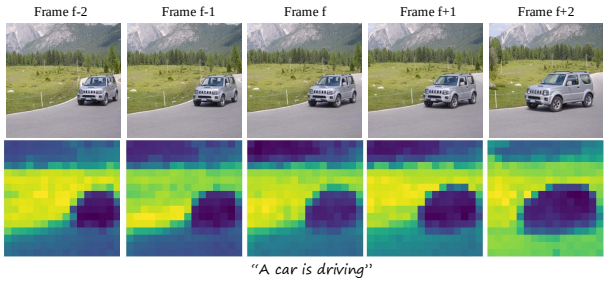
from the self-attention layer of the UNet decoder, where $\mathbf{f}$ and $\mathbf{O}$ are the input and output, respectively, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are queries, keys and values. To analyze the diffusion features of an image or video frame, we first perform DDIM Inversion [37] to get its initial noise. Then, we input the initial noise to the UNet to denoise and get the features.

3.1.1. Shallow Features from the UNet Decoder

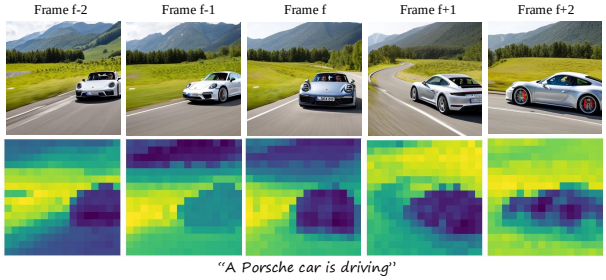
To understand what shallow features [1] from the UNet decoder capture, we perform a text-based image or video frame editing experiment. A general approach [29] to achieving this is to change the original text prompt to a new text prompt, and the UNet starts the editing denoising process from the DDIM inverted noise of the original image. In this experiment, we overwrite the shallow features with the corresponding features from the original video frame. This overwriting operation is called feature injection [41], detailed below.

Feature Injection. Features of the editing denoising UNet can be overwritten with the corresponding features from the source video frames or reference images at the same UNet layer. Fig. 3 shows the injection operation in the self-attention layer for different features. Fig. 3 (a) shows the features in DDIM reconstruction of the source image using the text prompt describing the source image. Fig. 3 (b) shows the features in editing the source image in Fig. 3 (a) based on a new text prompt, and thus the features are

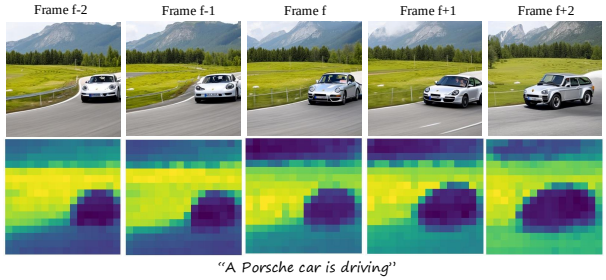
[1] Note that the shallow and deep features mentioned throughout this paper are extracted from the SD UNet decoder, and they are consistent with the those introduced in [41].



(a) Source Video.



(b) Synthesized (no injection).



(c) Synthesized (with injection).

Figure 4. We conduct a text-based video frame editing experiment to demonstrate that shallow features from the UNet decoder benefit object motion transfer. (a) shows source video frames, (b) shows frames edited using the prompt "a Porsche car is driving", and (c) shows frames generated using the same prompt in (b) but with shallow features injected from the source video. In (b), the generated car appears in various orientations that differ significantly from the source video. However, in (c), the synthesized car's position and pose closely match those of the source video, demonstrating that shallow features effectively preserve object motion.

changed accordingly. Fig. 3 (c) shows the features during text-based image editing in Fig. 3 (b) but with $\{\hat{\mathbf{K}}, \hat{\mathbf{V}}\}$ overwritten by $\{\mathbf{K}, \mathbf{V}\}$ of the source image. Fig. 3 (d) shows the input feature $\{\hat{\mathbf{f}}\}$ is overwritten by $\{\mathbf{f}\}$ of the source image. More formally, we define overwriting the keys $\hat{\mathbf{K}}$ and values $\hat{\mathbf{V}}$ with $\mathbf{K}^S$ and $\mathbf{V}^S$ from the source video in layer l as

$$\psi^S(\hat{\mathbf{K}}, \hat{\mathbf{V}}; l) := \{\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\} \rightarrow \{\hat{\mathbf{Q}}, \mathbf{K}^S, \mathbf{V}^S\}. \quad (3)$$

In this experiment, we use text prompts to edit a video. The source video frames are shown in Fig. 4 (a). The corresponding text prompt is "a car is driving". In Fig. 4 (b) we use the text prompt "a Porsche car is driving" and start from the DDIM inverted noise of the source video frames to edit the source video frames. We observe that the car appears in various orientations that differ significantly from the source video. In Fig. 4 (c), we use the same text prompt as used in (b) but with shallow features injected from the source video. We observe that the position and pose of the synthesized car correspond very well with the car in the source video, indicating that the shallow features correspond to low-frequency high-level features and preserve the motion of the object. Note that a similar analysis of the low-frequency high-level features has been conducted in [41]. However, the role of these features in video frame sequences during editing remains unexplored.

3.1.2. Deep Features from the UNet Decoder

Deep features from the UNet decoder correspond to high-frequency low-level features. In Fig. 2, we observe that these features reveal detailed visual properties of the object, such as the texture of the tires, the shape of the bear, and the eyes of the dog. This analysis is consistent with the properties of the deep features discussed in [41]. We present different feature injection strategies in Table 5 in the Supplementary Material.

3.2. Proposed Method

Based on the properties of the shallow and deep diffusion features, we propose FreeMix that achieves the customized video editing in a zero-shot manner. The framework of FreeMix is illustrated in Fig. 5. In this figure, the Source, Target and Reference Branches are UNets of the Stable Diffusion model with the same pre-trained and frozen weights. In our framework, we first apply DDIM Inversion to each frame of the source video and each reference image to get the initial noise. The noisy latents of the source video and the reference images pass through the Source Branch and the Reference Branch, respectively. The Target Branch starts from the same DDIM inverted noise as the Source Branch. In the denoising steps, shallow features from the Source Branch, and deep features from the Reference Branch are injected into the corresponding layers in the Target Branch. Finally, the Target Branch produces a synthesized video. We will detail the motion transfer and appearance transfer in the following sub-sections.

3.2.1. Motion Transfer

Our study in the previous subsection demonstrates that incorporating low-frequency high-level features from shallow decoder layers effectively transfers structure from each frame of the source video. When viewed sequentially, this process can be interpreted as motion transfer. Therefore, we use the shallow features from the source video to achieve object motion transfer in our method. In Fig. 5, during the denoising steps, shallow features of each source video

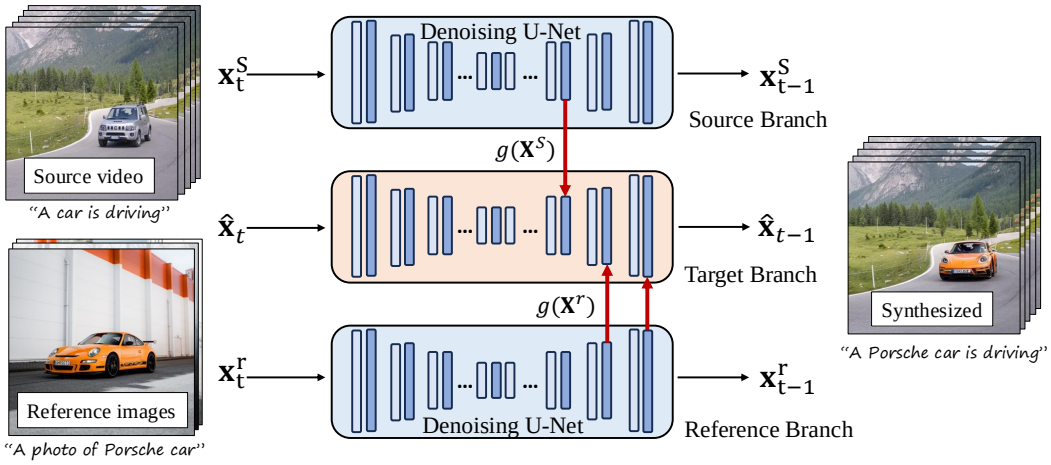


Figure 5. The framework of the proposed FreeMix. FreeMix requires text prompts describing the source video and the reference images, as well as a target text prompt where the source video's object name is replaced with that of the reference images. FreeMix enables zero-shot customized video editing by transferring diffusion features $g(\mathbf{X}^S)$ from the source video and $g(\mathbf{X}^r)$ from the reference images to the Target Branch, which produces the final synthesized video. During each denoising step, the noisy latent $\hat{\mathbf{x}}_t$ passes through the UNet to generate a denoised latent $\hat{\mathbf{x}}_{t-1}$. The process involves two key feature transfers in the UNet decoder: in the shallow layers, features from the source video $g(\mathbf{X}^S)$ overwrite the Target Branch features to transfer object motion, while in the deeper layers, features from the reference image $g(\mathbf{X}^r)$ overwrite the Target Branch features to transfer object appearance. Note that the denoising processes happen in the latent space and the VAE [20] encoder and decoder are omitted in this figure.

frame from the Source Branch are injected into the corresponding layer and frame of the Target Branch to transfer the motion of object in the source video.

3.2.2. Appearance Transfer

As analyzed earlier, high-frequency low-level diffusion features retain fine layout and preserve image details. Thus, we use these features from deep UNet decoder layers to transfer object appearance from reference images. Specifically, we manipulate diffusion features in the Target Branch in Fig. 5 by injecting features from the reference images. Unlike previous tuning-free personalization methods that use only one reference image, our approach supports multiple reference images, yielding improved video quality. To focus feature transfer on the object in the reference images and minimize background influence, we apply masks to the reference images.

Suppose we have N reference images, let $\mathbf{M}^{r_i}\mathbf{K}^{r_i}$ and $\mathbf{M}^{r_i}\mathbf{V}^{r_i}$ be the masked key and value, respectively, of the i-th reference image, for $i = 1, \cdots, N$. We denote by $\mathbf{K}^r$ the concatenation along the rows of the keys from the masked reference images, and by $\mathbf{V}^r$ the concatenation along the rows of the values from the masked reference images:

$$\mathbf{K}^r := [\cdots \mathbf{M}^{r_i}\mathbf{K}^{r_i} \cdots], \quad \mathbf{V}^r := [\cdots \mathbf{M}^{r_i}\mathbf{V}^{r_i} \cdots], \quad (4)$$

for $i = 1, \cdots, N$. We define overwriting the keys and values with $\{\mathbf{K}^r, \mathbf{V}^r\}$ of the reference images as

$$\psi^r(\hat{\mathbf{K}}, \hat{\mathbf{V}}; l) := \{\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\} \rightarrow \{\hat{\mathbf{Q}}, \mathbf{K}^r, \mathbf{V}^r\}. \quad (5)$$

The attention layer after feature injection from the reference images is computed as:

$$\text{Softmax}\left(\frac{\hat{\mathbf{Q}}(\mathbf{K}^r)^\intercal}{\sqrt{d}}\right)\mathbf{V}^r. \quad (6)$$

3.2.3. Enhancing Temporal Consistency via OT

Injecting shallow UNet decoder features effectively transfers motion from the source video but does not guarantee temporal smoothness in the synthesized video. This is because injecting shallow features does not ensure smooth feature transitions in the deep UNet decoder layers across consecutive frames, which more directly affect the final synthesized video. To address this, we apply regularization to the deep decoder layer features across frames to enhance temporal smoothness. A natural approach is to use feature correspondences from the deep decoder layers of consecutive frames in the source video to enforce similar correspondences in the Target Branch during synthesis. TokenFlow [11] introduces the Nearest Neighbor (NN) feature field between frames to find such correspondences. However, NN field cannot guarantee the global matching between two sets of features.

To overcome this, we apply Optimal Transport (OT) [5, 43] to the deep features of the source video frames to obtain feature correspondences. OT is a powerful distribution matching technique that can find the global correspondences between two sets of points, and has been shown ef-

fective in finding semantic correspondences between two images [27].

Let $\mathbf{O} \in \mathbb{R}^{h \times w \times d}$ be the deep diffusion features, and $\mathbf{M}^S \in \mathbb{R}^{h \times w}$ be the corresponding downsampled object mask of a source video frame, where $h \times w$ is the spatial dimension and d is the feature dimension. For each position coordinate (y, x) within the mask, we construct a new vector by concatenating (y, x) and the feature at (y, x), *i.e.*, $\mathbf{p} = [y, x, \lambda \mathbf{O}(y, x)] \in \mathbb{R}^{d+2}$, where λ balances the importance between the coordinate and the feature. We denote by $\mathbf{P}_f$ the collection all the constructed vectors within the mask for frame f. To find the feature correspondences between frame $f-1$ and f, we first compute the transport cost matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$, between each pair of vectors from $\mathbf{P}_{f-1}$ and $\mathbf{P}_f$, using the l_2-norm distance, where m and n are numbers of vectors in $\mathbf{P}_{f-1}$ and $\mathbf{P}_f$, respectively. Then we solve the OT problem:

$$\texttt{OT}(\mathbf{P}_{f-1}, \mathbf{P}_f) = \min_{\gamma \in \Gamma} \langle \gamma, \mathbf{C} \rangle + \text{reg} \cdot \Omega(\gamma), \tag{7}$$

where $\Gamma = \{\gamma \in \mathbb{R}_+^{m \times n} | \gamma \mathbf{1}_n = 1/m \cdot \mathbf{1}_m, \gamma^\intercal \mathbf{1}_m = 1/n \cdot \mathbf{1}_n\}$, $\mathbf{1}$ denotes the vector of all ones, γ is the transport plan matrix, with γ_{ij} representing the probability of vector i in $\mathbf{P}_{f-1}$ matching to vector j in $\mathbf{P}_f$, and Ω is the entropic regularization term. We denote by $\overrightarrow{\gamma}_f$ the solution to $\texttt{OT}(\mathbf{P}_{f-1}, \mathbf{P}_f)$, and similarly, $\overleftarrow{\gamma}_f$ the solution to $\texttt{OT}(\mathbf{P}_{f+1}, \mathbf{P}_f)$. $\overrightarrow{\gamma}_f$ represents the feature correspondences from frame $f-1$ to frame f in the source video, and $\overleftarrow{\gamma}_f$ represents the feature correspondences from frame $f+1$ to frame f. We apply these feature correspondences in generating the synthesized video:

$$\begin{aligned}\hat{\mathbf{O}}_f = &\frac{1}{2} \overrightarrow{\gamma}_f(\mathbf{M}^S_{f-1} \hat{\mathbf{O}}_{f-1}) + \frac{1}{2} \overleftarrow{\gamma}_f(\mathbf{M}^S_{f+1} \hat{\mathbf{O}}_{f+1}) \\ &+ (1 - \mathbf{M}^S_f) \mathbf{O}^S_f\end{aligned} \tag{8}$$

where $\mathbf{M}^S_{f-1}$, $\mathbf{M}^S_f$, and $\mathbf{M}^S_{f+1}$ are downsampled masks of the source video frames $f-1$, f and $f+1$, respectively, $\mathbf{O}^S_f$ are source video features for frame f, $\hat{\mathbf{O}}_{f-1}$ and $\hat{\mathbf{O}}_{f+1}$ are features for frames $f-1$ and f from the Target Branch in Fig. 5, $\overrightarrow{\gamma}_f(\cdot)$, $\overleftarrow{\gamma}_f(\cdot)$ denote the warping operations by taking the `argmax` index of each column of $\overrightarrow{\gamma}_f$ and $\overleftarrow{\gamma}_f$, respectively. The first two terms in Eq. 8 suggest we use the foreground feature correspondences from the source video to smooth frame f in generating the synthesized video. We considered the feature smoothing from both forward and backward directions. The last term in Eq. 8 means that we use the background features from the source video. We perform this smoothing operation to the self-attention output of the last layer of the UNet decoder, and denote the operation by $g(\hat{\mathbf{X}}) \leftarrow \mathcal{T}(g(\hat{\mathbf{X}}))$.

The whole process of our proposed method FreeMix is summarized in Algorithm 1.

Algorithm 1 Zero-shot Customized Video Editing

Input:
Stable Diffusion encoder $\mathcal{E}$, UNet ϵ_θ, and decoder $\mathcal{D}$
Source video $\mathbf{X}^S$ with F frames and its text prompt $\mathbb{T}^S$
Reference images $\mathbf{X}^r$ and text prompt $\mathbb{T}^r$
Target text prompt $\mathbb{T}$
Layers for source frame injection L^S
Layers for reference image injection L^r
The stop time step for the feature injection $t_{\texttt{max}}$
Output: Synthesized video $\hat{\mathbf{X}}$

$\mathbf{x}_0^S, \mathbf{x}_0^r \leftarrow \mathcal{E}(\mathbf{X}^S), \mathcal{E}(\mathbf{X}^r)$
$\{\mathbf{x}_t^S\}_{t=1}^T, \{\mathbf{x}_t^r\}_{t=1}^T \leftarrow$ DDIM-inv$(\mathbf{x}_0^S)$, DDIM-inv$(\mathbf{x}_0^r)$
$\hat{\mathbf{x}}_T \leftarrow \mathbf{x}_T^S$ ▷ copy source latents
for $t = T \cdots 1$ **do**
  ▷ denoising of the source and reference latents
  $\epsilon_{t-1}^S, g(\mathbf{X}^S) \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^S, \mathbb{T}^S, t)$
  $\epsilon_{t-1}^r, g(\mathbf{X}^r) \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^r, \mathbb{T}^r, t)$
  if $t > t_{\texttt{max}}$ **then**
    $\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, \mathbb{T}, t)$ **forwarding**
      $\psi^S(g(\hat{\mathbf{X}}); l)$ **for** $l \in L^S$ ▷ inject features
      $\psi^r(g(\hat{\mathbf{X}}); l)$ **for** $l \in L^r$ ▷ inject features
      $g(\hat{\mathbf{X}}) \leftarrow \mathcal{T}(g(\hat{\mathbf{X}}))$ ▷ temporal smoothing
    output $\hat{\epsilon}_{t-1}$
  else
    $\hat{\epsilon}_{t-1} \leftarrow \boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, \mathbb{T}, t)$
  end if
  $\mathbf{x}_{t-1}^S \leftarrow$ DDIM-samp$(\mathbf{x}_t^S, \epsilon_{t-1}^S)$
  $\mathbf{x}_{t-1}^r \leftarrow$ DDIM-samp$(\mathbf{x}_t^r, \epsilon_{t-1}^r)$
  $\hat{\mathbf{x}}_{t-1} \leftarrow$ DDIM-samp$(\hat{\mathbf{x}}_t, \hat{\epsilon}_{t-1})$
end for
$\hat{\mathbf{X}} \leftarrow \mathcal{D}(\hat{\mathbf{x}}_0)$

4. Experiments

We compare our method against the SOTA customized video editing methods: VideoSwap [13], and MotionDirector [54]. We also compare with text-based video editing methods: FateZero [32], TokenFlow [11], ControlVideo [52], and CCEdit [9], in terms of text alignment and temporal consistency. For the comparison with VideoSwap and MotionDirector, we conduct experiments on the 30 videos used by VideoSwap. To evaluate our approach against other methods, we use 15 representative videos from the DAVIS dataset [31] commonly used by text-to-video generation methods.

4.1. Qualitative Results

The qualitative comparisons of our method with baseline methods are shown in Figs. 6 and 7. Compared with text-based video editing methods in Fig. 6, FreeMix produces videos with clearer car appearances, more precise vehicle shapes, higher quality backgrounds, and car poses that more

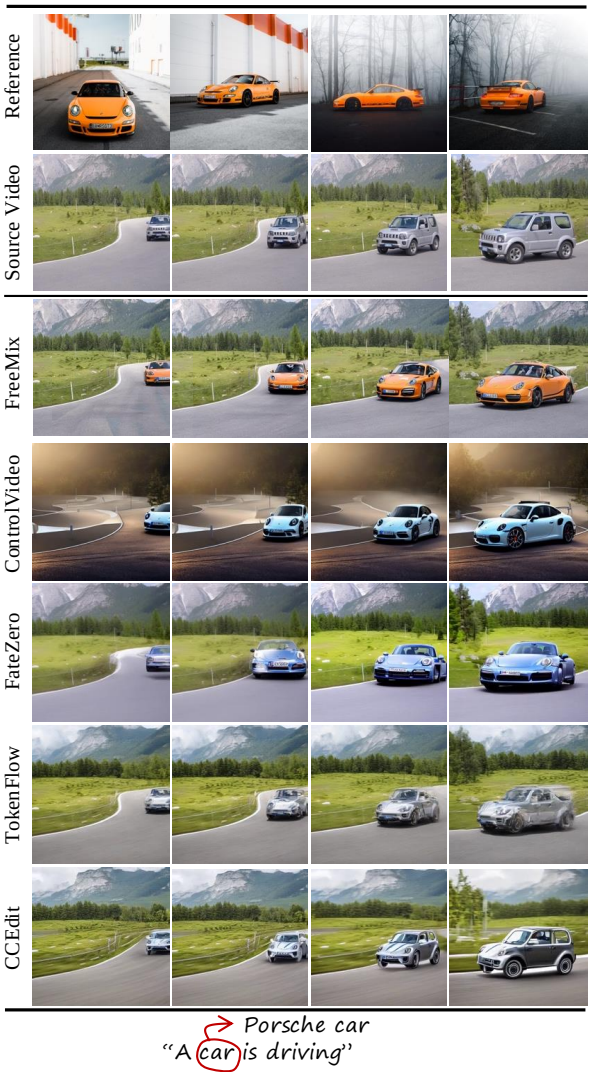


Figure 6. Qualitative comparisons among different methods.

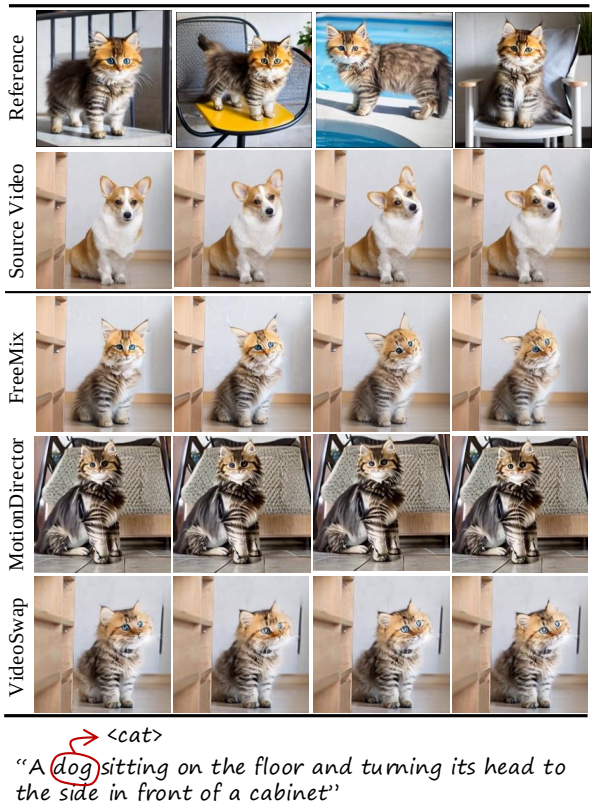


Figure 7. Qualitative comparisons among different customized video editing methods.

Table 1. Comparison among customized video editing methods.

	Concept Alignment	Text Alignment	Temporal Consistency
VideoSwap [13]	79.87	26.87	95.93
MotionDirector [54]	73.94	32.14	97.49
FreeMix (ours)	**80.34**	**32.19**	**97.53**

Table 2. Comparison with text-based video editing methods.

	Text Alignment	Temporal Consistency
FateZero [32]	30.39	92.77
TokenFlow [11]	30.38	95.51
ControlVideo [52]	30.76	95.13
CCEdit [9]	27.68	94.25
FreeMix (ours)	**30.98**	**95.71**

accurately align with the source video frames. In Fig. 7, MotionDirector fails to transfer the cat's head movements from the source video. While VideoSwap generates the cat, it generates the cat's face much larger than the cat's face in the reference images and shows subtle head movements that deviate significantly from the source video frames. In contrast, FreeMix generates a cat that faithfully reproduces both the appearance from the reference images and the head movements from the source video frames.

Figure 8 provides diverse synthesized videos produced by FreeMix. When provided with reference images, FreeMix effectively captures the appearance from the object, as well as the object's motion from the source video, as shown in Fig. 8 (left). Additionally, FreeMix can seamlessly incorporate pre-trained concept weights, the ED-LoRA weights from VideoSwap. By integrating these weights, FreeMix generates a video with the encoded concept while preserving the motion from the source video, as shown in Fig. 8 (right). More qualitative results and videos can be found in the Supplementary Material.

4.2. Quantitative Results

We measure the text alignment and the temporal consistency of different methods, as well as the concept alignment for customized video editing methods. Please refer to the Supplementary Material for more details of each metric.

Table 1 presents a comparison between our method

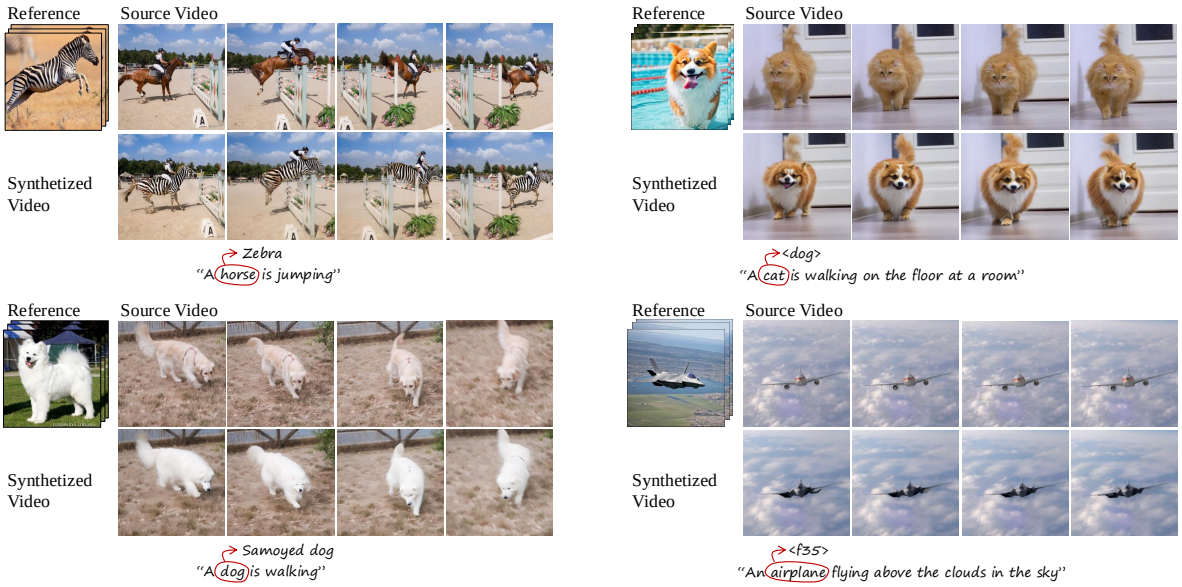


Figure 8. Sample results of FreeMix. (Left) By solely transferring diffusion features, FreeMix effectively transfers both appearance and motion. (Right) FreeMix seamlessly leverages pre-trained concept weights from VideoSwap [13] to produce videos with encoded concepts, while preserving the motion from the source video.

FreeMix and the customized video editing methods VideoSwap and MotionDirector. Both require fine-tuning on each source video and each customized object. Although FreeMix does not require any fine-tuning, it is better than VideoSwap and MotionDirector in all three metrics.

Table 2 compares our method with recent text-based video editing methods, showing the superiority of our method in both text alignment and temporal consistency. Note that these methods are designed for general text-to-video generation, with specific modules to improve temporal consistency, rather than for customized video editing. Nonetheless, FreeMix achieves better temporal consistency, demonstrating the effectiveness of our motion transfer and OT techniques in enhancing video coherence.

Effectiveness of Optimal Transport. We evaluate the effectiveness of OT in FreeMix for improving temporal smoothness on the DAVIS dataset. The results are provided in Table 3. As shown in the table, without OT in FreeMix, the temporal consistency score is 93.76. By incorporating OT, FreeMix improves the temporal consistency of the synthesized videos by at least 1.5 points for λ ranging from 0 to 0.1. This demonstrates the effectiveness of OT in improving temporal smoothness. When $\lambda = 0.001$, FreeMix achieves the highest temporal consistency score of 95.71, suggesting the necessity of including both coordinate and feature information when computing feature correspondences.

Moreover, we evaluate the effectiveness of OT by replacing it in FreeMix with an alternative regularization method: the Nearest Neighbor (NN) approach from TokenFlow [11]. This replacement results in a decrease in the temporal consistency score from 95.71 to 94.88. Furthermore, we assess the effectiveness of OT by incorporating it into an existing image editing framework, PnP [41], for video editing. Similar to FreeMix, we integrate OT into the final layer of the UNet in PnP across all frames. We compare with PnP that edits each video frame independently. The application of OT ($\lambda = 0.001$) significantly improves the temporal consistency from 93.44 to 95.63, compared to the version without OT.

Table 3. Ablation study of without (w/o) OT or with (w/) OT at different λ values for our method FreeMix in terms of temporal consistency.

w/o OT	w/ OT, λ =	0.1	0.01	0.001	0.0001	0
93.76		95.29	95.47	**95.71**	95.59	95.59

5. Conclusion

This paper presents a zero-shot customized video editing method using diffusion feature transfer. Through analyzing diffusion model features, we find that injecting features from specific layers effectively transfers motion from the source video and appearance from reference images, enabling customized editing. To enhance temporal consistency, we design OT to align feature correspondences. We demonstrate the effectiveness of our approach through both qualitative and quantitative results. Our method is built upon UNet, which remains the dominant architecture in video editing due to computational constraints. Extending this approach to DiT will be explored in future work.

References

[1] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *CVPR*, pages 22560–22570, 2023. 2

[2] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *ICCV*, pages 23040–23050, 2023. 2

[3] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *CVPR*, pages 6593–6602, 2024. 2

[4] Yuren Cong, Mengmeng Xu, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, Sen He, et al. Flatten: optical flow-guided attention for consistent text-to-video editing. In *ICLR*, 2024. 3

[5] M Cuturi. Lightspeed computation of optimal transportation distances. *NeurIPS*, pages 2292–2300, 2013. 5

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, pages 8780–8794, 2021. 2

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 2

[8] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, pages 7346–7356, 2023. 2

[9] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. In *CVPR*, pages 6712–6722, 2024. 3, 6, 7

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[11] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024. 2, 3, 5, 6, 7, 8

[12] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[13] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. Videoswap: Customized video subject swapping with interactive semantic point correspondence. In *CVPR*, pages 7621–7630, 2024. 2, 6, 7, 8

[14] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 2

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, 2023. 2

[16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *EMNLP*, 2021. 1

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, pages 6840–6851, 2020. 2

[18] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, pages 8633–8646, 2022. 2

[19] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *ICLR*, 2023. 2

[20] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 5

[21] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. 2023. 2

[22] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. 2023. 2, 1

[23] Min Seok Lee, WooSeok Shin, and Sung Won Han. Tracer: Extreme attention guided salient object tracing network (student abstract). In *AAAI*, pages 12993–12994, 2022. 1

[24] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *CVPR*, pages 8207–8216, 2024. 2

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 1

[26] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *CVPR*, pages 8599–8608, 2024. 2

[27] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4463–4472, 2020. 6

[28] Haoyu Ma, Shahin Mahdizadehaghdam, Bichen Wu, Zhipeng Fan, Yuchao Gu, Wenliang Zhao, Lior Shapira, and Xiaohui Xie. Maskint: Video editing via interpolative non-autoregressive masked transformers. In *CVPR*, pages 7403–7412, 2024. 3

[29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 3

[30] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. In *CVPR*, pages 8089–8099, 2024. 2

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

[32] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *ICCV*, pages 15932–15942, 2023. 2, 6, 7

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 1

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[35] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 2

[36] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. pages 2256–2265, 2015. 2

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 2, 3

[38] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE transactions on image processing*, pages 3998–4011, 2018. 2

[39] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10748–10757, 2022. 2

[40] Narek Tumanyan, Omer Bar-Tal, Shir Amir, Shai Bagon, and Tali Dekel. Disentangling structure and appearance in vit feature space. *ACM Transactions on Graphics*, 43(1):1–16, 2023. 2

[41] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 2, 3, 4, 8

[42] Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 3

[43] Cédric Villani et al. *Optimal transport: old and new*. Springer, 2009. 5

[44] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. p+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 2

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, pages 600–612, 2004. 2

[46] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pages 7623–7633, 2023. 2

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

[48] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *CVPR*, pages 8466–8476, 2024. 2

[49] Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6786–6795, 2024. 2

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, pages 3836–3847, 2023. 3

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[52] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, XIAOPENG ZHANG, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. In *ICLR*, 2024. 2, 3, 6, 7

[53] Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. Controlvideo: Conditional control for one-shot text-driven video editing and beyond. *arXiv preprint arXiv:2305.17098*, 2023. 3

[54] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. *ECCV*, 2024. 2, 6, 7

Zero-shot Customized Video Editing with Diffusion Feature Transfer

Supplementary Material

In this supplementary material, we provide more experimental setting details, and results.

6. Experimental Settings

Diffusion Feature Visualization. In Section 3, to visualize those high-dimensional SD features and understand how semantic spatial information is internally encoded within diffusion features, we use Principal Component Analysis (PCA) to interpret the dominant visual properties within these high-dimensional features. Specifically, we select the denoising step $t = 401$ and extract features $\mathbf{O}$ from each block of the UNet decoder. We represent these features using the first three leading components from PCA, and visualize them as RGB images.

Diffusion Feature Injection. The Target Branch in Fig. 5 starts from the same DDIM inverted noise as the Source Branch. In the denoising steps, the shallow features from the Shallow Block of the Source Branch, and the deep features from the Intermediate Block and Deep Block of the Reference Branch are injected into the Target Branch. We inject features from both branches during the first 60% denoising steps, *i.e.*, $t_{\texttt{max}} = (1 - 0.6)T$, where T is the number of total denoising time steps. Finally, the Target Branch produces a synthesized video.

Evaluation Metrics. We employ the adapted CLIP-Score [16] to evaluate text alignment and temporal consistency. Specifically, we calculate the CLIP similarity between each frame of the synthesized video and the target text prompt, then report the average CLIP similarity across all frames. For temporal consistency, we assess the CLIP similarity between consecutive frames of the synthesized video, and calculate the average value. To evaluate concept alignment, we follow Custom Diffusion [22] to measure the CLIP similarity between each frame of the synthesized video and all reference images, and report the average score across all video frames.

Details of Masks. In our experiments, the DAVIS dataset contains masks for each video. For videos without masks, we use the TRACER [23] to extract masks. In practice, any off-the-shelf segmentation tool, such as the Grounded-SAM-2 [25, 33], can be used to produce image and video masks.

Table 4. Human preferences.

Win Rate	Motion fidelity	Temporal consistency
Ours vs. MotionDirector	**93%** vs. 7%	**69%** vs. 31%
Ours vs. VideoSwap	**56%** vs. 44%	30% vs. **70%**

7. Experiment Results

7.1. Human Preferences

We provide human evaluation results on synthesized videos in Tab. 4. Specifically, we visually evaluate the temporal consistency and motion fidelity of the videos generated by different methods, where motion fidelity means whether the motion of object in the synthesized video is the same as that of the object in the source video. Tab. 4 presents the win rates of our FreeMix against VideoSwap and MotionDirector. The results demonstrate that our method outperforms both approaches in motion fidelity and exhibits higher temporal consistency compared to MotionDirector.

7.2. Ablation Study

Different Feature Injection Combinations. We present various feature injection strategies in Tab. 5. Specifically, we evaluate various injection strategies by testing different combinations of injecting source video or reference image features into shallow or deep UNet blocks: (1) Injecting features from the source video solely into either the Shallow Block or Deep Block; (2) Injecting features from the reference images solely into either the Shallow Block or Deep Block; (3) Injecting features from both the source video and reference images into the Shallow Block; (4) Injecting features from both the source video and reference images into the Deep Block; (5) Injecting features from the source video into the Deep Block, and injecting features from the reference images into the Shallow Block; (6) Injecting features from the source video into the Shallow Block, and injecting features from the reference images into the Deep Block. The results align with our observations: injecting source video features into shallow layers ensures good temporal consistency, while injecting both source video and reference image features into shallow layers compromises temporal consistency. Injecting reference image features into either shallow or deep layers achieves good concept alignment, but mixing source video and reference image features reduces concept alignment. Injecting source video features into shallow layers and reference image features into deeper layers, as configured in our paper, achieves both strong concept alignment and temporal consistency.

Table 5. Different feature injection combinations.

	Src(shallow)	Src(deep)	Ref(shallow)	Ref(deep)
Concept Alignment	71.44	71.48	75.17	75.88
Temporal Consistency	95.67	90.71	87.53	90.28
	Src(s) Ref(s)	Src(d) Ref(d)	Src(d) Ref(s)	Src(s) Ref(d)
Concept Alignment	73.43	71.8	70.66	**75.79**
Temporal Consistency	92.77	92.1	91.37	**95.71**

Table 6. Number of reference images.

Number of References	1	2	3	4	5
Concept Alignment	74.97	74.78	75.17	**75.92**	75.79
Temporal Consistency	93.76	94.35	95.11	95.44	**95.71**

Table 7. Influence of timesteps.

$t_{\max}$	$0.2T$	$0.4T$	$0.6T$	$0.8T$	T
Concept Alignment	74.29	75.6	**75.79**	75.73	74.23
Temporal Consistency	93.63	93.87	**95.71**	94.73	94.41

Table 8. GPU memory consumption and running time.

	FateZero	CCEdit	ControlVideo	VideoSwap	Ours
GPU (GB)	24	26	10	16	**8**
Time (s)	246	137	73	144	**50**

Number of Reference Images. Tab. 6 illustrates the impact of the number of reference images on temporal consistency and concept alignment. Using only 1-2 reference images results in lower scores, while using 4-5 reference images significantly improves both temporal consistency and concept alignment. Based on our experiments, we use 5 reference images.

Influence of Timesteps. We compare different values of $t_{\max}$, the stopping timestep for feature injection, ranging from $0.2T$ to T, where T denotes the total number of timesteps. The results are shown in Tab. 7. Based on our experiments, we select $t_{\max} = 0.6T$, as it achieves the best concept alignment and temporal consistency.

Computational Complexity. We present the GPU memory requirements and video generation times for different methods in Tab. 8. Among all methods, our approach requires the least memory and achieves the fastest speed. In our implementation, OT is computed once for each video and takes approximately 5 seconds.

Additional Evaluation Metrics. Tab. 9 extends the comparison with baseline methods using additional evaluation metrics, complementing the primary results presented in Tabs. 1 and 2. We evaluate image quality using three standard metrics. Peak Signal-to-Noise Ratio (PSNR) quantifies reconstruction fidelity based on the pixel-wise mean squared error. The Structural Similarity Index Measure (SSIM) [45] provides a perceptually-motivated assessment by comparing local patterns of luminance, contrast, and structure. Finally, the Learned Perceptual Image Patch Similarity (LPIPS) [51] leverages a pre-trained deep neural network to compute the distance between image patches in a feature space, which correlates closely with human perceptual judgment. We also use warp error to quantify the temporal consistency, which measures the geometric misalignment between a point's true position in a target image and its predicted position after being transformed from a source image by an estimated warp field. FreeMix outperforms baseline methods with better video quality and temporal consistency.

Table 9. Video quality and consistency with additional metrics. PSNR, SSIM, LPIPS are metrics for video equality, Warp error is the metric for temporal consistency.

	PSNR ↑	SSIM ↑	LPIPS (AlexNet) ↓	LPIPS (Vgg) ↓	Warp Error ↓
FateZero	15.88	0.58	0.38	0.42	0.08
FLATTEN	16.23	0.53	0.28	0.36	0.11
TokenFlow	18.46	0.6	0.33	0.39	0.05
ControlVideo	9.7	0.24	0.74	0.73	0.33
CCEdit	13.94	0.32	0.52	0.56	0.12
FreeMix (ours)	**19.87**	**0.68**	**0.19**	**0.28**	**0.03**

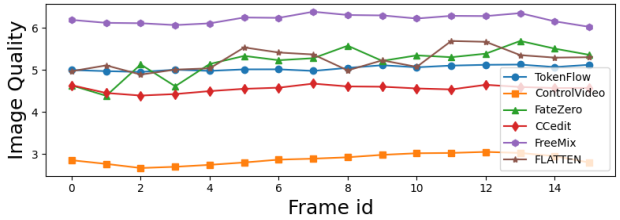


Figure 9. Image quality vs. each frame

Evaluate Image Quality Over Time. To assess temporal quality consistency, we evaluate each video frame using the pre-trained NIMA model [38], which predicts perceptual image quality. As illustrated in Fig. 9, the per-frame analysis demonstrates that FreeMix consistently outperforms all baseline methods in terms of overall image quality.

7.3. Qualitative Results

We show more qualitative results of FreeMix in Figs 10-13. As can be seen from those results, FreeMix successfully transfers the motion of the object from the source video, and the appearance of the object in the reference images, to the synthesized video. It is worth noting that FreeMix is flexible to deal with source video with different aspect ratios.

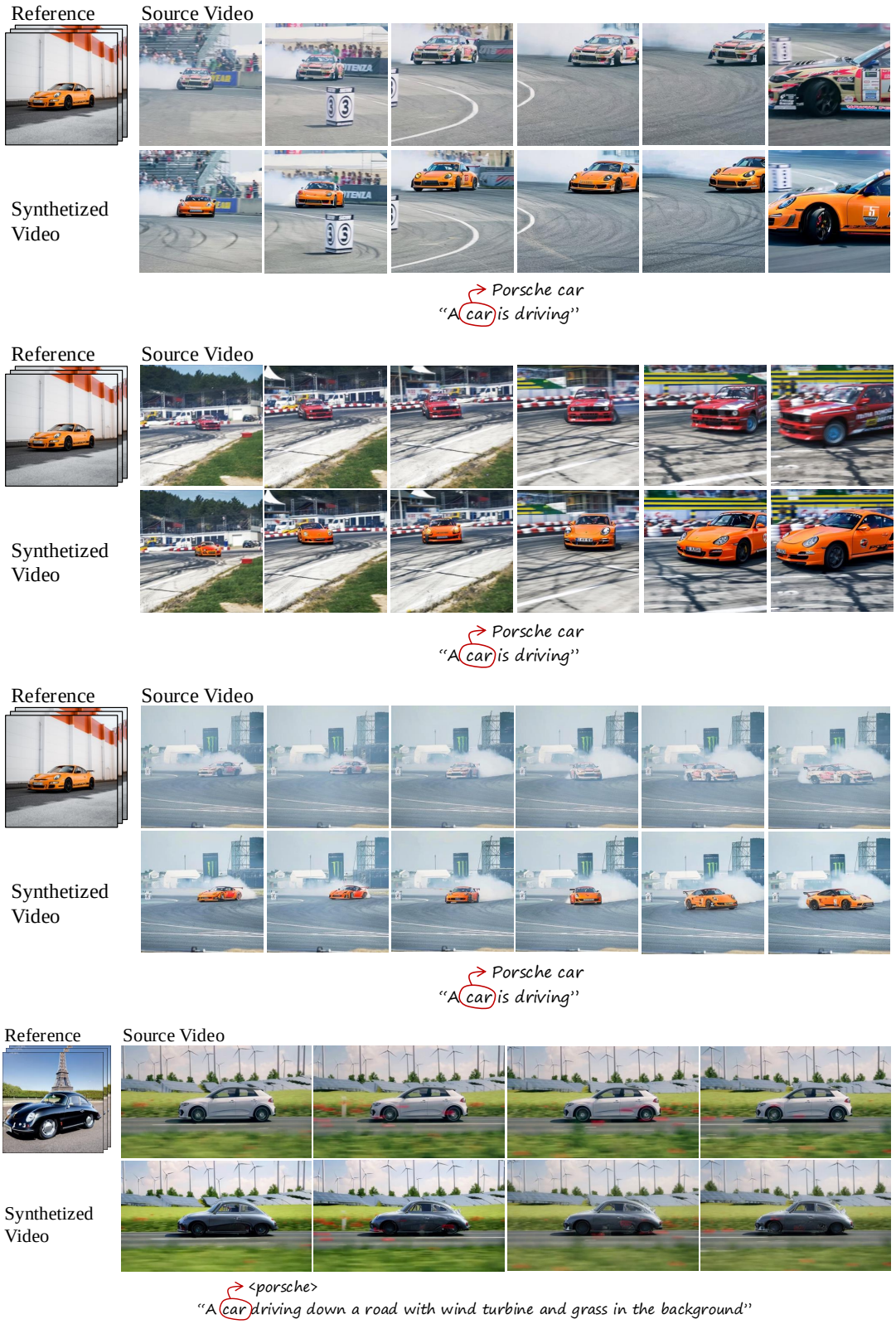


Figure 10. Sample results of FreeMix.

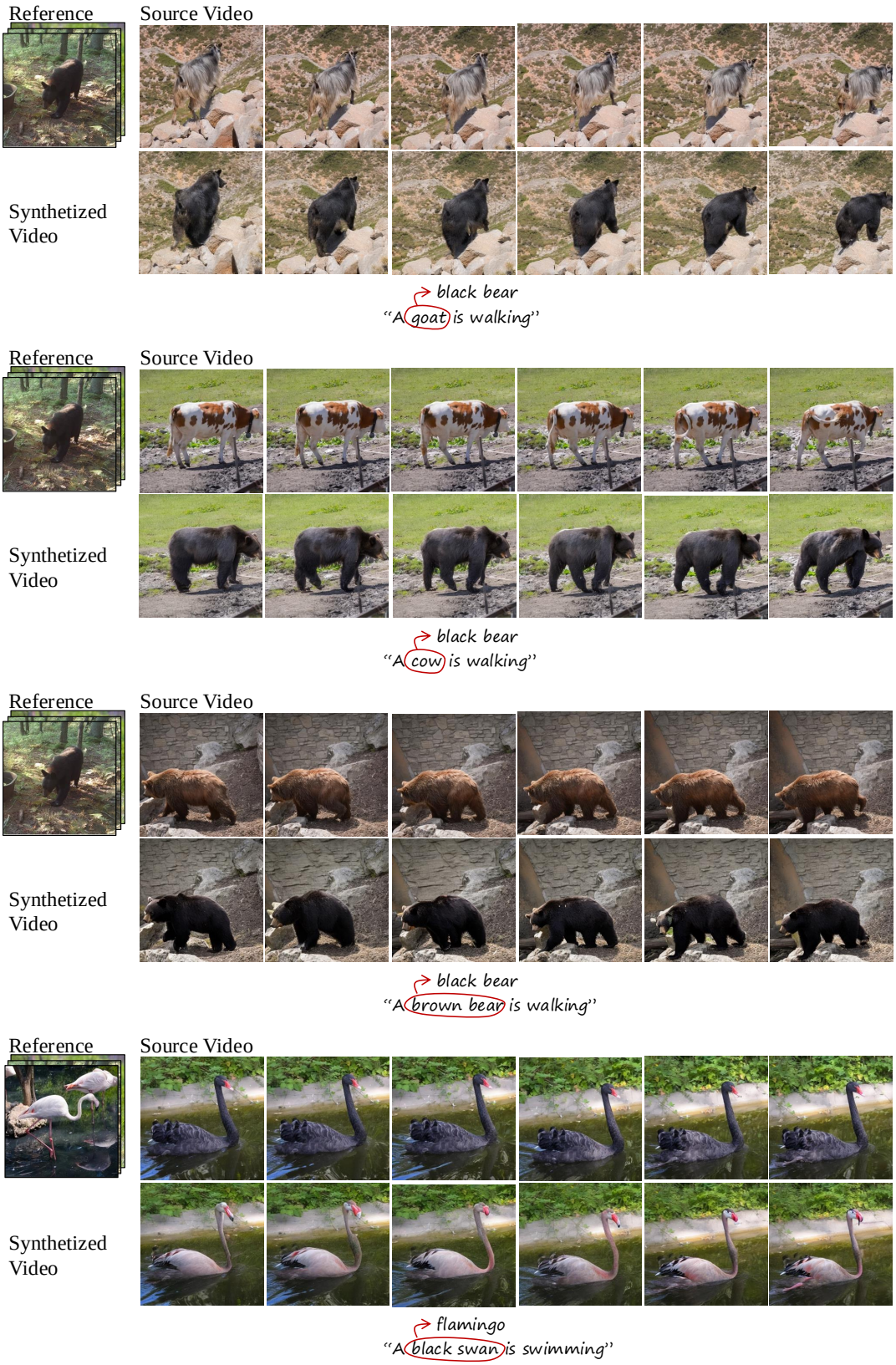


Figure 11. Sample results of FreeMix.

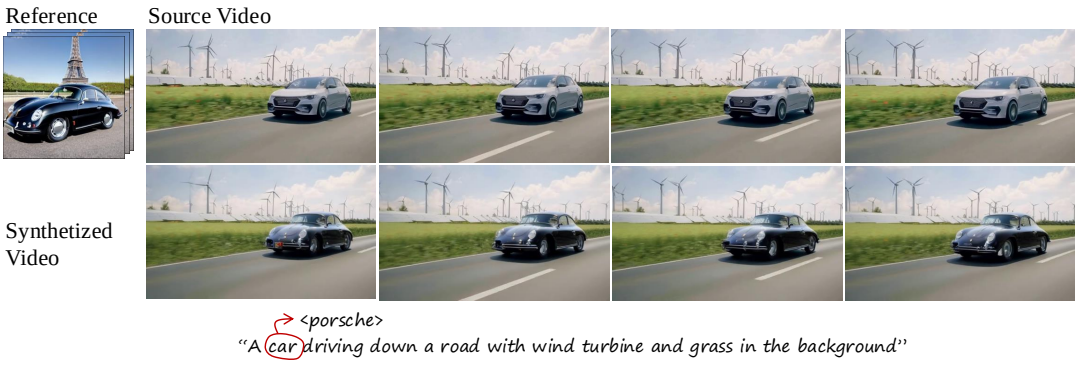


Figure 12. Sample results of FreeMix.

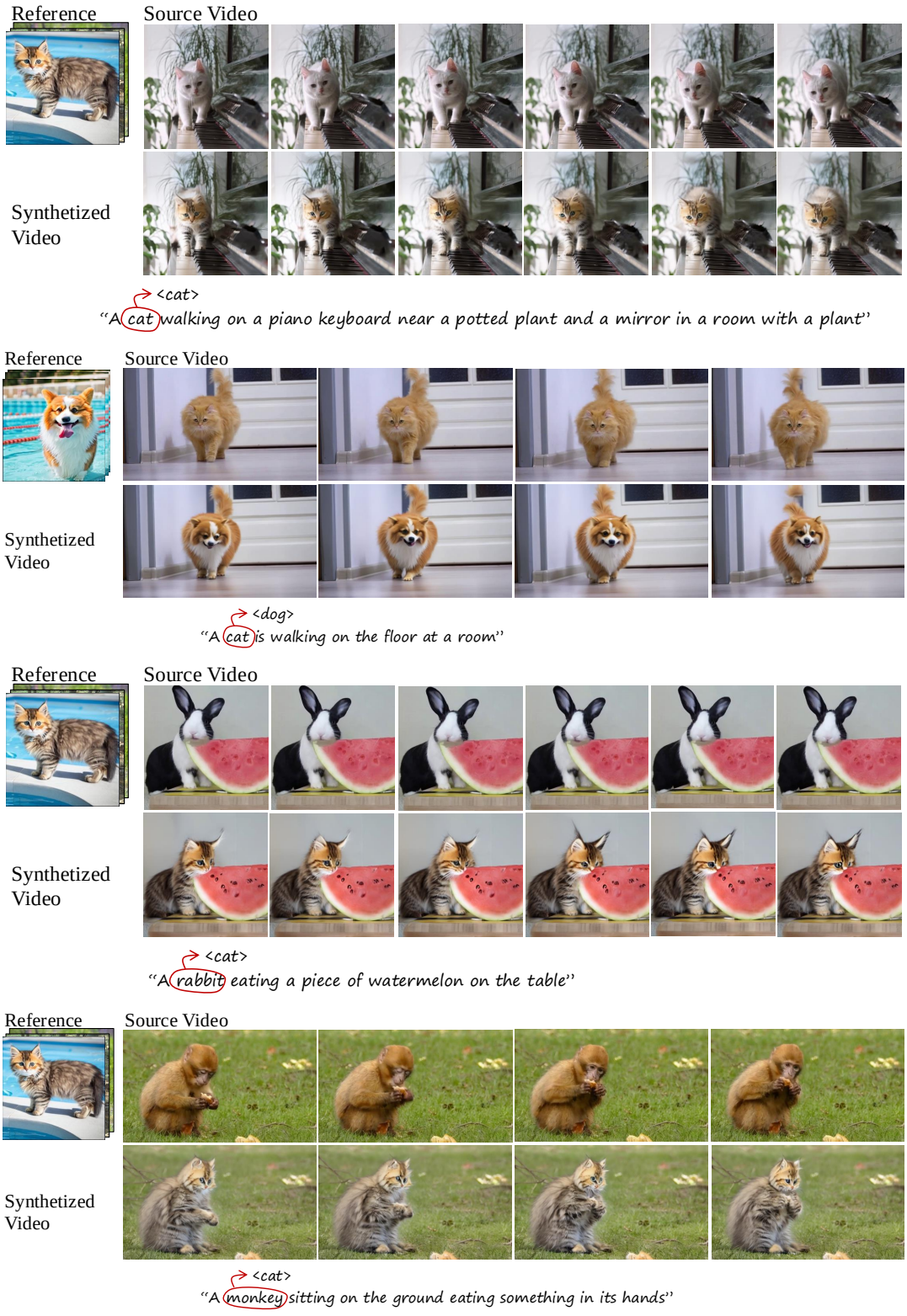


Figure 13. Sample results of FreeMix.