VidToMe: Video Token Merging for Zero-Shot Video Editing

Supplementary Material

A. Video Results

Our project page is available at https://vidtome-diffusion.github.io/. We present more video editing results in the supplementary video "Results.mp4", including qualitative comparison with prior methods and sample editing results on various videos. We summarize our work in another supplementary video at https://www.youtube.com/watch?v=cZPtwcRepNY.

B. Metrics

We explain the metrics used in quantitative evaluation, including Interpolation Error and PSNR [3], Warp Error, Frame CLIP Score, Directional CLIP Score [1], Text CLIP Score, and User Preference Rate. CLIP score metrics are computed in the feature space of the CLIP model [6] for both prompts and frames. Others estimate the optical flow [8] to measure the video continuity.

Warp Error. Previous works [4] use Warp Err to measure the pixel-level video continuity. It is obtained by warping the edited video frames to adjacent frames by the optical flow estimated on the source video and computing the average mean-squared pixel error between warped and target frames.

Interpolation Error and PSNR. Since Warp Error utilizes the source video to estimate the optical flow, it reflects whether the edited video precisely matches the source video in motion. To measure the video continuity independently, we proposed interpolation-based metrics. Following video interpolation works [3], we interpolate a target frame by its previous and next frames and compute the Interpolation Error and PSNR between the interpolated frame and the target frame, where the error is defined as root-mean-squared (RMS) difference between the two frames.

Frame CLIP Score. Frame CLIP Score is the average CLIP similarity between consecutive frames in the generated video, measuring the video consistency in the CLIP feature space.

Text CLIP Score. Text CLIP Score is the average CLIP similarity between the edit prompt and the edited frames. However, it is not enough to measure the edit performance with the Text CLIP Score. For example, we can directly generate frames with the edit prompt, omitting the source frames. The resulting frames probably achieve a higher Text CLIP Score than the edited frames, though they are not correlated to source frames.

Directional CLIP Score. Compared to the consistency between the prompt and frames, it is more important for the editing task to measure the consistency between their changes from source to edit, *i.e.*, whether the change in prompt matches the change in video frames. Therefore, we use the directional CLIP Score [1] to measure the editing effect more precisely, which is the cosine similarity in CLIP space between the difference between the source and edit prompts and the difference between the source and edited frames.

User Preference Rate. We conduct user studies to evaluate performance in terms of human perception. Users choose their favorite one among the editing results of baselines and our method. Each survey consists of 10 videos, and a total of 27 survey results are collected. User Preference Rate is the average rate of a method preferred by users.

	Interpolation Error ↓	Directional CLIP Score ↑
Rerender-A-Video [10]	0.116	0.091
TokenFlow [2]	0.131	0.108
ControlVideo [12]	0.127	0.147
Ours	0.111	0.140

Table 1. Performance Comparison with more video editing works. All use SDv1.5.

C. More Comparison Results

We compare our proposed VidToMe with more works in Tab. 1. VidToMe achieves comparable performance on temporal consistency and text alignment against these competitive methods. Rerender-A-Video [10] achieves similar interpolation errors as VidToMe, but suffers from low text alignment. TokenFlow [2] generates high-quality editing results while it does not perform favorably in terms of evaluation metrics. ControlVideo [12] achieves a high directional CLIP score with random initial noise. However, using the same noise for each sample causes an unnatural freeze-background effect on generated videos. We do not compare to Fate/Zero [5] since it requires adjusting its word-level attention blending module for each sample, which is time-consuming and cannot be finished within the rebuttal period. These results will be included in our revised version.

D. Implementation Details

Given a source video, we invert video frames into noise latent by DDIM inversion with a text-to-image latent diffusion model, Stable Diffusion [7]. A source prompt is provided as the text condition in inversion. Then we generate the edited video frames with the same diffusion model using an edit prompt as the text condition. Both inversion and generation use the DDIM scheduler with sampling step 50.

For the evaluation results, our method keeps video chunk size $B = 4$, local and global merging ratio $p = 0.9, 0.8$, and a fixed random seed. The hyperparameters are tuned for sample results. The video token merging is applied in the first two downsampling layers and the last two upsampling layers in the diffusion model, right before the self-attention module.

E. Details of Global Token Merging

There are two factors related to our global token merging performance, the order to process video chunks and the src, dst assignment in global token merging.

Chunk Processing Order. In each denoising iteration, video frames are split into consecutive video chunks. The order to process the chunks is related to the global token updating behavior, as global tokens are maintained across chunks. One option is to process the chunks in sequential order. The global tokens are shared among near chunks, boosting the video consistency in consecutive frames. However, distant video frames are still not likely to share tokens as the global tokens are updated gradually. Another choice is to process the chunks in fully random order. The global tokens are randomly shared between chunks independent of the temporal order, promoting global consistency among all video frames. However, tokens from distant frames are less correlated to the current frame, sometimes resulting in quality degradation. We can also combine the two choices to process part of the chunks in random order and the others sequentially, balancing their effect.

Random Global Token Updating. In global token merging, local tokens and global tokens are merged to T_{gm}. Global tokens are then updated to the local tokens unmerged from T_{gm}. Since we use the values of dst tokens as the merged token values, the updated global tokens $T_g^{'}$ are close to the dst tokens. If local tokens (src) are merged to the global tokens (dst), $T_g^{'}$ consists of most original global tokens and a few new local tokens. Otherwise $T_g^{'}$ has most of its tokens from the current frame chunk. We find that always merging local tokens with global tokens degrades the video quality in some cases since most frames share the same global tokens, overcompressing the video in the feature space. Therefore, we randomly assign dst, src to local and global tokens in the global token merging so that the tokens are properly shared among video chunks. In evaluation, we use random chunk order and assign dst to local tokens with probability 0.5.

F. Details of Controlling Methods

Our method combines an existing controlling method for image editing to maintain the source frame structure. In this work, we apply Plug-and-Play (PnP) [9], ControlNet [11].

PnP. As PnP injects the self-attention map from source frames to edit frames, their tokens should be aligned. However, token merging may combine different tokens in source and edit frames as the similarity-based matching. To keep the token alignment between source and edit, we enforce their matching map to be the same where the token matching follows the one with a larger similarity in source or edit.

ControlNet. When combined with ControlNet, the diffusion model may generate over-saturated frames with the DDIM inverted initial noise. We propose to solve the problem by controlled DDIM inversion where the ControlNet is applied in both the inversion and generation process. It ensures the frame can be reconstructed with the source prompt when generated with the same ControlNet as inversion.

G. Algorithm.

To clarify our method, we provide the pseudocode of the VidToMe algorithm (Algo. 1). Readers can refer to it for more algorithm details.

References

[1] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 1

[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. 1

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018. 1

[4] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, pages 170–185, 2018. 1

[5] 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:2303.09535*, 2023. 1

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 3

[8] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 1

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

Algorithm 1: VidToMe

Data: $V = (z_1, z_2, \dots, z_n)$: Source video latents with n frames. $\mathcal{P}_{src}, \mathcal{P}_{edit}$: Source prompt and edit prompt. ϵ_θ: Pretrained text-to-image diffusion model. G: Existing controlling method [7, 9, 11].
Result: V^*: Edited video.
Hyperparameters: Chunk Size B, local and global merging ratio p_l, p_g, chunk processing order O, merge-to-local probability q

$c_{src}, c_{edit} \leftarrow$ TextEncoder$(\mathcal{P}_{src}, \mathcal{P}_{edit})$; `// Encode text prompts to feature space.`
$V^T = (z_1^T, z_2^T, \dots, z_n^T) \leftarrow$ DDIM-Inversion$(V, c_{src}, \epsilon_\theta)$; `// Invert frame into noise latent.`
for t:$T \rightarrow 1$ **do**
  $C \leftarrow$ Chunk(V^t, B) ; `// Split video into chunks.`
  $C = (C_1, C_2, \dots, C_m) \leftarrow$ Perm(C, O) ; `// Permute chunks.`
  for i: $1 \rightarrow m$ **do**
    $\epsilon_i \leftarrow \epsilon_\theta(C_i, t, c_{edit}; G)$; `// Estimate noise direction using diffusion model with video token merging.`
  end
  $\epsilon \leftarrow (\epsilon_1, \epsilon_2, \cdots, \epsilon_m)$;
  $V^{t-1} \leftarrow$ Denoise(V^t, ϵ, t);
end
$V^* \leftarrow$ Decode(V^0) ; `// Decode latents to image.`
`// Perform video token merging inside the diffusion model.`
`// Before the self-attention modules.`
$T_{in} \leftarrow \{T_{in}^f\}_{f=0}^{B-1}$;
`// Local Token Merging`
$k \leftarrow$ RandInt$(0, B-1)$;
$r \leftarrow p_l(B-1)N$;
$E_l \leftarrow$ Match$(\{T_{in}^f\}_{f=0, f\neq k}^{B-1}, T_{in}^k, r))$;
$T_{lm} \leftarrow$ M(T_{in}, E_l);
`// Global Token Merging`
if $i == 1$ **then**
  $T_g \leftarrow T_{lm}$;
  $T_{gm} \leftarrow T_{lm}$; `// Initialize global tokens.`
else
  $r \leftarrow p_g(B-1)N$;
  if *Rand*$(0,1) < q$ **then**
    $E_g \leftarrow$ Match(T_g, T_{lm}, r);
  else
    $E_g \leftarrow$ Match(T_{lm}, T_g, r);
  end
  $T_{gm} \leftarrow$ M$(\{T_{lm}, T_g\}, E_g)$;
  $T_{lm}^{'}, T_g^{'} \leftarrow$ U(T_{gm}, E_g);
  $T_g \leftarrow T_{lm}^{'}$; `// Update global tokens.`
end
$T_o \leftarrow$ Self-Attention(T_{gm});
`// Token Unmerging`
$T_{local}, T_{global} \leftarrow$ U(T_o, E_g);
$T_{out} \leftarrow$ U(T_{local}, E_l);

[10] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH*, 2023. 1

[11] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 3

[12] 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. 1