

VideoDirector: Precise Video Editing via Text-to-Video Models

Supplementary Material

A. Base Models Comparison

Our approach offers a fair comparison with other methods since most video editing methods rely on SD along with auxiliary models. Our base model, AnimateDiff, is built upon SD1.5, with a motion module (MM) trained while keeping the weights of SD1.5 frozen, ensuring a fair comparison. Base models and auxiliary models of different methods are presented in Tab. A.

Model	FRESCO [5]	RAVE	Tokenflow	Flatten	V-P2P	Ours
Base	SD1.5	SD1.5	SD2.1	SD1.5	SD2.1	SD1.5
Auxiliary	Controlnet, GMFlow	Controlnet	-	T2S	RAFT	MM, SAM2

Table A. Comparison of base models.

B. More Results

Comparison with FRESCO [5]: Our method (Fig. B) achieves better visual quality than FRESCO (Fig. A) and outperforms it across all metrics (Tab. B). The GIF requires *Adobe Acrobat Reader* to be viewed.

Methods	MS \uparrow	PS \uparrow	m.P \uparrow	m.L \downarrow
FRESCO	96.28%	21.53	18.43	0.368
Ours	97.68%	21.64	21.37	0.270

Table B. Comparison.

Figure A Figure B

Dynamic Results: Directly integrating DDIM inversion-based methods with T2V models cannot produce satisfactory results (Fig. 2, Fig. C-D GIFs, **V.opt** means vanilla null-text optimization). Our approach demonstrates the feasibility of controlling the denoising trajectory for precise editing via T2V model. Our STDG maintains the same spatial resolution as the latent noise, integrating both temporal and appearance cues. This alignment guides the denoising trajectory toward precise resampling and editing (Fig. C-H, the GIF requires *Adobe Acrobat Reader* to be viewed).

More Results: More edited results are shown in Fig K to Fig L, Fig M, and Fig N. along with our editing prompts. Additionally, we provide an MP4 video in the supplementary material.

(a) input (b) V.opt. (c) ours (a) input (b) V.opt. (c) ours
Figure C. Recon. example #1. Figure D. Recon. example #2.

(a) input (b) w/o STDG (c) ours (a) input (b) w/o STDG (c) ours
Figure E. Editing example #1. Figure F. Editing example #2.

(a) input (b) w/o STDG (c) ours (a) input (b) w/o STDG (c) ours
Figure G. Editing example #3. Figure H. Editing example #4.

C. Discussion about Null-text Optimization

Replacing the multi-frame strategy with a shared null-text embedding is effective for objects with minimal deformation, such as the “driving car” shown in Fig. I. In these cases, the STDG provides sufficient temporal and motion guidance. However, relying solely on the STDG leads to suboptimal reconstruction and editing results in videos with dynamic objects that undergo significant deformation, as illustrated in Fig. I. Multi-frame null-text optimization is crucial for videos featuring such dynamic objects. While the STDG offers global temporal and spatial guidance, the null-text embedding refines detailed motion and appearance information by building on the STDG and pivotal latent.

D. Discussion about SAM2 Mask

While the mask generated by SAM2 is able to segment fine structures, these rich details can make the editing process fragile and vulnerable to disruptions caused by segmentation masks, as shown in Fig. J. To mitigate this issue, we combine the mask with an ellipse mask that is coarsely aligned with the mask during the pivotal inversion and editing process. In this way, the combined mask enhances robustness of our method to mask disruptions and improves the harmony between the edited and the remaining contents, as illustrated in Fig. J.

E. Pseudo Code

The pseudo-code for our method is provided in Algorithm 1. Descriptions of the variables used in the algorithm can be found in Sec. 3. **Stage 1** corresponds to Sec. 3.2, and **Stage 2** corresponds to Sec. 3.3. Here, e_t^* denotes the DDIM sampling latents of the editing path in **Stage 2**.

F. Limitation

The edited videos in this paper are limited to 16 frames due to the high memory cost of the T2V model. Our method consumes approximately 16GB more GPU memory usage compared to Video-p2p [3]. In the future, we will further focus on extending the method to handle longer video sequences.

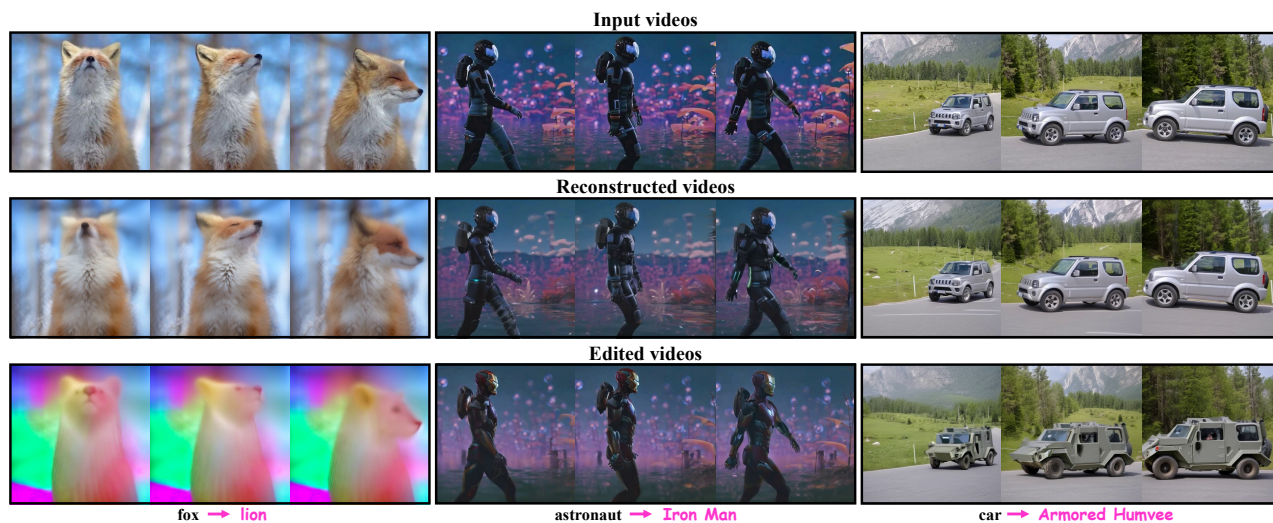


Figure I. *Shared Null-text optimization used for reconstruction and editing.*

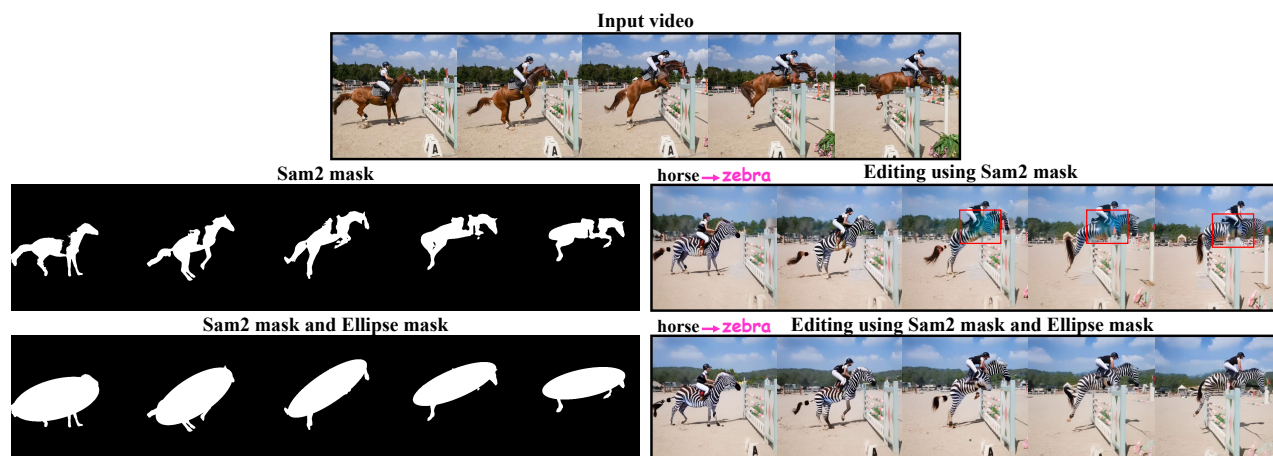
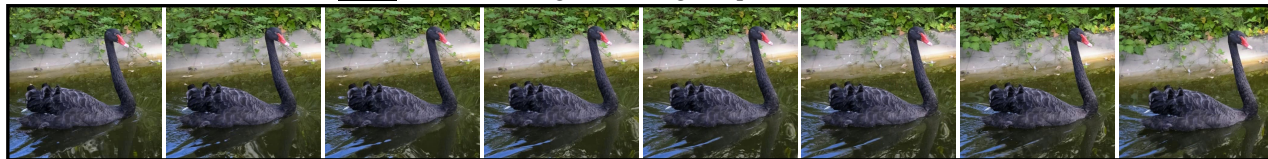
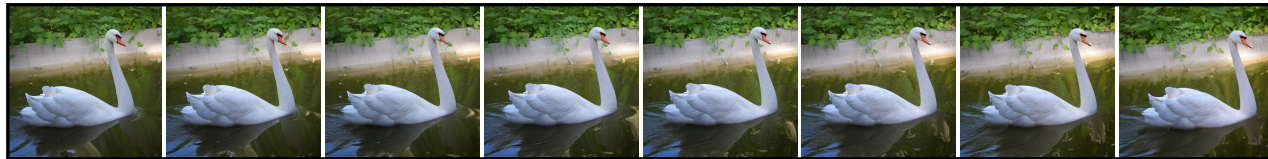


Figure J. *Sam2 Mask combines the ellipse mask to enhance the editing robustness.*

A **black** swan swimming in a river, green plants on the bank.



A **white** swan swimming in a river, green plants on the bank.



A **blue** swan swimming in a river, green plants on the bank.

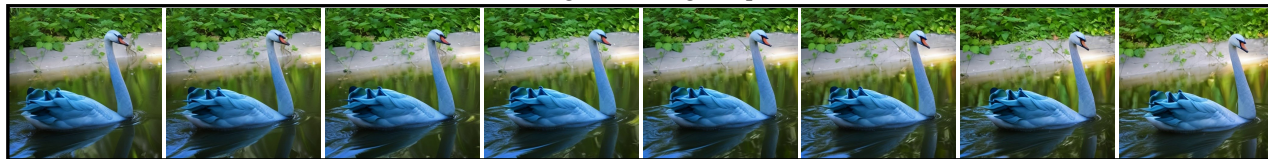


Figure K. *More results.*

A wolf is turning head with some trees in the background



A cheetah is turning head with some trees in the background



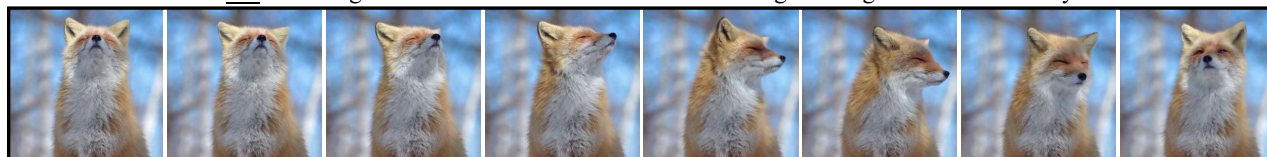
A husky is turning head with some trees in the background



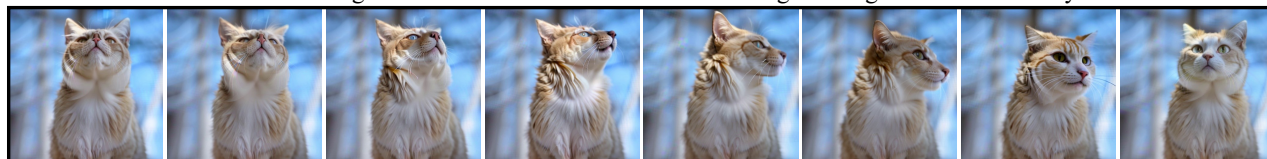
A lion is turning head with some trees in the background



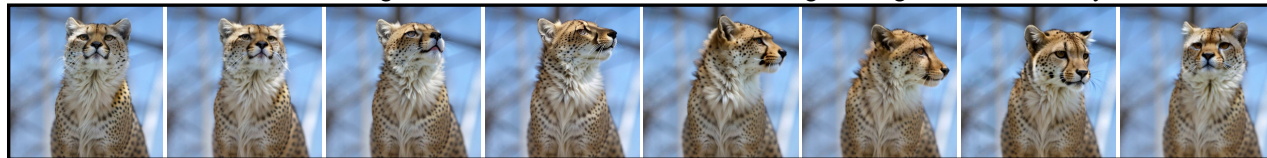
A fox is turning head with some trees blurred in the background against a soft blue sky.



A cat is turning head with some trees blurred in the background against a soft blue sky.



A cheetah is turning head with some trees blurred in the background against a soft blue sky.



A lion is turning head with some trees blurred in the background against a soft blue sky.



Figure L. *More results.*

A car is drifting on the track of a racing circuit.



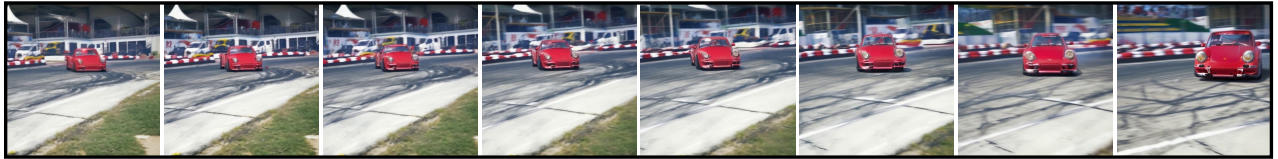
A **red Tesla** is drifting on the track of a racing circuit.



A **silver Porsche** is drifting on the track of a racing circuit.



A **red Porsche** is drifting on the track of a racing circuit.



A rhino is walking on the ground with stones besides, rocks and trees in the background



A **lion** is walking on the ground with stones besides, rocks and trees in the background



A **tiger** is walking on the ground with stones besides, rocks and trees in the background



A **hippopotamus** is walking on the ground with stones besides, rocks and trees in the background



Figure M. *More results.*

A car driving through an intersection with some roads and buildings in the background.



An armored Humvee driving through an intersection with some roads and buildings in the background.



A LEGO car driving through an intersection with some roads and buildings in the background.



A Porsche Cayenne driving through an intersection with some roads and buildings in the background.



Kid is playing football on a soccer field with many trees in the background.



Messi is playing football on a soccer field with many trees in the background.



Cristiano Ronaldo is playing football on a soccer field with many trees in the background.



Kid is playing football on a Worldcup soccer stadium with spectators in the background.



Figure N. *More results.*

Algorithm 1 VideoDirector

Require: Input: video $V_i \in \mathbb{R}^{F \times H \times W}$, regularization term \mathcal{R} : SAM2 masks $\mathcal{M} \in \mathbb{R}^{F \times H \times W}$ [4], original and editing prompts: \mathcal{C} and \mathcal{C}^e , generation model G: T2V diffusion network ϵ_θ [2].

Ensure: Edited video $V_o \in \mathbb{R}^{F \times H \times W}$.

Stage 1: Video Pivotal Inversion

```

1:  $z^* = \mathcal{E}(V_i)$  ▷ Encoder  $\mathcal{E}(\cdot)$  convert the input video to latents.
2: for  $t = 0$  to  $T$  do ▷ Iterate over  $T$  timesteps.
3:    $z_{t+1}^* = \sqrt{\alpha_{t+1}} \left( \frac{z_t^* - \sqrt{1 - \alpha_t} \epsilon_\theta(z_t^*)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t+1}} \epsilon_\theta(z_t^*)$  ▷ DDIM inversion.
4: end for
5: for  $t = T$  to  $0$  do ▷ Iterate over  $T$  timesteps in reverse.
6:    $\mathcal{T}_+ = \epsilon_\theta^{(T)}(z_t^*, \mathcal{C}, t)$ ,  $\mathcal{T}_- = \epsilon_\theta^{(T)}(z_t, \mathcal{C}, t)$  ▷ Extract temporal features.
7:    $\mathcal{K}_+ = \epsilon_\theta^{(K)}(z_t^*, \mathcal{C}, t)$ ,  $\mathcal{K}_- = \epsilon_\theta^{(K)}(z_t, \mathcal{C}, t)$  ▷ Extract spatial features.
8:    $\mathcal{L}_\mathcal{T} = \mathcal{M}_\mathcal{T}^{f/b} \cdot \mathcal{M}_\mathcal{T} \cdot \|(\mathcal{T}_+ - \mathcal{T}_-)\|_2^2$ ,  $\mathcal{G}_\mathcal{T}^{f/b} = \frac{\partial(\mathcal{L}_\mathcal{T})}{\partial z_t}$  ▷ Temporal Guidance.
9:    $\mathcal{L}_\mathcal{K} = \mathcal{M}_\mathcal{K}^{f/b} \cdot \|(\mathcal{K}_+ - \mathcal{K}_-)\|_2^2$ ,  $\mathcal{G}_\mathcal{K}^{f/b} = \frac{\partial(\mathcal{L}_\mathcal{K})}{\partial z_t}$  ▷ Spatial Guidance.
10:   $\mathcal{G}_t = \eta_f \cdot \mathcal{G}_\mathcal{T}^f + \eta_b \cdot \mathcal{G}_\mathcal{T}^b + \zeta_f \cdot \mathcal{G}_\mathcal{K}^f + \zeta_b \cdot \mathcal{G}_\mathcal{K}^b$  ▷ Total Guidance.
11:  for  $iter = 0$  to  $N$  do ▷ Iterative Null-text Optimize for  $N$  steps.
12:     $\hat{\epsilon}_\theta = \epsilon_\theta(z_t, \mathcal{C}, t) + \omega[\epsilon_\theta(z_t, \mathcal{C}, t) - \epsilon_\theta(z_t, \{\phi_t\}, t)]$  ▷ CFG.
13:     $\bar{\epsilon}_\theta = \hat{\epsilon}_\theta - (\sqrt{1 - \alpha_t}) \mathcal{G}_t$  ▷ STDG, the guidance is applied following the formula (14) from [1].
14:     $z_{t-1} = \sqrt{\alpha_{t-1}} \left( \frac{z_t - \sqrt{1 - \alpha_t} \hat{\epsilon}_\theta}{\sqrt{\alpha_t}} \right) + (\sqrt{1 - \alpha_{t-1}}) \bar{\epsilon}_\theta$  ▷ DDIM sampling.
15:     $\mathcal{L}(\{\phi_t\}) = \|z_{t-1}^* - z_{t-1}\|_2^2$  ▷ Null-text Optimize.
16:  end for
17: end for

Stage 2: Attention Control for Video Editing
18: for  $t = T$  to  $0$  do ▷ DDIM sampling.
19:   for  $l = 0$  to  $L$  do ▷ Pass through the U-Net of the T2V model.
20:     $Q_t^{(l)} = \epsilon_\theta^{(l)(Q)}(z_t^*)$ ,  $K_t^{(l)} = \epsilon_\theta^{(l)(K)}(z_t^*)$ ,  $V_t^{(l)} = \epsilon_\theta^{(l)(V)}(z_t^*)$  ▷ Extract  $Q, K, V$  of reconstruction path.
21:     $Q_t^{*(l)} = \epsilon_\theta^{(l)(Q)}(e_t^*)$ ,  $K_t^{*(l)} = \epsilon_\theta^{(l)(K)}(e_t^*)$ ,  $V_t^{*(l)} = \epsilon_\theta^{(l)(V)}(e_t^*)$  ▷ Extract  $Q, K, V$  of editing path.
22:    if SelfAttention then ▷ Self Attention Control.
23:       $\widehat{Attn} = \begin{cases} W_t^{(l)} \cdot V_t^{*(l)}, & \text{if } t < \tau_s, \\ S \left( \frac{Q_t^{*(l)} \cdot \hat{K}_t^\top}{\sqrt{d}} \otimes [\mathbf{1} \mid \mathcal{M}^f] \right) \cdot \hat{V}_t, & \text{otherwise.} \end{cases}$  ▷ Calculate attention features in SA-I and SA-II.
24:    else if CrossAttention then ▷ Cross Attention Control.
25:       $M_t^{C(l)} = \begin{cases} \mathcal{C} \cdot [\gamma \cdot (M_t^{*(l)}) + (1 - \gamma) \cdot (M_t^{(l)})], & \text{if } t < \tau_c, \\ M_t^{*(l)}, & \text{otherwise.} \end{cases}$  ▷ Calculate Cross Attention Maps.
26:    end if
27:    Update edited latent  $\epsilon_\theta^{(l)}(e_t^*)$ . ▷ This edited latent updating contains  $\epsilon_\theta^{(l)}(e_t^*, \mathcal{C}, t)$  and  $\epsilon_\theta^{(l)}(e_t^*, \{\phi_t\}, t)$ .
28:  end for
29:   $\hat{\epsilon}_\theta = \epsilon_\theta(e_t^*, \mathcal{C}, t) + \omega[\epsilon_\theta(e_t^*, \mathcal{C}, t) - \epsilon_\theta(e_t^*, \{\phi_t\}, t)]$  ▷ CFG.
30:   $\bar{\epsilon}_\theta = \hat{\epsilon}_\theta - (\sqrt{1 - \alpha_t}) \mathcal{G}_t$  ▷ STDG.
31:   $e_{t-1}^* = \sqrt{\alpha_{t-1}} \left( \frac{e_t^* - \sqrt{1 - \alpha_t} \hat{\epsilon}_\theta}{\sqrt{\alpha_t}} \right) + (\sqrt{1 - \alpha_{t-1}}) \bar{\epsilon}_\theta$  ▷ DDIM sampling using edited latent.
32: end for
33: return  $V_o = \mathcal{DE}(e_0^*)$ . ▷ Decoder  $\mathcal{DE}(\cdot)$  convert edited latents into output video.

```

References

- [1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34, 2021. [6](#)
- [2] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yao-hui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *ICLR*, 2024. [6](#)
- [3] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *CVPR*, 2024. [1](#)
- [4] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. [6](#)
- [5] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Fresco: Spatial-temporal correspondence for zero-shot video translation. In *CVPR*, 2024. [1](#)