VideoSwap: Customized Video Subject Swapping with Interactive Semantic Point Correspondence

Supplementary Material

Notice: The video results, included in both the main paper and the supplementary material, are accessible on the webpage located within the supplementary folder. It is worth noting that ***the webpage is static and NOT modifiable once uploaded as the supplementary file, and all linked videos are included in the folder as well***.

Contents

1. Additional Details about Methods 1
- 1.1. Latent Blend . . . 1
- 1.2. Drag-based Point Control . . . 1
- 1.3. Discussion the Relation to Human Keypoint . 2

2. Experimental Details 3
- 2.1. Implementation Details . . . 3
- 2.2. Time Cost Analysis . . . 3
- 2.3. Memory Cost Analysis . . . 3

3. Quantitative Evaluation 3
- 3.1. Dataset and Evaluation Setting . . . 3
- 3.2. Automatic Evaluation by CLIP-Score . . . 3
- 3.3. Human Evaluation Interface . . . 4
- 3.4. Human Evaluation for Ablation Study . . . 4

4. Qualitative Evaluation 5

5. Limitation and Future Works 5
- 5.1. Limitation Analysis . . . 5
- 5.2. Future Works . . . 5
- 5.3. Potential Negative Social Impact . . . 6

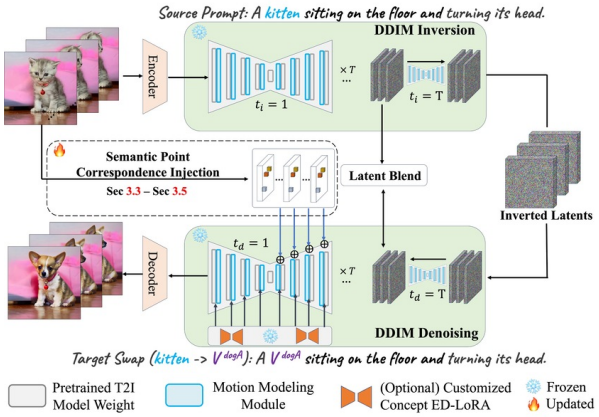

Figure 1. Overview of the VideoSwap pipeline for customized video subject swapping.

1. Additional Details about Methods

1.1. Latent Blend

Given our focus on subject swapping, where the objective is to maintain the unedited background region identical to the source video, this is achieved through latent blend [1, 2], as shown in Fig. 1.

The key idea is that the latent noise in DDIM denoising and DDIM inversion provides information for the swapped subject and background, respectively. These two latent noises can be blended using a mask that indicates the foreground region, thus blending the swapped target with the source background.

To initiate the process, we acquire the foreground mask for timestep t as $\mathcal{M}^t = \mathcal{M}_i^t \cup \mathcal{M}_d^t$, formed by merging the subject masks $\mathcal{M}_i$ during inversion and $\mathcal{M}_d$ during denoising at the same timestep t. This subject mask is automatically generated through the cross-attention of the concept token, following the approach of Prompt2Prompt [3].

Subsequently, the foreground mask is used to blend the latent features, resulting in $z^t = (1 - \mathcal{M}^t) \cdot z_i^t + \mathcal{M}^t \cdot z_d^t$, where z_i^t and z_d^t represent the latent features of timestep t in DDIM inversion and DDIM denoising, respectively. Through latent blend, we can effectively preserve the unedited background in the source video.

1.2. Drag-based Point Control

1.2.1 Layered Neural Atlas Training

As mentioned in Sec. 3.5 of the main paper, we introduce interactive dragging on the key frame for handling point correspondence with shape morphing in customized video

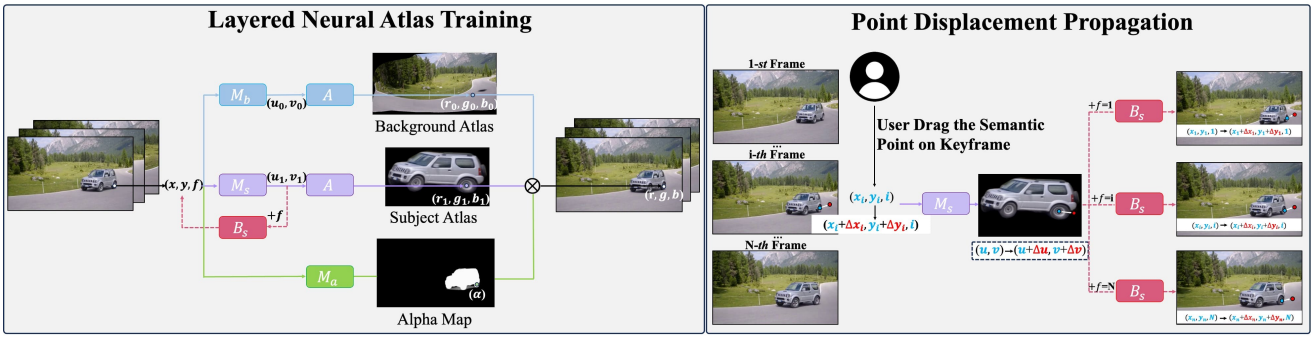

Figure 2. Point displacement propagation based on layered neural atlas (LNA) [5, 7]. Once a trained LNA is provided, users can drag a semantic point at the keyframe, and this displacement is consistently propagated to every frame through the canonical space of the LNA.

subject swapping. This function is supported by the learned canonical space of Layered Neural Atlas [7] (LNA). Here, we present a detailed formulation of LNA.

LNA [7] represents a video through the following three sets of parameterized MLPs:

1. **Coordinate Mapping MLPs**. The coordinate mapping MLPs map the spatial-temporal coordinates of video pixels to the 2D canonical space (*i.e.*, the UV map), denoted as M: $(x, y, f) \rightarrow (u, v)$. We employ separate mappings, M_s and M_b, for the foreground subject and background, respectively. Additionally, following the approach of INVE [5], we include a background mapping B_s: $(u, v) \rightarrow (x, y, f)$ to learn the coordinate mapping of the foreground subject from the canonical space back to the video pixel.
2. **Atlas MLPs**. The atlas MLPs, denoted as A: $(u, v) \rightarrow (r, g, b)$, learn to predict the color of the coordinates on the UV map.
3. **Alpha MLPs**. The alpha MLPs, denoted as M_α: $(x, y, f) \rightarrow \alpha$, predict the blending ratio α of the color value from the subject atlas and background atlas.

Based on these sets of learnable MLPs, the training objective of LNA is to reconstruct the RGB values of the source video, accompanied by the following regularization losses:

1. **Rigidity Loss**. The rigidity loss encourages the learned mapping from pixel coordinates in the video to the 2D canonical space to exhibit local rigidity.
2. **Consistency Loss**. The consistency loss encourages the mapping of corresponding video pixels across consecutive frames to be consistent, with correspondence estimated through pre-computed optical flow.
3. **Sparsity Loss**. The sparsity loss encourages the many-to-one mapping from the video coordinates to the canonical coordinates, penalizing duplicate contents in the canonical space.

We refer the reader to the LNA [7] paper for the complete formulation.

1.2.2 Point Displacement Propagation based on LNA

After learned LNA representation, we can propagate the dragged displacement at the keyframe to the whole video through LNA. Given a semantic point, with coordinates at the keyframe f_{key} represented as (x, y, f_{key}), its trajectory over time can be expressed as a function of time f: $(x(f), y(f)) = P(f)$. Suppose a user drag it to a new position at $(x + dx, y + dy, f_{key})$, we aim to estimate the edited trajectory $P'(f)$ for $f = \{0, ..., N\}$. We resort to LNA's representation, and first compute a linearized estimation of its shifted position on the canonical coordinate:

$$[du, dv]^T = J_M(x, y, f_{key})[dx, dy]^T, \quad (1)$$

where J_M denote the Jacobian matrix with respect to (x, y). Next, at a given time f, we estimate the edited coordinate in the pixel space as

$$P'(f) = P(f) + J_B(u, v, f)[du, dv]^T, \quad (2)$$

where $(u, v) = B(x, y, f_{key})$ and J_B denote the Jacobian matrix with respect to (u, v). In practice, we approximate the Jacobian computation by

$$J_M = \begin{bmatrix} M_s(x+\varepsilon, y, f) - M_s(x, y, f) \\ M_s(x, y+\varepsilon, f) - M_s(x, y, f) \end{bmatrix}^T \begin{bmatrix} 1/\varepsilon \\ 1/\varepsilon \end{bmatrix}, \quad (3)$$

$$J_B = \begin{bmatrix} B_s(u+\varepsilon, v, f) - B_s(u, v, f) \\ B_s(u, v+\varepsilon, f) - B_s(u, v, f) \end{bmatrix}^T \begin{bmatrix} 1/\varepsilon \\ 1/\varepsilon \end{bmatrix}, \quad (4)$$

where ε represents the small coordinate shift. We then use this edited trajectory $P'(f)$ for the dragged semantic point during inference.

1.3. Discussion the Relation to Human Keypoint

The ControlNet [25] and T2I-Adapter [13] also incorporate control over human keypoints. These human keypoints can be viewed as a type of sparse semantic points, where the semantic position and total number of human keypoints are predefined by the existing pose detectors, and their semantic embedding for controlling the diffusion model is implicitly aligned through large-scale paired data. However, defining keypoints or collecting paired data for open-set concepts proves challenging due to the variability in semantic points.

Therefore, our method provides a more generic framework for point-based video editing, with human keypoints serving as a specific use case within our framework.

2. Experimental Details

2.1. Implementation Details

We implement our method using the Latent Diffusion Model [18] and incorporate the pretrained motion layer from AnimateDiff [2] as the foundational model. All experiments are conducted on an Nvidia A100 (40GB) GPU. All video samples consist of 16 frames with a time stride of 4, matching the temporal window of the motion layer in AnimateDiff. We crop the videos to two alternate resolutions ($H \times W$): 512×512 and 448×768. For all experiments, we employ the Adam optimizer with a learning rate of 5e-5, optimizing for 100 iterations. Regarding the point patch loss, we use a patch size of 4×4 around the semantic point.

2.2. Time Cost Analysis

In this section, we analyze the time cost of editing a video in VideoSwap. All time costs are calculated on an Nvidia A100 GPU to process a 16 frame video clip.

Time Cost of Preprocess. The preprocessing step involves (1) extracting point trajectories and their DIFT embeddings, and (2) registering those semantic points to guide the diffusion model, and (3) generate DDIM-inverted noise. The extraction of trajectories and embeddings takes approximately 30 seconds. The registration step requires 100 iterations, taking about 3 minutes. And the DDIM inversion of 50 steps takes approximately 30 seconds. To summarize, it takes about 4 minutes to pre-process a video for editing.

Time Cost of Each Edit. Then for each edit, the time cost of VideoSwap remain the similar to AnimateDiff [2], necessitating 50 seconds with the latent blend technique. The introduction of semantic point correspondence does not notably increase the time cost, given its lightweight computation.

Time Cost of User-Point Interaction. The time cost for user-point interaction (*e.g.*, removing or dragging a point) can be negligible. Dragging a point at the keyframe only takes 1 seconds to propagate to all other frames through a learned layered neural altas (LNA).

Extra Time Cost in Training LNA. Our support for drag-based editing is built upon a learned LNA of the given video. In contrast to the original LNA, which necessitates approximately 10 hours of training, we do not require full training as we only adopt the forward/backward coordinate mapping. This training process takes about 2 hours for a video.

Methods/Metrics	Text Alignment (↑)	Image Alignment (↑)	Temporal Consistency (↑)
Compare to Previous Video Editing Methods			
Tune-A-Video [22]	25.34	-	95.79
FateZero [17]	24.39	-	95.49
Text2Video-Zero [9]	24.85	-	95.02
Rerender-A-Video [24]	24.99	-	92.28
VideoSwap (Ours)	**26.87**	-	**95.93**
Compare to Baselines on AnimateDiff			
w/ DDIM	27.36	79.79	95.89
w/ DDIM + TAV	24.75	75.93	95.49
w/ DDIM + T2I-Adapter	25.86	77.54	95.50
VideoSwap (Ours)	26.87	**79.87**	**95.93**

Table 1. Automatic Quantitative Evaluation on Video Subject Swapping Results.

2.3. Memory Cost Analysis

The overall memory cost is similar to AnimateDiff, where we don't incur significant additional memory costs, as our semantic points and MLPs are lightweight. It only requires a memory cost of 16/12 GB for point registration and inference, respectively.

3. Quantitative Evaluation

3.1. Dataset and Evaluation Setting

We collect 30 videos from Shutterstock and DAVIS [16]. Each category—human, animal, and object—comprises 10 videos. Besides, we gather 13 customized concepts: 5 for human characters, 3 for animals, and 5 for objects. Due to legal concerns, we cannot demonstrate qualitative results involving human characters. For each source video, we adopt 8 predefined concepts and 2-5 customized concepts as swap targets, yielding approximately 300 edited results. For comparison to previous video-editing methods that don't support customized concepts, we only compute the metric on predefined concepts. In comparison to the baselines built upon AnimateDiff [2], we compute the metric on both predefined concepts and customized concepts.

3.2. Automatic Evaluation by CLIP-Score

We conduct a quantitative evaluation using the automatic metric, CLIP-Score [4]. The metric includes text alignment and temporal consistency, following [23]. Additionally, for customized concepts, we follow Custom Diffusion [10] to compute pairwise image alignment between each edited frame and each reference concept image. The results are summarized in Table. 1. In comparison to previous video editing methods, VideoSwap demonstrates the best text alignment and temporal consistency. Moreover, when compared to baselines built on AnimateDiff, we achieve superior image alignment and temporal consistency. However, it is important to note that CLIP-Score is primarily based on frame-wise computation and may not align well with human perception, as discussed in EvalCrafter [11]. Therefore, we

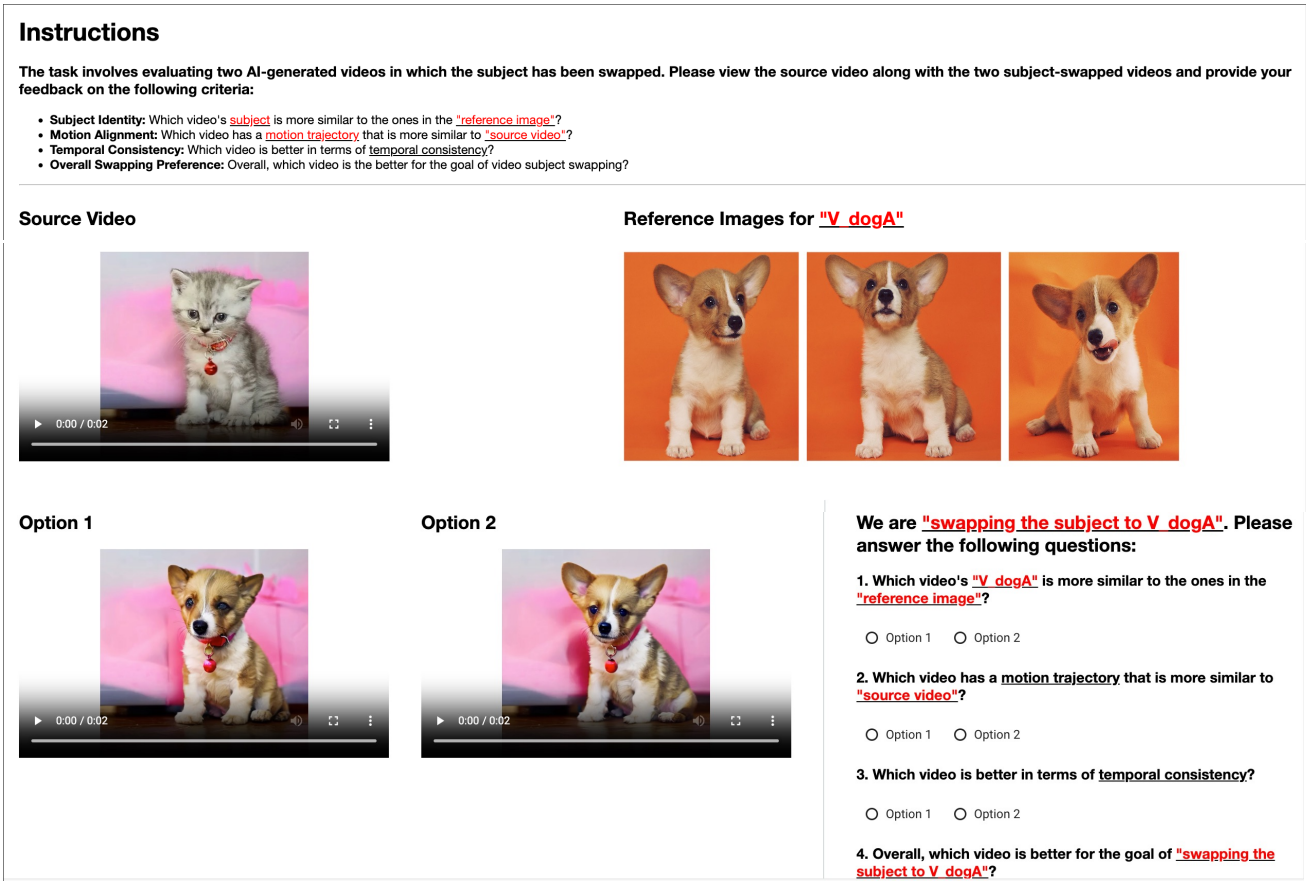

Figure 3. Human evaluation interface on Amazon Mturk. We provide the source video and reference images for target concept and ask user to select favorable video in terms of different criteria of video subject swapping.

	Methods/Metrics	Subject Identity	Motion Alignment	Temporal Consistency	Overall Preference
	Ablation of Sparse Motion Feature				
VideoSwap v.s.	Point Map + T2I-Adapter (100 iters)	**87%** v.s. 13%	**90%** v.s. 10%	**87%** v.s. 13%	**90%** v.s. 10%
VideoSwap v.s.	Learnable Embedding + MLP (100 iters)	**87%** v.s. 13%	**95%** v.s. 5%	**90%** v.s. 10%	**90%** v.s. 10%
VideoSwap v.s.	Learnable Embedding + MLP (300 iters)	**52%** v.s. 48%	**52%** v.s. 48%	**52%** v.s. 48%	**55%** v.s. 45%
	Ablation of Point Patch Loss				
VideoSwap v.s.	w/o. Point Patch Loss	**73%** v.s. 27%	**73%** v.s. 27%	**78%** v.s. 22%	**78%** v.s. 22%
	Ablation of Semantic-Enhanced Schedule				
VideoSwap v.s.	w/o. Semantic-Enhanced Schedule	**85%** v.s. 15%	**90%** v.s. 10%	**90%** v.s. 10%	**87%** v.s. 13%

Table 2. Human Evaluation for Ablation Study in VideoSwap. VideoSwap utilizes DIFT embedding + MLP (100 iterations) and incorporates the point patch loss and a semantic-enhanced schedule to improve the learning of semantic point correspondence.

present these results for reference purposes and primarily evaluate and compare using human evaluation.

3.3. Human Evaluation Interface

We primarily conduct human evaluations to compare different methods based on several criteria: subject identity, motion alignment, temporal consistency, and overall swapping preference. As depicted in Fig. 3, we present the source video and reference images for the target concept in the interface and ask users to select their preferred video based on various criteria related to customized video subject swapping. We distribute 1000 questionnaires on Amazon Mturk. The human evaluation results in Table. 1 of the main paper clearly demonstrate our advantage.

3.4. Human Evaluation for Ablation Study

We employ human evaluation to quantitatively assess various variants of our methods, and the results are summarized in Table. 2. In terms of creating sparse motion features, our DIFT embedding significantly outperforms point map +

(a) Tracking Error with Self-Occlusion

(b) Tracking Error with View-Change

Figure 4. Limitations in point tracking inherited from Co-Tracker [6] in scenarios involving self-occlusion and significant view changes.

T2I-Adapter and the learnable embedding + MLP with the same registration iterations. In comparison to the learnable embedding and MLP, our explicit DIFT embedding already contains sufficient semantic information, requiring 3× less time to achieve similar preference. The introduction of the point patch loss and semantic-enhanced schedule further enhances VideoSwap, leading to higher preferences compared to variants without these enhancements.

4. Qualitative Evaluation

Note that all our qualitative results and analysis are presented ***on the static webpage*** attached in the supplementary materials. We encourage readers to check out that webpage for a more comprehensive comparison.

5. Limitation and Future Works

5.1. Limitation Analysis

The limitation of VideoSwap is inherited from inaccurate point tracking and an imperfect canonical space representation of Layered Neural Atlas.

Inaccurate Point Tracking by Co-Tracker. VideoSwap relies on accurate point trajectory extraction. However, the existing point tracking method Co-Tracker [6] is not stable enough when the video contains self-occlusion and large view changes, as shown in Fig. 4(a) and Fig. 4(b). To address this issue, users may choose to remove inaccurate semantic points; however, this would result in less motion alignment. Nevertheless, since tracking any point is a newly formed problem, any progress in this area can seamlessly integrate into VideoSwap.

Imperfect Canonical Space by Layer Neural Atlas. As discussed in Layered Neural Atlas (LNA) [7], LNA fails to represent videos involving 3D rotations and non-rigid motion with self-occlusion. VideoSwap resorts to LNA to propagate the dragged point displacement. Therefore, due to the limitations of LNA, we cannot support drag-based interaction in such cases. Improvement in LNA representation will further broaden support for drag-based video editing.

Time Cost for Interactive Editing. The time cost of VideoSwap prohibits its use for real-time interactive editing. Setting up semantic points for a video takes approximately 4 minutes. And to support drag-based editing, an additional 2 hours are required to prepare the LNA for the given video. Furthermore, constrained by diffusion model sampling, it takes about 50 seconds to perform an edit, falling short of real-time editing. We anticipate that advancements in neural field acceleration [5, 8, 14] and diffusion model distillation [12, 19, 20] will significantly reduce the preprocess cost and enhance speed for real-time interactive editing.

5.2. Future Works

VideoSwap embarks on video editing with shape change. With semantic points as correspondence, VideoSwap can support interactive editing for large shape changes while aligning motion trajectories. We list several promising directions motivated by VideoSwap.

Interactive Video Editing. VideoSwap supports drag-based interaction at the keyframe, propagating the dragged displacement throughout the entire video and obtaining the source and dragged trajectories with similar motion. As we can obtain the source point trajectory and target point trajectory, future work may extend the idea of DragGAN [15] to the video domain for drag-based real video editing.

Video Editing with Shape Change. VideoSwap has demonstrated promising results in swapping the subject in the source video with a target concept that may have a different shape. In our paper, we focus on the swapping foreground subject, without considering background swapping or stylization. Further research could delve into a more general framework for video editing involving shape changes, thereby enhancing the flexibility of the video editing.

Application based on Customized Video Editing.

VideoSwap has shown promising results in swapping the subject in the source video with a target concept with customized identity. Future work may further investigate its application in movie generation and storytelling by fixing subjects' identities.

5.3. Potential Negative Social Impact

This project aims to provide the community with an effective method to swap their customized concept into the video. However, a risk exists wherein malicious entities could exploit this framework to create deceptive video with real-world figures, potentially misleading the public. This concern is not unique to our approach but rather a shared consideration in other concept customization methodologies. One potential solution to mitigate such risks involves adopting methods similar to anti-dreambooth [21], which introduce subtle noise perturbations to the published images to mislead the customization process. Additionally, applying unseen watermarking to the generated video could deter misuse and prevent them from being used without proper recognition.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 1

[2] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 1, 3

[3] 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. 1

[4] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 3

[5] Jiahui Huang, Leonid Sigal, Kwang Moo Yi, Oliver Wang, and Joon-Young Lee. Inve: Interactive neural video editing. *arXiv preprint arXiv:2307.07663*, 2023. 2, 5

[6] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 5

[7] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 2, 5

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 5

[9] 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. 3

[10] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022. 3

[11] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv preprint arXiv:2310.11440*, 2023. 3

[12] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 5

[13] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2

[14] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 5

[15] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 5

[16] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 3

[17] 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 preprint arXiv:2303.09535*, 2023. 3

[18] 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. 3

[19] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 5

[20] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 5

[21] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2116–2127, 2023. 6

[22] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 3

[23] Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, et al. Cvpr 2023 text guided video editing competition. *arXiv preprint arXiv:2310.16003*, 2023. 3

[24] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. *arXiv preprint arXiv:2306.07954*, 2023. 3

[25] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2