FRESCO: Spatial-Temporal Correspondence for Zero-Shot Video Translation

Shuai Yang[1*] Yifan Zhou[2] Ziwei Liu[2] Chen Change Loy[2] ✉

[1]Wangxuan Institute of Computer Technology, Peking University
[2]S-Lab, Nanyang Technological University

williamyang@pku.edu.cn {yifan006, ziwei.liu, ccloy}@ntu.edu.sg

Supplementary Material

Contents

* Work done when Shuai Yang was RAP at S-Lab, NTU.

1. Implementation Details

The experiment is conducted on one NVIDIA Tesla V100 GPU. By default, we set batch size $N \in [6, 8]$ based on the input video resolution, the loss weight $\lambda_{\text{spat}} = 50$, the scale factors $\lambda_s = \lambda_t = 5$. We use 20 steps of DDPM sampling, meaning the maximum value of the SDEdit [4] parameter T is 20. The spatial-guided attention is activated in the step T. The temporal-guided attention is used from step T to step 7. For feature optimization, we update $\mathbf{f}$ for $K = 20$ iterations with Adam optimizer and learning rate of 0.4, from step T to step 5. GMFlow [7] is used to estimate optical flows and occlusion masks. Background smoothing [3] is applied at the steps of 4 and 3 to improve temporal consistency in the background region.

In terms of the model design, the order of cross-frame and temporal attentions follows FLATTEN [1]. We experimentally found that inserting spatial attention in the beginning produce sharper results than after the cross-frame attention.

For translating video into cartoon styles, we use the customized model 'Flat-2D Animerge' (https://civitai.com/models/35960?modelVersionId=42138). For translating video into CG styles, we use the customized model 'ReV Animated' (https://civitai.com/models/7371?modelVersionId=19575). For translating video into photo-realistic styles, we use the customized model 'Realistic Vision' (https://civitai.com/models/4201?modelVersionId=29460).

The testing videos are mainly from Pexels (https://www.pexels.com/). Parts are from the project of Tune-A-Video [6] and the LOVEU-TGVE-2023 dataset (https://github.com/showlab/loveu-tgve-2023).

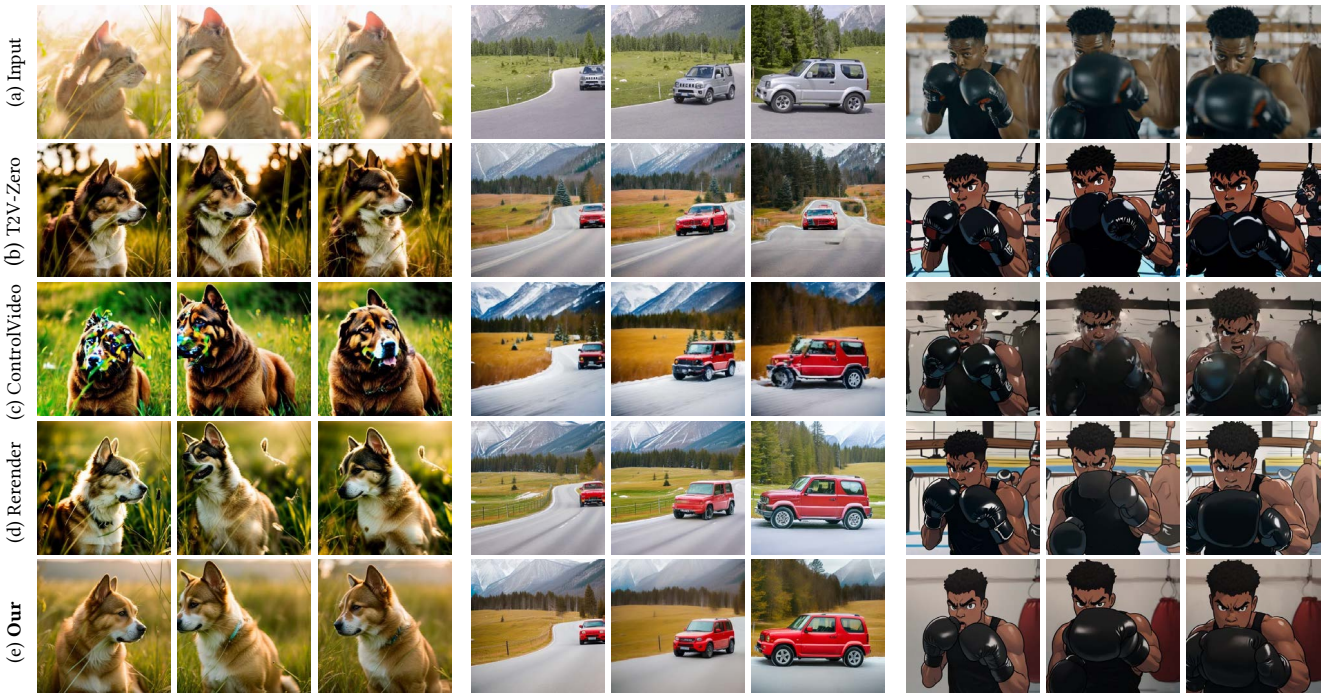
2. Running Time

Inferencing on a 512×512 video with one NVIDIA Tesla V100 GPU, a batch size of 8 and total sampling steps of $T = 15$ takes about 53.60 s per batch. Therefore, the running time per frame is about 6.70 s, where the preprocessing (edge extraction, optical flow estimation, single-step DDPM forward and backward process for self-similarity calculation) takes about 0.77 s and the DDPM sampling takes about 5.93 s. Since our method directly optimizes on the feature without calculating and back-propagating gradients in the U-Net, our feature optimization does not increase excessive time consumption and memory consumption. Specifically, optimization uses about extra 4.88s and 0.84 GB memory per frame.

3. Comparison with State-of-the-Art Methods

3.1. Qualitative comparison

We compare with three recent inversion-free zero-shot methods: Text2Video-Zero [3], ControlVideo [9], Rerender-A-Video [8] in Figs. 1-4. To ensure a fair comparison, all methods employ identical settings of ControlNet, SDEdit, and LoRA. As shown in Figs. 1-4, all methods successfully translate videos according to the provided text prompts. However, the inversion-free methods, relying on ControlNet conditions, may experience a decline in video editing quality if the conditions are of low quality, due to issues like defocus or motion blur. In contrast, our method can generate consistent videos based on the proposed robust FRESCO guidance.

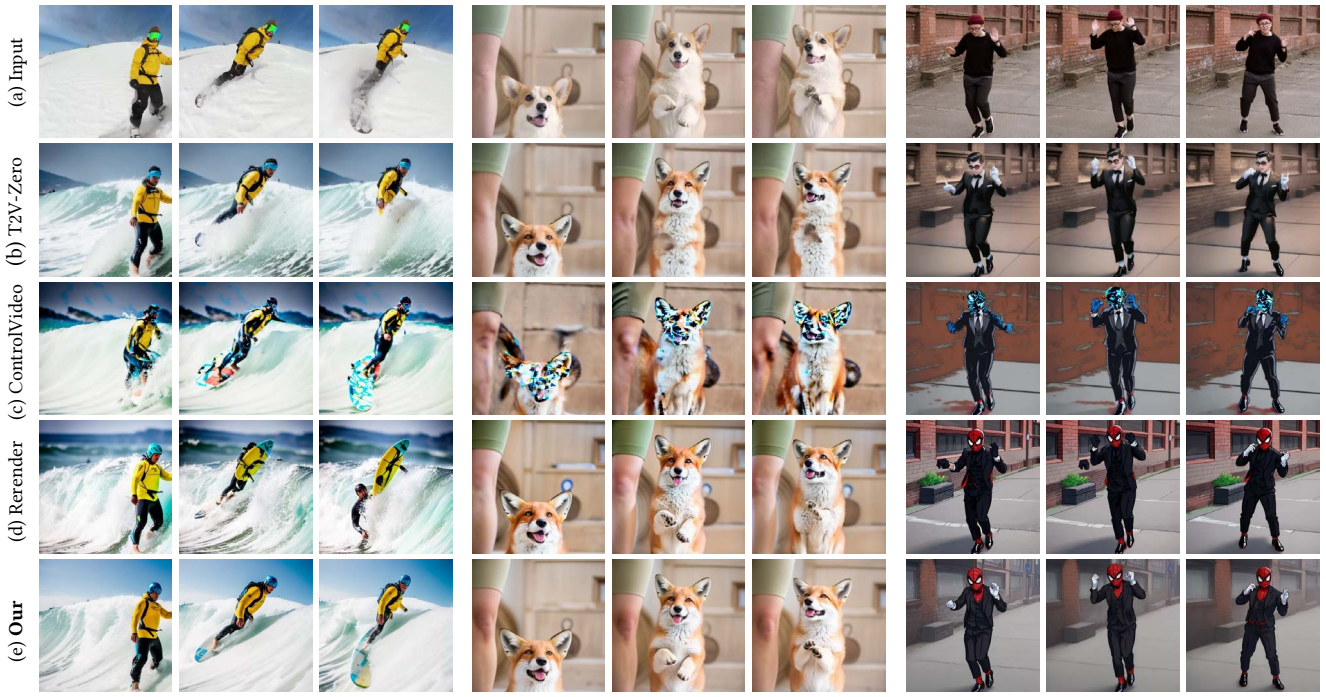


Prompt: *A dog in the grass in the sun*

Prompt: *A red car turns in the winter*

Prompt: *A black boxer wearing black boxing gloves punches towards the camera, cartoon style*

Figure 1. Visual comparison with inversion-free zero-shot video translation methods. (Part I)



Prompt: A *man surfing in the sea*

Prompt: *Greetings from a fox by shaking front paws*

Prompt: *A cartoon spiderman in black suit, black shoes and white gloves is dancing*

Figure 2. Visual comparison with inversion-free zero-shot video translation methods. (Part II)

Figure 3. Visual comparison with inversion-free zero-shot video translation methods. (Part III)

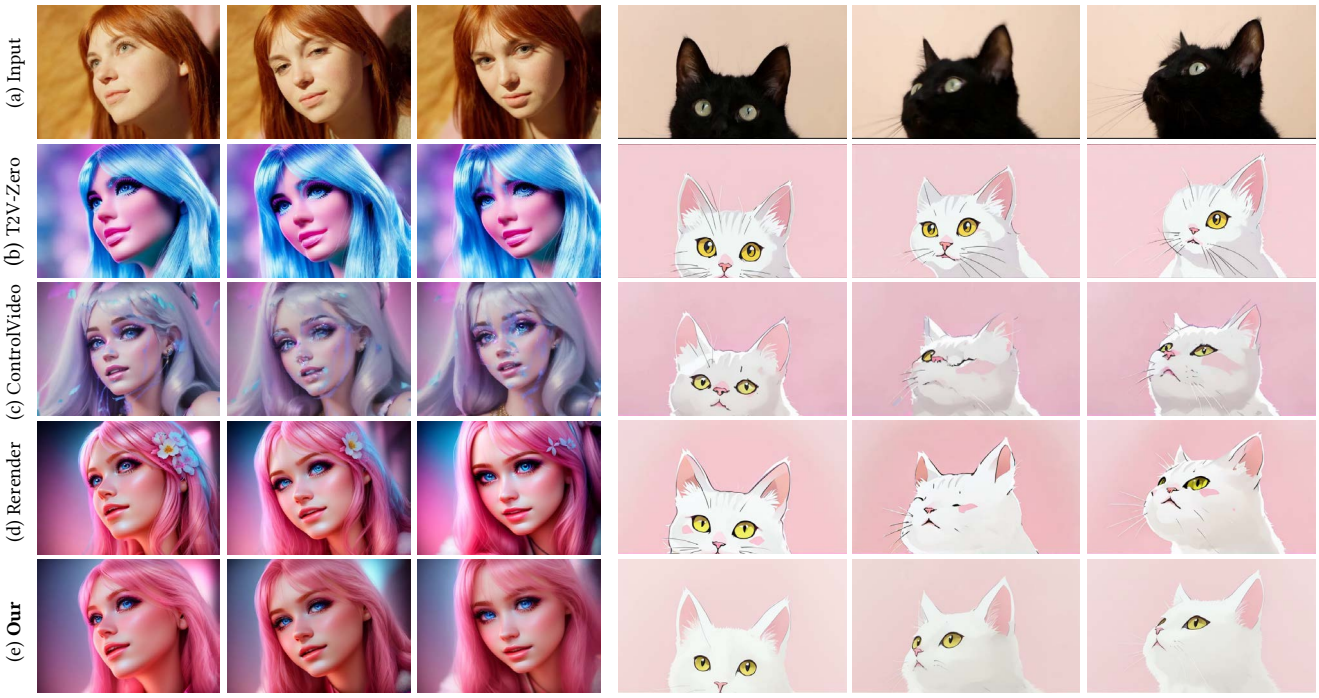


Figure 4. Visual comparison with inversion-free zero-shot video translation methods. (Part IV)

3.2. User study

We further conduct a user study with 57 participants. Participants are tasked with selecting the most preferable results among the four methods. Table 1 presents the preference rates of 11 test videos in Figs. 1-4 and the overall averaged preference rates, revealing that our method emerges as the most favored choice.

Table 1. User preference rates.

ID	Text2Video-Zero [3]	ControlVideo [9]	Rerender-A-Video [8]	Ours
01. A dog in the grass in the sun	1.8%	0.0%	33.3%	**64.9%**
02. A red car turns in the winter	0.0%	10.5%	12.3%	**77.2%**
03. A black boxer punches towards the camera	8.8%	0.0%	22.8%	**68.4%**
04. A man surfing in the sea	31.6%	3.5%	14.0%	**50.9%**
05. Greetings from a fox by shaking front paws	1.8%	0.0%	45.6%	**52.6%**
06. A cartoon spiderman in black suit is dancing	0.0%	0.0%	40.4%	**59.6%**
07. A white deer in the snow	10.5%	8.8%	14.0%	**66.7%**
08. A beautiful woman holding her glasses	3.5%	3.5%	24.6%	**68.4%**
09. Shower of sand in desert	31.6%	1.8%	8.8%	**57.8%**
10. A beautiful barbie girl	8.8%	0.0%	26.3%	**64.9%**
11. A white cat in pink background	1.8%	0.0%	14.0%	**84.2%**
Averaged	9.1%	2.6%	23.3%	**65.0%**

3.3. Compare with inversion-based zero-shot video translation method

Zero-shot video translation methods can be divided into inversion-based and inversion-free methods. Our method is inversion-free and we compare with three inversion-free methods in the main paper for a fair comparison.

In this section we compare with the latest inversion-based method TokenFlow [2]. Different pipelines are not highly comparable. This comparison is just to show the unique characteristics of these two pipelines and to demonstrate why we choose the inversion-free pipeline.

TokenFlow extends existing image editing methods to video domain by propagating diffusion features based on inter-frame correspondences. At the time of this submission, TokenFlow provides two versions based on two editing methods Plug-and-Play (pnp) [5] and SDEdit [4]. Figures 5-8 give the qualitative comparison with the two versions and Table 2 gives the quantitative comparison, with the same settings of SDEdit and LoRA as our method, if applicable.

It can be seen that compared to inversion-based methods, inversion-free methods allow for more flexible conditioning and higher compatibility with the customized models, enabling users to conveniently control the output appearance. For example, the diffusion feature fusion in TokenFlow sometimes makes the style less convincing (less cartoon-like in *boxer*, *spiderman*) and may destroy some detailed features (missing face in *boxer*, blurring face in *barbie*). Another problem for inversion-based method is that the inversion feature guidance sometimes can be too strong to make a fully translation, *e.g.*, failure translation to *spiderman* or *white cat*. The low Fram-Acc in Table 2 also verifies this problem. By comparison, inversion-free methods are more flexible to control the output appearance.

Compared to inversion-free methods, inversion-based methods can better reconstruct the original frame in the unedited region. It can be seen that the boxing gloves are well reconstructed in *boxer*. While the inversion feature guidance sometimes can be too strong to make valid translation, it can well improve the temporal consistency by accurately reconstructing the original frames, as indicated in the low Pixel-MSE score in Table 2. To take full advantage of editability of the diffusion model, we choose the inversion-free pipeline and propose FRESCO-based guidance to improve its temporal consistency.

Table 2. Quantitative comparison with inversion-based method.

Metric	Fram-Acc ↑	Tem-Con ↑	Pixel-MSE ↓
TokenFlow-pnp [2]	0.869	0.978	**0.007**
TokenFlow-SDEdit [2]	0.920	0.973	0.012
ours	**1.000**	**0.980**	0.012

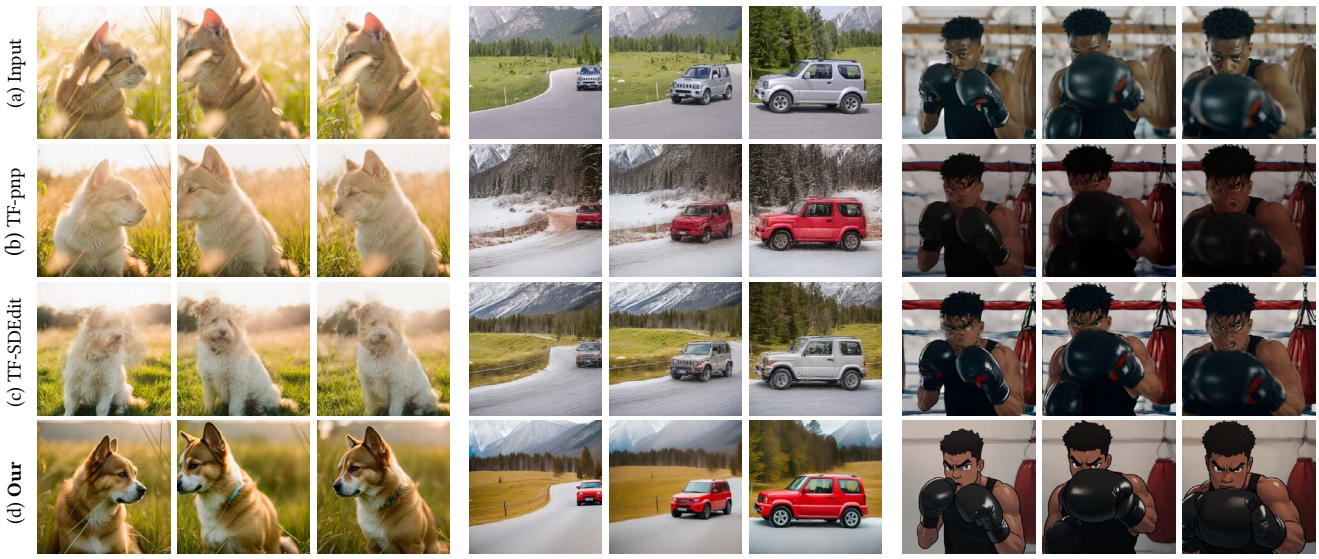


Figure 5. Visual comparison with inversion-based zero-shot TokenFlow. (Part I)

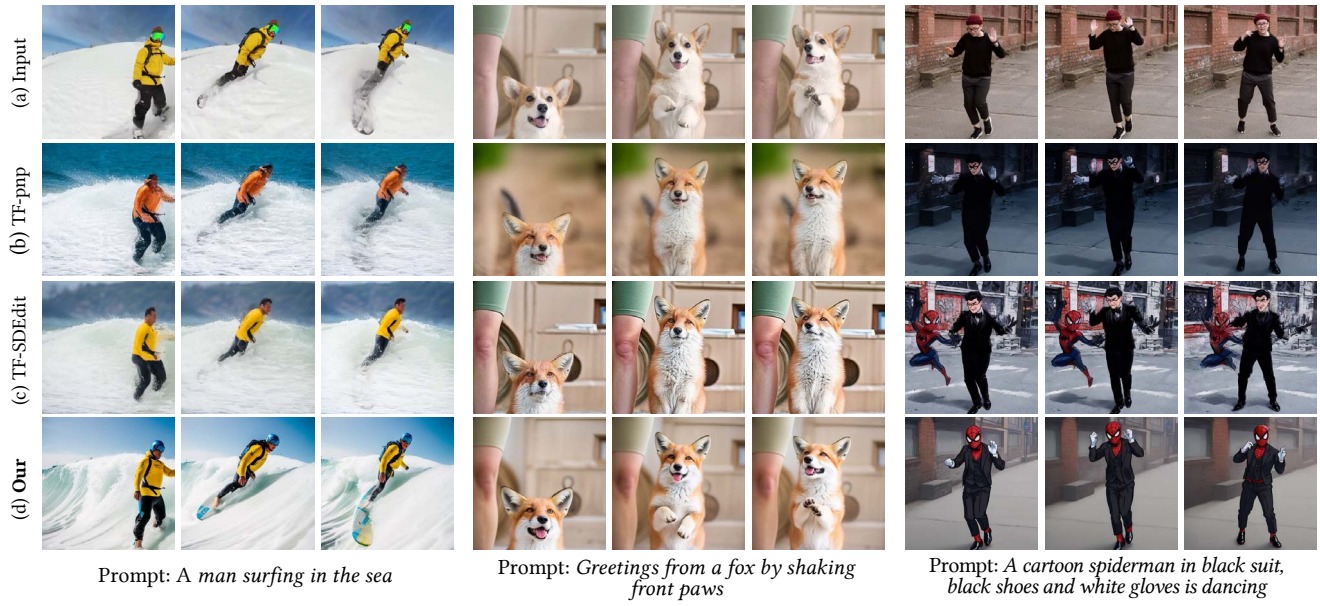


Figure 6. Visual comparison with inversion-based zero-shot TokenFlow. (Part II)

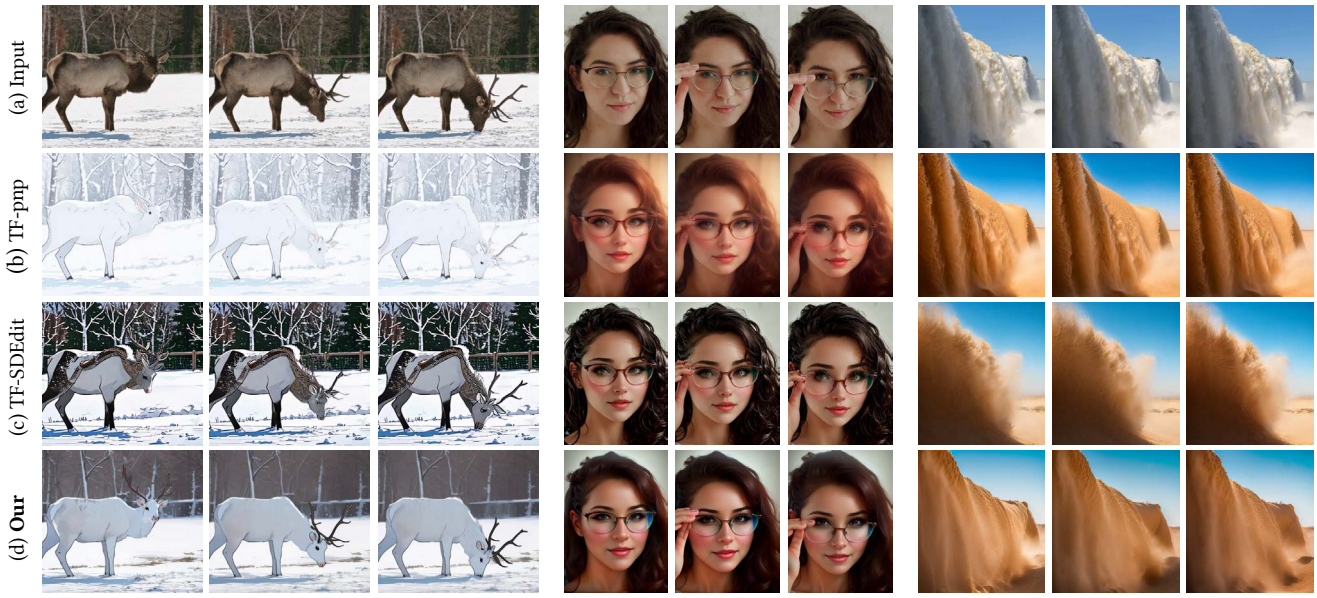


Figure 7. Visual comparison with inversion-based zero-shot TokenFlow. (Part III)

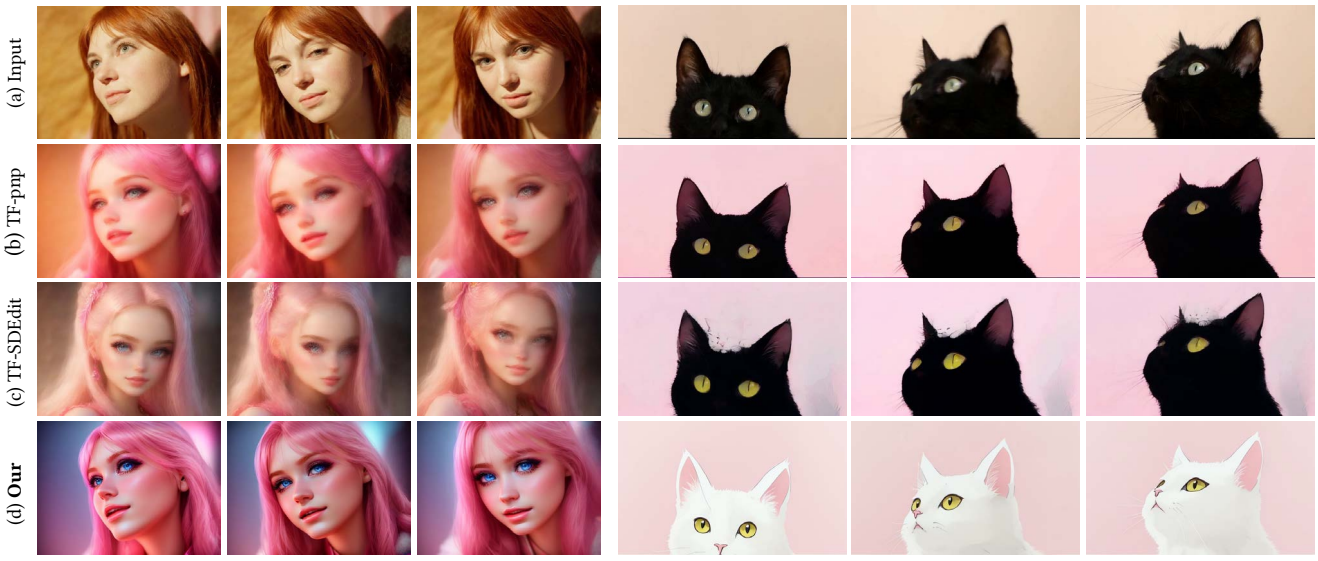
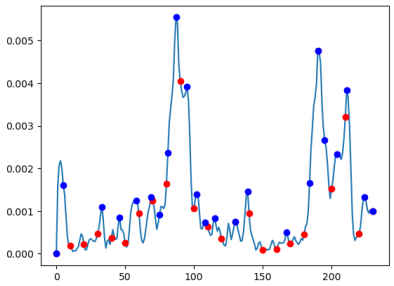


Figure 8. Visual comparison with inversion-based zero-shot TokenFlow. (Part IV)

4. Ablation Study

4.1. Keyframe selection

Figure 9 illustrates the effect of our heuristic keyframe selection algorithm. Compared with uniformly sampling, our sampling will more densely sample the frames where motions are large, indicated as the peaks in Fig. 9(a). In Fig. 9(b), it can be seen that the uniform sampling will choose many similar keyframes (*e.g.*, frames #0, #10, #20, #30, #40). Our heuristic keyframe selection algorithm samples more unique frames like the fully closed eyes in frames #75 in Fig. 9(c).



(a) Motion intensity and sampled frames in a video

(b) Uniformly sampled keyframes

(c) Keyframes sampled by our heuristic keyframe selection algorithm

Figure 9. Comparison between uniform sampling and the proposed heuristic keyframe selection algorithm. (a) The x-axis shows the frame indexes of the video. The y-axis shows the motion intensity (measured by L_2 distance between frames). The red dots are the uniformly sampled indexes. The blue dots are the sampled indexes by our heuristic keyframe selection algorithm. (b) Uniformly sampled keyframes. (c) Keyframes sampled by our heuristic keyframe selection algorithm.

5. More Results

5.1. Applications

Video colorization. Our method can be applied to video recolorization. As shown in Fig. 10, we first translate the input video to a more colorful one with prompts like 'high saturation', 'high contrast'. Then, we apply guided filter to filter the translated frame with the input frame as guidance to preserve the content. The resulting video has higher saturation and contrast.

Figure 10. Video recolorization. Prompt: A woman raises a gun, golden hair, white pearl necklace, high saturation, high contrast.

Cartoon ↔ photo. Using models customized for cartoons or photos, we can achieve non-photorealistic and photorealistic rendering in Fig. 11.

Prompt: A red hair girl wearing a white uniform is talking to the camera in the night

Prompt: A woman swimming in the pool in cartoon style

Figure 11. Application: Cartoon ↔ Photo. Top row: input. Bottom row: our results.

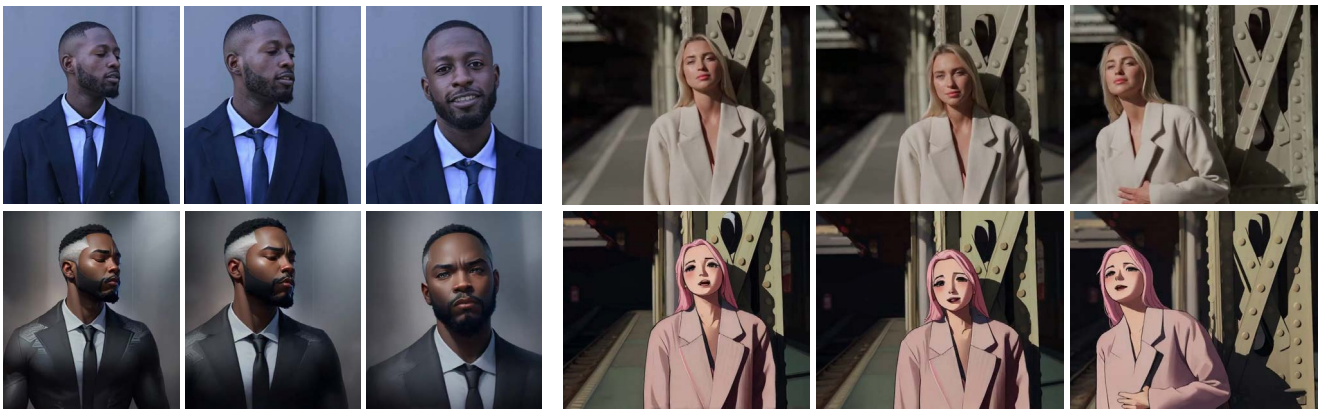
Photo → fantasy. Using the fantasy description as prompts, we can translate real scenes and objects into scenes or objects that do not exist in real life in Fig. 12.

Prompt: A cyberpunk fox in the cyberpunk style

Prompt: A paladin walks in the dark forest carrying a holy sword

Figure 12. Application: Real ↔ Fantasy. Top row: input. Bottom row: our results.

Identity transfer. By specifying a specific character name, we can change the identity of the character in the input video, as shown in Fig. 13.



Prompt: A black panther in CG style

Prompt: Cartoon Haruno Sakura looking in the railway station, flat, 2D

Figure 13. Application: identity transfer. Top row: input. Bottom row: our results.

5.2. Long video translation

Figure 14 presents an example of long video translation. A 16-second video comprising 400 frames are processed, where 32 frames are selected as keyframes for diffusion-based translation and the remaining 368 non-keyframes are interpolated. Thank to our FRESCO guidance to generate coherent keyframes, the non-keyframes exhibit coherent interpolation.

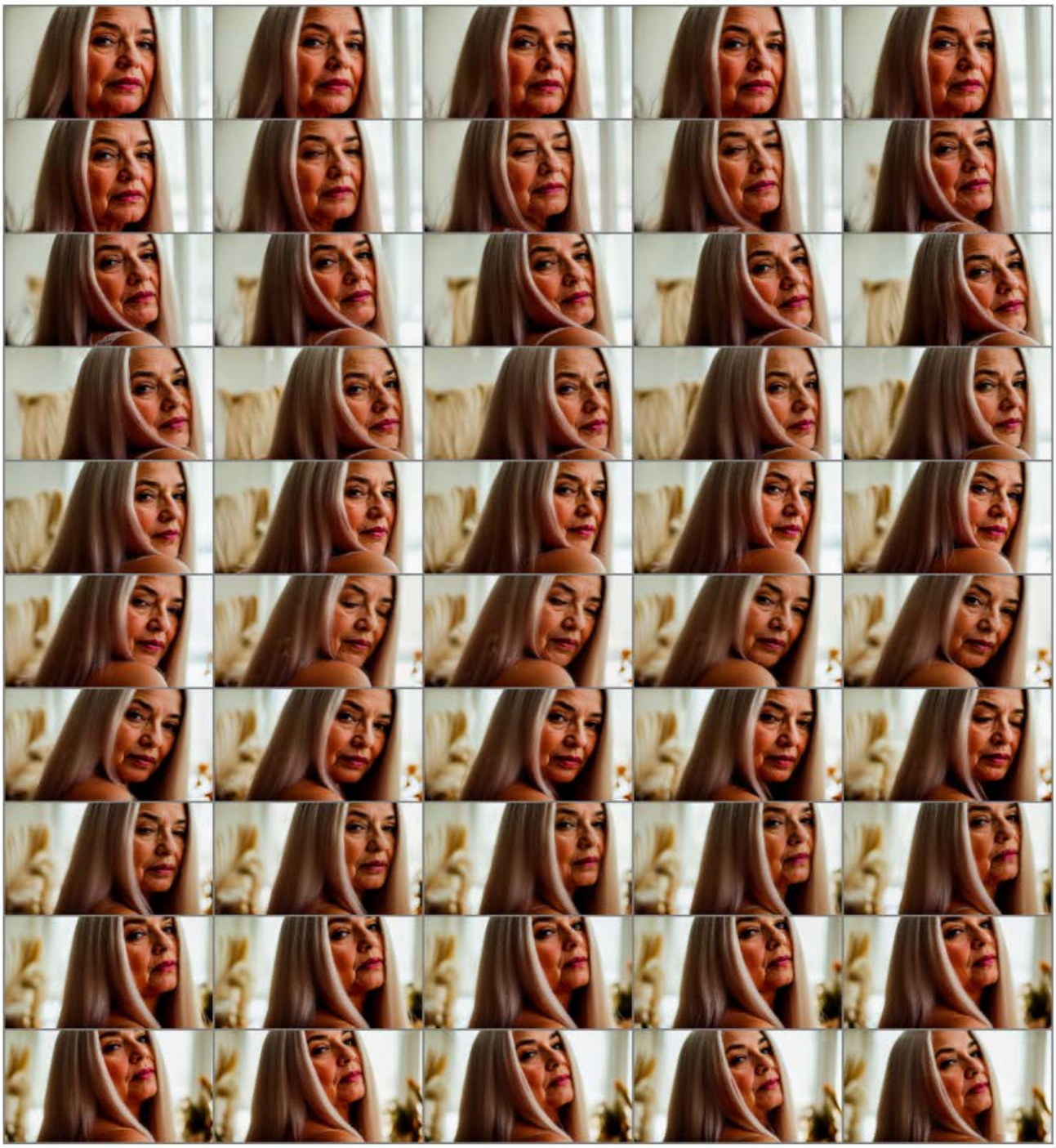


Figure 14. Long video generation.

6. Potential Negative Societal Impacts

Our model can be used to synthesize videos. The model may be applied to generate fake videos, which can be potentially avoided by using more advanced fake video detection methods.

References

[1] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. FLATTEN: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023. 2

[2] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 5

[3] 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. In *Proc. Int'l Conf. Computer Vision*, 2023. 2, 5

[4] 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 *Proc. Int'l Conf. Learning Representations*, 2021. 2, 5

[5] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 5

[6] 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 *Proc. Int'l Conf. Computer Vision*, pages 7623–7633, 2023. 2

[7] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 2

[8] Shuai Yang, Yifan Zhou, Ziwei Liu, , and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2023. 2, 5

[9] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. ControlVideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023. 2, 5