FlowVid: Taming Imperfect Optical Flows for Consistent Video-to-Video Synthesis

Feng Liang[*1], Bichen Wu[†2], Jialiang Wang[2], Licheng Yu[2], Kunpeng Li[2], Yinan Zhao[2], Ishan Misra[2], Jia-Bin Huang[2], Peizhao Zhang[2], Peter Vajda[2], Diana Marculescu[1]
[1]The University of Texas at Austin, [2]Meta GenAI
{jeffliang,dianam}@utexas.edu, {wbc,stzpz,vajdap}@meta.com
https://jeff-liangf.github.io/projects/flowvid

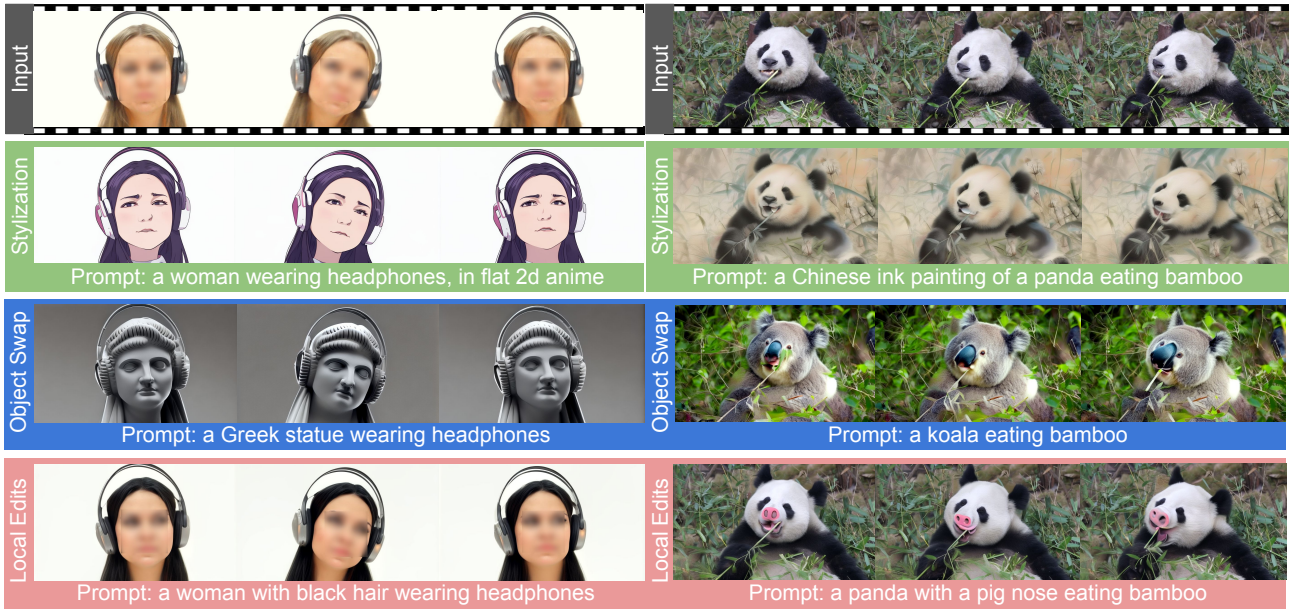


Figure 1. We present **FlowVid** to synthesize a consistent video given an input video and a target prompt. Our model supports multiple applications: (1) global stylization, such as converting the video to 2D anime (2) object swap, such as turning the panda into a koala bear (3) local edit, such as adding a pig nose to a panda.

Abstract

Diffusion models have transformed the image-to-image (I2I) synthesis and are now permeating into videos. However, the advancement of video-to-video (V2V) synthesis has been hampered by the challenge of maintaining temporal consistency across video frames. This paper proposes a consistent V2V synthesis framework by jointly leveraging spatial conditions and temporal optical flow clues within the source video. Contrary to prior methods that strictly adhere to optical flow, our approach harnesses its benefits while handling the imperfection in flow estimation. We encode the optical flow via warping from the first frame and serve it as a supplementary reference in the diffusion model. This enables our model for video synthesis by editing the first frame with any prevalent I2I models and then propagating edits to successive frames. Our V2V model, FlowVid, demonstrates remarkable properties: (1) Flexibility: FlowVid works seamlessly with existing I2I models, facilitating various modifications, including stylization, object swaps, and local edits. (2) Efficiency: Generation of a 4-second video with 30 FPS and 512×512 resolution takes only 1.5 minutes, which is 3.1×, 7.2×, and 10.5× faster than CoDeF, Rerender, and TokenFlow, respectively. (3) High-quality: In user studies, our FlowVid is preferred 45.7% of the time, outperforming CoDeF (3.5%), Rerender (10.2%), and TokenFlow (40.4%).

1. Introduction

Text-guided Video-to-video (V2V) synthesis, which aims to modify the input video according to given text prompts, has wide applications in various domains, such as short-video creation and more broadly in the film industry. Notable ad-

[*]Work partially done during an internship at Meta GenAI.
[†]Corresponding author.

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.00784

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.00784

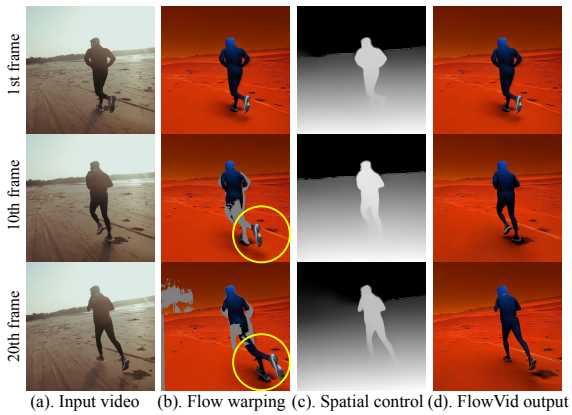


Figure 2. (a) Input video: `'a man is running on beach'`. (b) We edit the 1st frame with `'a man is running on Mars'`, then conduct flow warping from the 1st frame to the 10th and 20th frames (using input video flow). Flow estimation of legs is inaccurate. (c) Our FlowVid uses spatial controls to rectify the inaccurate flow. (d) Our consistent video synthesis results.

vancements have been seen in text-guided Image-to-Image (I2I) synthesis [4, 16, 33, 44], greatly supported by large pre-trained text-to-image diffusion models [38, 40, 41]. However, V2V synthesis remains a formidable task. In contrast to still images, videos encompass an added temporal dimension. Due to the ambiguity of text, there are countless ways to edit frames so they align with the target prompt. Consequently, naively applying I2I models on videos often produces unsatisfactory pixel flickering between frames.

To improve frame consistency, pioneering studies edit multiple frames jointly by inflating the image model with spatial-temporal attention [6, 27, 37, 48]. While these methods offer improvements, they do not fully attain the sought-after temporal consistency. This is because the motion within videos is merely retained in an *implicit* manner within the attention module. Furthermore, a growing body of research employs *explicit* optical flow guidance from videos. Specifically, flow is used to derive pixel correspondence, resulting in a pixel-wise mapping between two frames. The correspondence is later utilized to obtain occlusion masks for inpainting [21, 51] or to construct a canonical image [34] However, these hard constraints can be problematic if flow estimation is inaccurate, which is often observed when the flow is determined through a pre-trained model [43, 49, 50].

In this paper, we propose to harness the benefits of optical flow while handling the imperfection in flow estimation. Specifically, we perform flow warping from the first frame to subsequent frames. These warped frames are expected to follow the structure of the original frames but contain some occluded regions (marked as gray), as shown in Figure 2(b). If we use flow as hard constraints, such as inpainting [21, 51] the occluded regions, the inaccurate legs estimation would persist, leading to an undesirable outcome. We seek to include an additional spatial condition, such as a depth map in Figure 2(c), along with a temporal flow condition. The legs' position is correct in spatial conditions, and therefore, the joint spatial-temporal condition would rectify the imperfect optical flow, resulting in consistent results in Figure 2(d).

We build a video diffusion model upon an inflated spatial controlled I2I model. We train the model to predict the input video using spatial conditions (*e.g.*, depth maps) and temporal conditions (flow-warped video). During generation, we employ an *edit-propagate* procedure: (1) Edit the first frame with prevalent I2I models. (2) Propagate the edits throughout the video using our trained model. The decoupled design allows us to adopt an autoregressive mechanism: the current batch's last frame can be the next batch's first frame, allowing us to generate lengthy videos.

We train our model with 100k real videos from ShutterStock [1], and it generalizes well to different types of modifications, such as stylization, object swaps, and local edits, as seen in Figure 1. Compared with existing V2V methods, our FlowVid demonstrates significant advantages in terms of efficiency and quality. Our FlowVid can generate 120 frames (4 seconds at 30 FPS) in high-resolution (512×512) in just 1.5 minutes on one A-100 GPU, which is 3.1×, 7.2× and 10.5× faster than state-of-the-art methods CoDeF [34] (4.6 minutes) Rerender [51] (10.8 minutes), and TokenFlow [15] (15.8 minutes). We conducted a user study on 25 DAVIS [36] videos and designed 115 prompts. Results show that our method is more robust and achieves a preference rate of 45.7% compared to CoDeF (3.5%) Rerender (10.2%) and TokenFlow (40.4%)

Our contributions are summarized as follows: (1) We introduce FlowVid, a V2V synthesis method that harnesses the benefits of optical flow, while delicately handling the imperfection in flow estimation. (2) Our decoupled edit-propagate design supports multiple applications, including stylization, object swap, and local editing. Furthermore, it empowers us to generate lengthy videos via autoregressive evaluation. (3) Large-scale human evaluation indicates the efficiency and high generation quality of FlowVid.

2. Related Work

Image-to-image Diffusion Models Benefiting from large-scale pre-trained text-to-image (T2I) diffusion models [2, 13, 40, 41], progress has been made in text-based image-to-image (I2I) generation [12, 16, 26, 32, 33, 35, 44, 53]. Beginning with image editing methods, Prompt-to-prompt [16] and PNP [44] manipulate the attentions in the diffusion process to edit images according to target prompts. Instruct-pix2pix [4] goes a step further by training an I2I model that can directly interpret and follow human instructions. More recently, I2I methods have extended user control by allowing the inclusion of reference images to precisely define target image compositions. Notably, ControlNet, T2I-Adapter [33], and Composer [22] have introduced spatial conditions, such

as depth maps, enabling generated images to replicate the structure of the reference. Our method falls into this category as we aim to generate a new video while incorporating the spatial composition in the original one. However, it's important to note that simply applying these I2I methods to individual video frames can yield unsatisfactory results due to the inherent challenge of maintaining consistency across independently generated frames (per-frame results can be found in Section 5.2).

Video-to-video Diffusion Models To jointly generate coherent multiple frames, it is now a common standard to inflate image models to video: replacing spatial-only attention with spatial-temporal attention. For instance, Tune-A-Video [48], Vid-to-vid zero [45], Text2video-zero [27], Pix2Video [6], FateZero [37] and Fairy [47] perform cross-frame attention of each frame on anchor frame, usually the first frame and the previous frame to preserve appearance consistency. TokenFlow [15] further explicitly enforces semantic correspondences of diffusion features across frames to improve consistency. Furthermore, more works are adding spatial controls, *e.g.*, depth map to constraint the generation. ControlVideo [52] from Zhang *et al.* proposes to extend image-based ControlNet to the video domain with full cross-frame attention. Gen-1 [14], VideoComposer [46], Control-A-Video [8] and ControlVideo [54] from Zhao *et al.* train V2V models with paired spatial controls and video data. Our method falls in the same category but it also includes the imperfect temporal flow information into the training process alongside spatial controls. This addition enhances the overall robustness and adaptability of our method.

Another line of work is representing video as 2D images, as seen in methods like layered atlas [25], Text2Live [3], shape-aware-edit [28], StableVideo [7] and CoDeF [34]. However, these methods often require per-video optimization and they also face performance degradation when dealing with large motion, which challenges the creation of image representations.

Optical flow for video-to-video synthesis The use of optical flow to propagate edits across frames has been explored even before the advent of diffusion models, as demonstrated by the well-known Ebsythn [24] approach. In the era of diffusion models, Video ControlNet [10] from Chu *et al.* employs the ground-truth (gt) optical flow from synthetic videos to enforce temporal consistency among corresponding pixels across frames. However, it's important to note that ground-truth flow is typically unavailable in real-world videos, where flow is commonly estimated using pretrained models [43, 49, 50]. Recent methods like Rerender [51], MeDM [9], and VideoControlNet [21] from hu *et al.* use estimated flow to generate occlusion masks for in-painting. In other words, these methods "force" the overlapped regions to remain consistent based on flow estimates. Similarly, CoDeF [34] utilizes flow to guide the generation of canonical images. FLATTEN [11] enforces the patches on the same flow path across different frames to attend to each other in the attention module. These approaches all assume that flow could be treated as an accurate supervision signal that must be strictly adhered to. In contrast, our FlowVid recognizes the imperfections inherent in flow estimation and presents an approach that leverages its potential without imposing rigid constraints.

3. Preliminary

Latent Diffusion Models Denoising Diffusion Probabilistic Models (DDPM) [18] generate images through a progressive noise removal process applied to an initial Gaussian noise, carried out for T time steps. Latent Diffusion models [40] conduct diffusion process in latent space to make it more efficient. Specifically, an encoder $\mathcal{E}$ compresses an image $I \in \mathbb{R}^{H\times W\times 3}$ to a low-resolution latent code $z = \mathcal{E}(I) \in \mathbb{R}^{H/8\times W/8\times 4}$. Given $z_0 := z$, the Gaussian noise is gradually added on z_0 with time step t to get noisy sample z_t. Text prompt τ is also a commonly used condition. A time-conditional U-Net ϵ_θ is trained to reverse the process with the loss function:

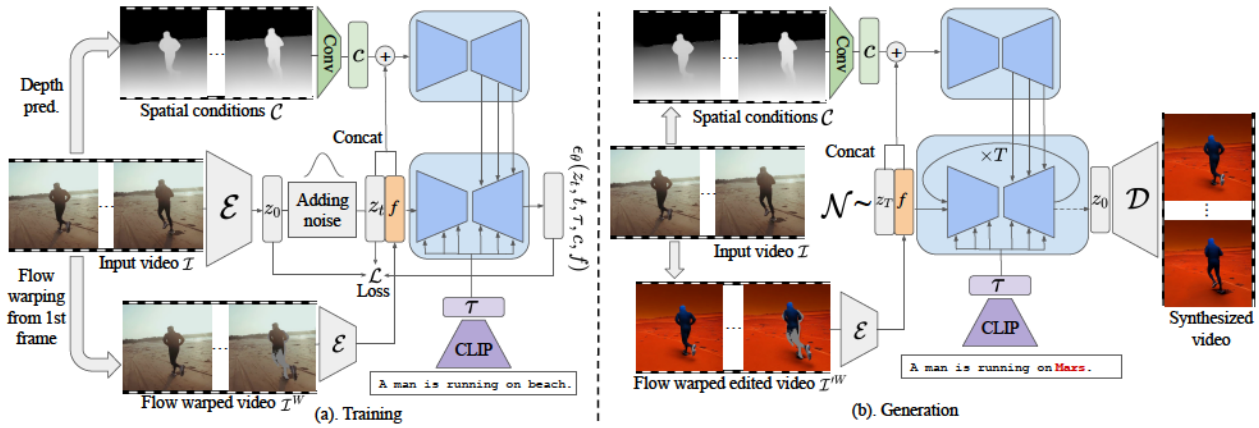


Figure 3. **Overview of our FlowVid.** (a) Training: we first get the spatial conditions (predicted depth maps) and estimated optical flow from the input video. For all frames, we use flow to perform warping from the first frame. The resulting flow-warped video is expected to have a similar structure as the input video but with some occluded regions (marked as gray, better zoomed in). We train a video diffusion model with spatial conditions c and flow information f. (b) Generation: we edit the first frame with existing I2I models and use the flow in the input video to get the flow warped edited video. The flow condition spatial condition jointly guides the output video synthesis.

decoder blocks. Each block has two components: a residual convolutional module and a transformer module. The transformer module, in particular, comprises a spatial self-attention layer, a cross-attention layer, and a feed-forward network. To extend the U-Net architecture to accommodate an additional temporal dimension, we first modify all the 2D layers within the convolutional module to pseudo-3D layers, *i.e.*, replacing 3×3 kernels with $1\times3\times3$, and add an extra-temporal self-attention layer [20]. Following common practice [6, 20, 27, 37, 48], we further adapt the spatial self-attention layer to a spatial-temporal self-attention layer. For video frame I_i, the attention matrix would take the information from the first frame I_1 and the previous frame I_{i-1}. Specifically, we obtain the query feature from frame I_i, while getting the key and value features from I_1 and I_{i-1}. The $\text{Attention}(Q, K, V)$ of spatial-temporal self-attention could be written as

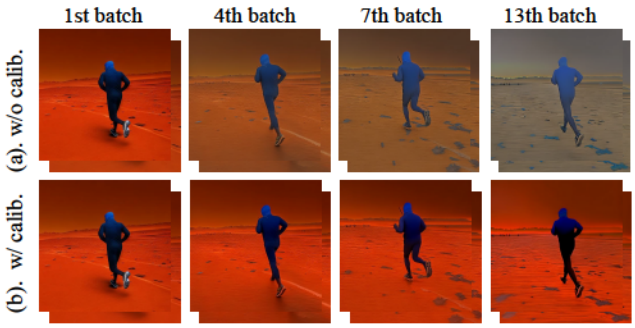


Figure 4. **Effect of color calibration in autoregressive evaluation.** (a) When the autoregressive evaluation goes from the 1st batch to the 13th batch, the results without color calibration become gray. (b) The results are more stable with the proposed color calibration.

used ϵ-parameterization. This finding is consistent with other video diffusion models, such as Gen-1 [14] and Imagen Video [19] (see ablation in Section **??**). Second, incorporating additional elements beyond the flow-warped video would further improve the performance. Specifically, including the first frame as a constant video sequence, $\mathcal{I}^{1st} = \{I_1, \ldots, I_1\}$, and integrating the occlusion masks $\mathcal{O} = \{O^{bwd}_{1\to1}, \ldots, O^{bwd}_{N\to1}\}$ enhanced the overall output quality. We process $\mathcal{I}^{1st}$ by transforming it into a latent representation and then concatenating it with the noisy latent, similar to processing $\mathcal{I}^W$. For $\mathcal{O}$, we resize the binary mask to match the latent size before concatenating it with the noisy latent. Further study is included in Section 5.4.

4.3. Generation: edit the first frame then propagate

During the generation, we want to transfer the input video $\mathcal{I}$ to a new video $\mathcal{I}'$ with the target prompt τ'. To effectively leverage the prevalent I2I models, we adopt an edit-propagate method. This begins with editing the first frame I_1 using I2I models, resulting in an edited first frame I'_1. We then propagate the edits to subsequent i^{th} frame by using the flow $\mathcal{F}_{i\to1}$ and the occlusion mask $O^{bwd}_{i\to1}$, derived from the input video $\mathcal{I}$. This process yields the flow-warped edited video $\mathcal{I}'^W = \{I'^W_1, \ldots, I'^W_N\}$. We input $\mathcal{I}'^W$ into the same encoder $\mathcal{E}$ and concatenate the resulting flow latent f with a randomly initialized Gaussian noise z_T drawn from the normal distribution $\mathcal{N}(0, 1)$. The spatial conditions from the input video are also used to guide the structural layout of the synthesized video. Intuitively, the flow-warped edited video serves as a texture reference while spatial controls regularize the generation, especially when we have inaccurate flow. After DDIM denoising, the denoised latent z_0 is brought back to pixel space with a decoder $\mathcal{D}$ to get the final output.

In addition to offering the flexibility to select I2I models for initial frame edits, our model is inherently capable of producing extended video clips in an autoregressive manner. Once the first N edited frames $\{I'_1, \ldots, I'_N\}$ are generated, the N^{th} frame I'_N can be used as the starting point for editing the subsequent batch of frames $\{I_N, \ldots, I_{2N-1}\}$. However, a straightforward autoregressive approach may lead to a *grayish* effect, where the generated images progressively become grayer, see Figure 4(a). We believe this is a consequence of the lossy nature of the encoder and decoder, a phenomenon also noted in Rerender [51]. To mitigate this issue, we introduce a simple global color calibration technique that effectively reduces the graying effect. Specifically, for each frame I'_j in the generated sequence $\{I'_1, \ldots, I'_{M(N-1)+1}\}$, where M is the number of autoregressive batches, we calibrate its mean and variance to match those of I'_1. The effect of calibration is shown in Figure 4(b), where the global color is preserved across autoregressive batches.

(a). Prompt: `a pirate is rowing a boat on a lake.` (b). Prompt: `a oil painting of a tiger walking.`

Figure 5. **Qualitative comparison with representative V2V models.** Our method stands out in terms of prompt alignment and overall video quality. We highly encourage readers to refer to video comparisons in our supplementary videos.

guidance [17] with a scale of 7.5 and use 20 inference sampling steps. Additionally, the Zero SNR noise scheduler [29] is utilized. We also fuse the self-attention features obtained during the DDIM inversion of corresponding key frames from the input video, following FateZero [37]. We evaluate our FlowVid with two different spatial conditions: canny edge maps [5] and depth maps [39]. A comparison of these controls can be found in Section 5.4.

Evaluation We select the 25 object-centric videos from the public DAVIS dataset [36], covering humans, animals, *etc*. We manually design 115 prompts for these videos, spanning from stylization to object swap. We conduct both qualitative (see Section 5.2) and quantitative comparisons (see Section 5.3) with state-of-the-art methods including Rerender [51], CoDeF [34] and TokenFlow [15]. We use their official codes with the default settings.

5.2. Qualitative results

In Figure 5, we qualitatively compare our method with several representative approaches. Starting with a per-frame baseline directly applying I2I models, ControlNet, to each frame. Despite using a fixed random seed, this baseline often results in noticeable flickering, such as in the man's clothing and the tiger's fur. CoDeF [34] produces outputs with significant blurriness when motion is big in input video, evident in areas like the man's hands and the tiger's face. Rerender [51] often fails to capture large motions, such as the movement of paddles in the left example. Also, the color of the edited tiger's legs tends to blend in with the background. TokenFlow [15] occasionally struggles to follow the prompt, such as transforming the man into a pirate in the left example. It also erroneously depicts the tiger with two legs for the first frame in the right example, leading to flickering in the output video. In contrast, our method stands out in terms of editing capabilities and overall video quality, demonstrating superior

Table 1. **Quantitative comparison with existing V2V models** The preference rate indicates the frequency the method is preferred among all the four methods in human evaluation. Runtime shows the time to synthesize a 4-second video with 512×512 resolution on one A-100-80GB. Cost is normalized with our method.

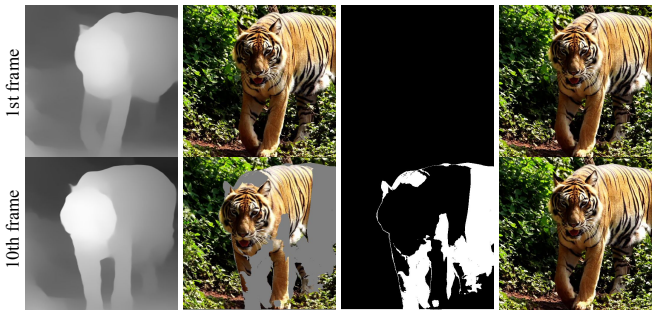
	Preference rate (mean ± std %) ↑	Runtime (mins) ↓	Cost ↓
TokenFlow	40.4 ± 5.3	15.8	10.5 ×
Rerender	10.2 ± 7.1	10.8	7.2 ×
CoDeF	3.5 ± 1.9	4.6	3.1 ×
FlowVid (Ours)	**45.7** ± 6.4	**1.5**	**1.0** ×

performance over these methods. We highly encourage readers to refer to more video comparisons in our supplementary videos.

5.3. Quantitative results

User study We conducted a human evaluation to compare our method with three notable works: CoDeF [34], Rerender [51], and TokenFlow [15]. The user study involves 25 DAVIS videos and 115 manually designed prompts. Participants are shown four videos and asked to identify which one has the best quality, considering both temporal consistency and text alignment. The results, including the average preference rate and standard deviation from five participants for all methods, are detailed in Table 1. Our method achieved a preference rate of 45.7%, outperforming CoDeF (3.5%) Rerender (10.2%), and TokenFlow (40.4%). During the evaluation, we observed that CoDeF struggles with significant motion in videos. The blurry constructed canonical images would always lead to unsatisfactory results. Rerender occasionally experiences color shifts and bright flickering. TokenFlow sometimes fails to sufficiently alter the video according to the prompt, resulting in an output similar to the original video. We release all user study videos on our project page.

Pipeline runtime We also compare runtime efficiency with existing methods in Table 1. Video lengths can vary resulting in different processing times. Here, we use a video containing 120 frames (4 seconds video with FPS of 30) The resolution is set to 512 × 512. Both our FlowVid model and Rerender [51] use a key frame interval of 4. We generate 31 keyframes by applying autoregressive evaluation twice followed by RIFE [23] for interpolating the non-key frames. The total runtime, including image processing, model operation, and frame interpolation, is approximately 1.5 minutes. This is significantly faster than CoDeF (4.6 minutes), Rerender (10.8 minutes) and TokenFlow (15.8 minutes), being 3.1×, 7.2×, and 10.5 × faster, respectively. CoDeF requires per-video optimization to construct the canonical image. While Rerender adopts a sequential method, generating each frame one after the other, our model utilizes batch processing, allowing for more efficient handling of multiple frames simultaneously. In the case of TokenFlow, it requires a large number of DDIM inversion steps (typically around 500) for all frames to obtain the inverted latent, which is a resource-intensive process. We further report the runtime breakdown (Figure **??**) in the Appendix.



(a) Condition types.

Condition choices				Winning rate ↑
(I)	(II)	(III)	(IV)	
✓	×	×	×	9%
✓	✓	×	×	38%
✓	✓	✓	×	42 %

(b) Winning rate over our FlowVid (I + II + III + IV).

Figure 6. **Ablation study of condition combinations.** (a) Four types of conditions. (b) The different combinations all underperform our final setting which combines all four conditions.

5.4. Ablation study

Condition combinations We study the four types of conditions in Figure 6(a): (I) Spatial controls: such as depth maps [39]. (II) Flow warped video: frames warped from the first frame using optical flow. (III) Flow occlusion: masks indicate which parts are occluded (marked as white). (IV) First frame. We evaluate combinations of these conditions in Figure 6(b), assessing their effectiveness by their winning rate against our full model which contains all four conditions. The spatial-only condition achieved a 9% winning rate, limited by its lack of temporal information. Including flow warped video significantly improved the winning rate to 38%, underscoring the importance of temporal guidance. We use gray pixels to indicate occluded areas, which might blend in with the original gray colors in the images. To avoid potential confusion, we further include a binary flow occlusion mask, which better helps the model to tell which part is occluded or not. The winning rate is further improved to 42%. Finally, we added the first frame condition to provide better texture guidance, particularly useful when the occlusion mask is large and few original pixels remain.

(a). Input frame (b). Spatial control (c). Style transfer (d). Objec

Figure 7. **Ablation study of different spatial conditions.** edge and depth map are estimated from the input frame. edge provides more detailed controls (good for stylization depth map provides more editing flexibility (good for objec

Table 2. **Ablation study of optical flow quality.** User pre (%) with different flow models.

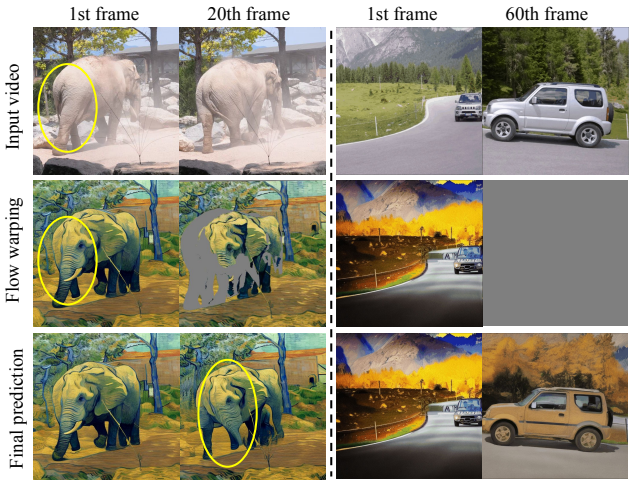
UniMatch wins	Draw	RAFT wins
28.7	45.2	26.1

Different control type: edge and depth We study two types of spatial conditions in our FlowVid: canny edge [5] and depth map [39]. Given an input frame as shown in Figure 7(a), the canny edge retains more details than the depth map, as seen from the eyes and mouth of the panda. The strength of spatial control would, in turn, affect the video editing. For style transfer prompt `'A Chinese ink painting of a panda eating bamboo'`, as shown in Figure 7(c), the output of canny condition could keep the mouth of the panda in the right position while the depth condition would guess where the mouth is and result in an open mouth. The flexibility of the depth map, however, would be beneficial if we are doing object swap with prompt `'A koala eating bamboo'`, as shown in Figure 7(d); the canny edge would put a pair of panda eyes on the face of the koala due to the strong control, while depth map would result in a better koala edit. During our evaluation, we found canny edge works better when we want to keep the structure of the input video as much as possible, such as stylization. The depth map works better if we have a larger scene change, such as an object swap, which requires more considerable editing flexibility.

Optical flow quality. We study the effect of the flow quality. We replaced UniMatch [50] with the lower-quality RAFT [43] to assess preferences, as shown in Table 2. The similar preferences observed for different flow models indicate FlowVid's robustness to flow quality.

5.5. Limitations

Although our FlowVid achieves significant performance, it does have some limitations. First, our FlowVid heavily relies on the first frame generation, which should be structurally aligned with the input frame. As shown in Figure 8(a), the edited first frame identifies the hind legs of the elephant as the front nose. The erroneous nose would propagate to the following frame and result in an unsatisfactory final prediction. The other challenge is when the camera or the object moves so fast that large occlusions occur. In this case, our model would guess, sometimes hallucinate, the missing blank regions. As shown in Figure 8(b), the color of the car changes when we don't have any reference in the flow warping. However, FlowVid doesn't completely fail because we still have spatial conditions and the first frame as the reference. Additionally, FlowVid can adjust the keyframe interval to manage occlusions effectively.



(a). Misaligned first frame (b). Big occlusion

Figure 8. **Limitations of FlowVid.** Failure cases include (a) the edited first frame doesn't align structurally with the original first frame, and (b) large occlusions caused by fast motion.

6. Conclusion

In this paper, we propose a consistent video-to-video synthesis method using joint spatial-temporal conditions. In contrast to prior methods that strictly adhere to optical flow, our approach incorporates flow as a supplementary reference in synergy with spatial conditions. Our model can adapt existing image-to-image models to edit the first frame and propagate the edits to consecutive frames. Our model is also able to generate lengthy videos via autoregressive evaluation. Both qualitative and quantitative comparisons with current methods highlight the efficiency and high quality of our proposed techniques.

Acknowledgments

This research was supported in part by ONR Minerva program, iMAGiNE - the Intelligent Machine Engineering Consortium at UT Austin, and a UT Cockrell School of Engineering Doctoral Fellowship.

References

[1] Stock footage video, royalty-free hd, 4k video clips, 2023. 2, 5

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[3] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022. 3

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 3, 6, 8

[6] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23206–23217, 2023. 2, 3, 4

[7] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023. 3

[8] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 3

[9] Ernie Chu, Tzuhsuan Huang, Shuo-Yen Lin, and Jun-Cheng Chen. Medm: Mediating image diffusion models for video-to-video translation with temporal correspondence guidance. *arXiv preprint arXiv:2308.10079*, 2023. 3

[10] Ernie Chu, Shuo-Yen Lin, and Jun-Cheng Chen. Video controlnet: Towards temporally consistent synthetic-to-real video translation using conditional image diffusion models. *arXiv preprint arXiv:2305.19193*, 2023. 3

[11] 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. 3

[12] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[13] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 2

[14] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023. 3, 5

[15] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 2, 3, 5, 6, 7

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

[17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[19] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 5

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 4

[21] Zhihao Hu and Dong Xu. Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet. *arXiv preprint arXiv:2307.14073*, 2023. 2, 3

[22] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023. 2

[23] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 5, 7

[24] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Stylizing video by example. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019. 3

[25] 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. 3

[26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[27] 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. 2, 3, 4

[28] Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14317–14326, 2023. 3

[29] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023. 6

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[31] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[32] 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. *arXiv preprint arXiv:2108.01073*, 2021. 2

[33] 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

[34] 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. *arXiv preprint arXiv:2308.07926*, 2023. 2, 3, 4, 6, 7

[35] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2, 6

[37] 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. 2, 3, 4, 5, 6

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3, 2022. 2

[39] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3, 6, 7, 8

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[42] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 4, 5

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2, 3, 8

[44] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 2

[45] Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023. 3

[46] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023. 3

[47] Bichen Wu, Ching-Yao Chuang, Xiaoyan Wang, Yichen Jia, Kapil Krishnakumar, Tong Xiao, Feng Liang, Licheng Yu, and Peter Vajda. Fairy: Fast parallelized instruction-guided video-to-video synthesis. *arXiv preprint arXiv:2312.13834*, 2023. 3

[48] 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. 2, 3, 4

[49] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 2, 3

[50] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 3, 4, 8

[51] 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. 2, 3, 4, 5, 6, 7

[52] 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. 3

[53] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023. 2

[54] Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. Controlvideo: Adding conditional control for one shot text-to-video editing. *arXiv preprint arXiv:2305.17098*, 2023. 3