FADE: Frequency-Aware Diffusion Model Factorization for Video Editing

Supplementary Material

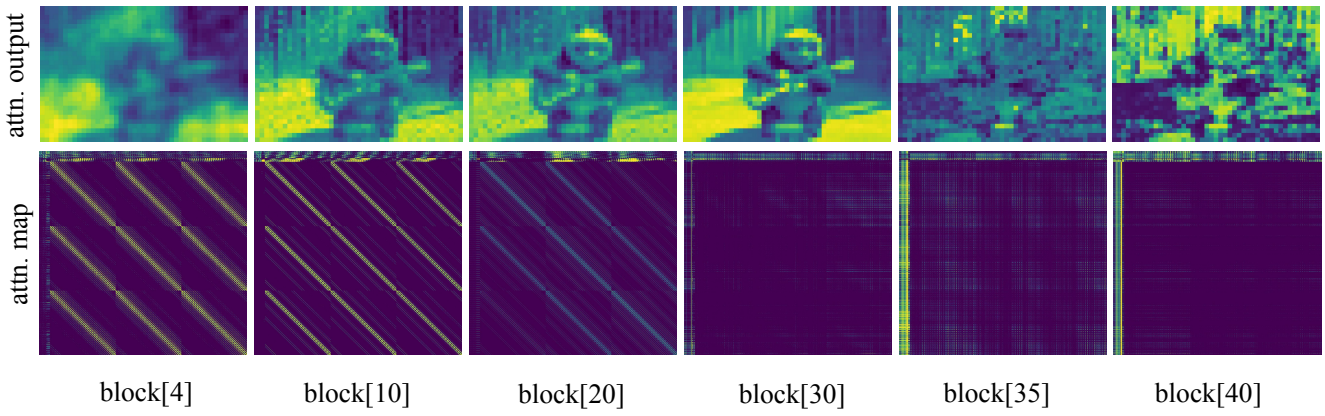

Figure A. **Visualization of attention outputs and attention maps.** We can clearly observe that the patterns of the attention features across blocks. Early blocks primarily process low-frequency structures, reflected in their blurred attention outputs and maps with multiple diagonal patterns. Conversely, later blocks focus on high-frequency details, producing attention maps with a uniformly distributed pattern.

A. Detailed Implementations

To leverage video generation priors, we employ the widely used T2V model, CogVideoX-5b [35], for video inversion and generation. In the inversion stage, the resolution of the input video is first adjusted to 720×480 to comply with the resolution constraint of CogVideoX-5b. Additionally, since the model is trained with long prompts, a chatbot is utilized to enrich the input prompt with additional details. Subsequently, DDIM inversion is performed with the total number of steps set to $T = 50$, yielding the latent trajectories. In the editing stage, the first four blocks are designated as sketching blocks. To capture spectral characteristics, we apply 3D DFT on the attention output of these blocks. Additionally, the output of the last block in DiT is utilized to compute an auxiliary guidance term. The guidance mechanism is applied during the interval $[0, 0.6T]$ of the DDIM sampling process. Furthermore, to isolate the target object from unrelated regions, we adopt the local editing trick [18] with a mask during the interval $[0, 0.8T]$. As CogVideoX-5b does not provide ideal masks, we instead utilize the method proposed in [38] to generate the necessary masks. The proposed approach is implemented on a single NVIDIA LS20 GPU with 48GB of VRAM, achieving an average processing time of approximately 4 minutes per video, making it suitable for near-real-time applications.

B. Different Functions of Attention Blocks

In this section, we delve deeper into the different functions of attention blocks in the T2V model. In Figure A, we visualize the attention output and attention map for six blocks out of a total of 42. As the block number increases, the attention output contains progressively more high-frequency components, indicating that earlier blocks establish the low-frequency, foundational structure of the video, such as object placement and movement, while later blocks focus on high-frequency refinements and details. Furthermore, the attention in earlier blocks is densely concentrated along diagonal lines, while the attention in later blocks becomes more evenly distributed. This further demonstrates that earlier blocks emphasize the key shapes and correspondences in the video, while later blocks focus on fine details. For video editing tasks, which require maintaining low-frequency features like object location and general shape while modifying high-frequency features, the best performance is achieved by using only the first four blocks, *i.e.*, the sketching blocks.

C. More Discussions

About the static background. The static background in the winter example is due to the limited prompt. As shown in Figure B, by explicitly adding '*moving forward*' to the prompt, we can generate a dynamically moving background.

Figure B. The generated video features a dynamic background, achieved by changing the prompt.

About the ablation results. Using all blocks and the entire spectrum for guidance can slightly improve the preservation (M.PSNR, SSIM) since it leverages more video features for reconstruction. However, it introduces two major issues: (1) Editing quality (CLIP) may degrade, as features in sharpening blocks often need not remain unchanged, and (2) The additional blocks increase inference time and GPU memory usage (67.6GB for all blocks), making it less feasible.

About the sensitivity of λ. Since we normalize the scale of the gradient in Eq.7, a reasonable range of λ is [10, 15], which consistently produces plausible results. As illustrated in Figure C, a larger λ improves preservation but slightly reduces textual alignment.

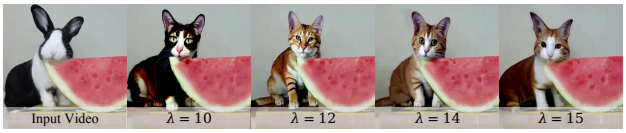

Figure C. The sensitivity of λ.

About computational cost. Our method is zero-shot and only involves sampling time (Table 2). With CPU offloading, our method uses 23.6GB GPU memory, comparable to Video-P2P (25.7GB), AnyV2V (17.1GB) and TokenFlow (11.4GB).

About the performance. We acknowledge that our method's quantitative improvement is not very significant compared with the existing methods, which heavily rely on *one-shot tuning* to enhance temporal consistency and adapt to the input sample. However, our approach is *zero-shot*. For a fair comparison, we found that fine-tuning our T2V model on the input sample significantly improves performance, as shown in Table A.

Table A. Quantitative comparisons under fine-tuning scenarios

Method	CLIP ↑	M.PSNR ↑	LPIPS ↓	OSV ↓	PF ↑
Tune-A-Video [34]	0.3522	19.86	0.4625	35.01	0.09
Video-P2P [15]	0.3589	20.57	0.3199	34.71	0.14
TokenFlow (ICLR'24)	0.3614	20.39	0.3212	34.50	0.12
FADE	0.3762	20.69	0.3085	31.36	0.27
FADE (tuned)	**0.3946**	**22.19**	**0.2937**	**30.27**	**0.38**

About video prior and block choice. In Figure D, We plotted the energy ratio of low and high-frequency components in the 3D DFT spectrum of the attention results at timestep=20 for all blocks. The frequency patterns align with the observations in Figure 2, supporting our motivation to distinguish between sketching and sharpening blocks. Moreover, this pattern remains consistent across sampling steps, as shown by the attention results in blocks [4] and [30] at timesteps $t = [10, 20, 30]$.

D. More Comparisons

In this section, we add qualitative results (Row 1&2) and comparisons with FLATTEN, TokenFlow, Rerender-a-Video, RAVE, and AnyV2V (Row 3), covering shape changes (sports car, duck), occlusion (desert, cat in the second figure of this PDF) and long video (taxi, >60 frames). Due to space limitations, we present one frame for comparison. As shown in Figure E, our method achieves competitive performance and textual alignment.

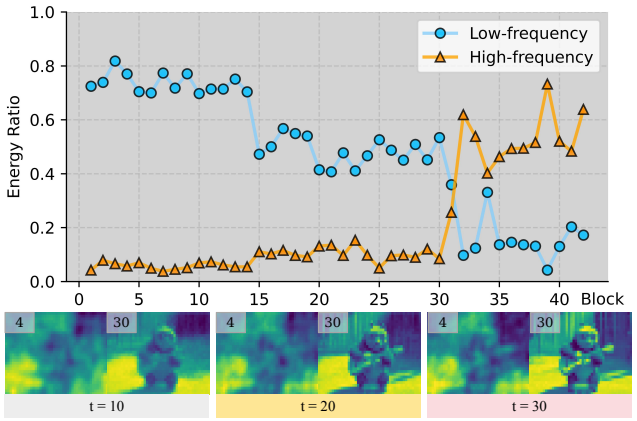

Figure D. The spectral energy distribution and visualization of the attention results.

Figure E. Qualitative results and comparisons

E. More Qualitative Results

We provide additional qualitative results to further demonstrate the effectiveness and versatility of our FADE framework across a variety of editing tasks. As shown in Figure F, our method produces highly realistic and coherent outcomes, even in challenging scenarios. For instance, in the first example, transforming "child" into "panda", FADE successfully handles complex motions, such as riding a bike, which requires accurately modeling significant object movements from far to close perspectives. This highlights the framework's ability to maintain spatial consistency during large-scale transformations. Another example, the transformation from "squirrel" to "robotic mouse", emphasizes FADE's flexibility in editing both textures and shapes. The results demonstrate the framework's capacity to adaptively adjust fine details while preserving overall coherence, enabling seamless and visually plausible edits.

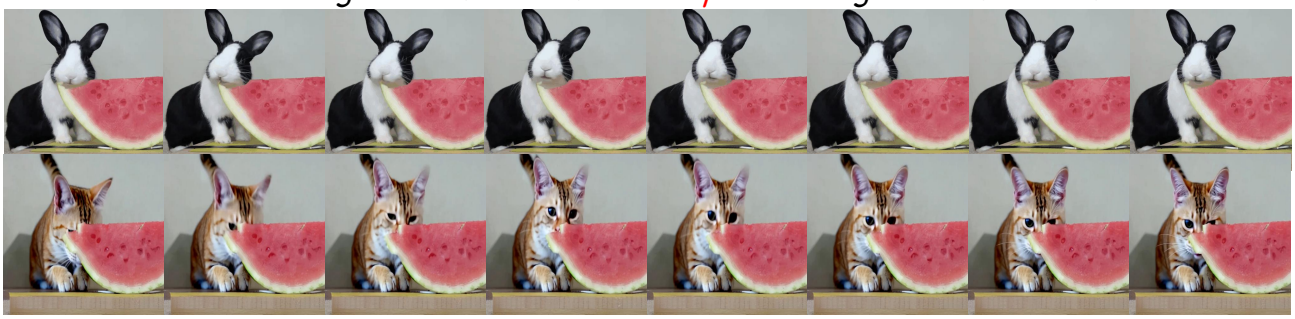

Figure F. **More Qualitative Results.** FADE demonstrates impressive performance across a range of video editing tasks.

F. Failure Cases

While our framework achieves impressive results across various editing tasks, it still faces limitations in handling certain complex scenarios. As illustrated in Figure G, our method occasionally produces artifacts, such as incorrect object orientation compared to the source video. For example, the dog that should be facing left in the edited output

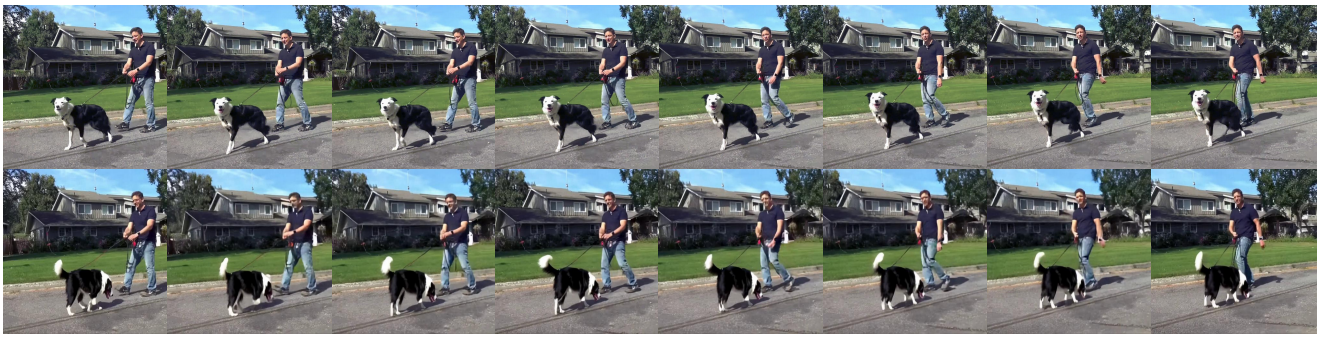

Figure G. **Failure Case.** Our framework may occasionally generates artifacts, such as incorrect object orientation.

incorrectly faces right. This issue arises from the absence of specific constraints to guide object placement, resulting in random variations in the generated object's orientation. Moreover, challenges also emerge when the edited objects are partially or fully occluded.