SmoothDiffusion-VE: Real-time Generative Video Editing Using Adaptive Feature Cache

Supplementary Material

A. Additional User Studies

We conduct additional user preference studies to compare our method with StreamDiffusion [21], StreamV2V [26], PNP [11], DMT [52], SDEdit [33], and T2V-Zero [20]. We adopt a Two-alternative Forced Choice (2AFC) protocol used in [11, 26, 51], where participants are shown two result videos and asked to identify which video has the best overall quality, best temporal consistency, and best alignment with the text prompt. For each comparison, feedback was gathered from at least sixteen different participants. We provide details on past user studies and their total participants in 5 to show we have more participants than past competing methods like StreamV2V [26] and FlowVid [27].

Table 5. **Total participants in user study of competing methods.**

Method	Total Participants
SmoothDiffusion-VE (Ours)	16
StreamV2V	3
FlowVid	5

The final preference rates of our method versus the competing methods in terms of temporal consistency are shown in Table 6.

Table 6. **User temporal consistency preference study.** The rate of the user preferring the competing method is denoted as Theirs Preferred. Tie indicates that both methods were equally preferred. Ours Preferred indicates our method was preferred to the competing method and is in bold. We color the results green when SmoothDiffusion-VE is preferred, and red when the alternative method is preferred.

Method	Theirs Preferred (%)	Tie (%)	Ours Preferred (%)
StreamDiffusion	2.3	21.1	**76.6**
DMT	49.2	2.4	**48.4**
PNP	12.9	14.1	**73.0**
SDEdit	11.7	13.7	**74.6**
T2V-Zero	8.9	4.8	**86.3**
StreamV2V	31.3	14.8	**53.9**

In terms of temporal consistency we find users favor our SmoothDiffusion-VE 48.4% of the time versus DMT [52], 76.6% of the time versus StreamDiffusion [21], 53.9% of the time versus StreamV2V [26], and 74.2% of the time versus PNP [48] as indicated in the ours preferred column of Table 6.

Table 7. **User prompt alignment preference study.** The rate of the user preferring the competing method is denoted as Theirs Preferred. Tie indicates that both methods were equally preferred. Ours Preferred indicates our method was preferred to the competing method and is in green. We color the results green when SmoothDiffusion-VE is preferred.

Method	Theirs Preferred (%)	Tie (%)	Ours Preferred (%)
StreamDiffusion	10.9	11.8	**77.3**
DMT	11.7	5.5	**82.8**
PNP	29.3	11.3	**59.4**
SDEdit	14.5	22.2	**63.3**
T2V-Zero	10.5	10.2	**79.3**
StreamV2V	14.9	28.9	**56.2**

In terms of prompt alignment we find users favor our SmoothDiffusion-VE 82.8% of the time versus DMT [52], 77.3% of the time versus StreamDiffusion [21], 56.2% of the time versus StreamV2V [26], 59.4% of the time versus PNP [48], and 63.3% of the time versus SDEdit [33] as indicated in the ours preferred column of Table 7.

B. Additional Ablation Studies

B.1. AFC vs. EFC vs. MFC

We show an ablation study showing the impacts of the Adaptive Feature Cache (AFC) compared to only having an extended-cache mode (EFC) and only having a mini-cache mode (MFC). This illustrates that the AFC switching between extended and mini cache mode preserves quality without sacrificing speed.

Table 8 shows that the MFC achieves the best speed, but with a worse CLIP score (97.5) and Warp Error (91.1). The AFC is only slightly slower (2.5 ms/frame slower), but is able to match the CLIP Score and Warp Error of the EFC with faster frame generation (9.8 ms/frame faster).

Table 8. **Ablation Study on AFC, EFC, and MFC.** We provide a comparison of CLIP score, warp error, and latency to show how interpolation, our AFC, EFC, and MFC compare. The final results of our method are in bold. A higher CLIP score indicates better results, while a lower warp error indicates better results.

Metric	Ours	EFC	MFC
CLIP Score ↑	**98.1**	98.1	97.5
Warp Error ↓	**88.1**	88.1	91.1
ms/frame ↓	**36**	45.8	33.5

B.2. Motion-Guided Attention

We perform an ablation study on our motion-guided attention. Table 9 shows that the motion-guided attention improves the frame generation speed by 6.1 ms/frame and also improves the CLIP Score and Warp Error.

Table 9. **Ablation Study on Motion-Guided Attention.** We provide a comparison of CLIP score, warp error, and latency to show how Motion-Guided Attention improves the temporal consistency and the frame generation speed of SmoothDiffusion-VE. A higher CLIP score indicates better results, while a lower warp error indicates better results.

Metric	Motion-Guided Attention	Regular Self-Attention
CLIP Score ↑	**97.3**	97.1
Warp Error ↓	**99.2**	100.5
ms/frame ↓	**55.2**	61.3

B.3. Adaptive Feature Cache Sizes

We ablate the effect of the size of the mini-cache and the extended-cache in the AFC in Table 10. The EFC size of 8 and the MFC size of 2 provide the best tradeoff of speed and quality as demonstrated with the low warp error and low frame generation speed of these cache sizes.

Table 10. **Effect of AFC cache size.** We report the average warp error over 20 text-video prompt pairs for different cache sizes for the extended-cache mode (EFC) and mini-cache mode (MFC) in the AFC. Our method's results are in bold.

EFC Size	MFC Size	Warp Error ↓	ms/frame ↓
8	**2**	**88.1**	**36**
16	4	88.0	46.5
4	1	92.3	34.4

B.4. LPIPS Threshold Sensitivity

We perform a sensitivity analysis on the LPIPS threshold to identify the optimal balance between performance and quality. As shown in Table 11, a lower threshold improves temporal consistency (lower Warp Error) at the cost of higher latency, while a higher threshold improves speed but degrades quality. Our default value of 0.3 is chosen as it provides a robust trade-off, maintaining high quality while being significantly faster than more sensitive settings.

C. Additional Runtime Comparisons

We provide additional runtime comparisons with SDEdit [33] and T2V-Zero [20] in Table 12, along with the original comparisons and speedups against StreamV2V [26], StreamDiffusion [21], PNP [48], and DMT [52].

Table 11. **Effect of LPIPS Threshold.** We report Warp Error and latency for different LPIPS threshold values. Our default of 0.3 offers the best balance of quality and speed.

LPIPS Threshold	Warp Error ↓	ms/frame ↓
0.1	88.0	39.6
0.2	88.1	38.4
0.3 (Ours)	**88.1**	**36.0**
0.5	89.5	35.5
0.8	92.0	34.6

Table 12. **Runtime and Speedup Comparison.** We report the average runtime (ms/frame) and the relative speedup of SmoothDiffusion-VE compared to each method.

Method	ms/frame ↓	SmoothDiffusion-VE Speedup
SmoothDiffusion-VE	**36**	**1.0**×
StreamV2V	117.5	3.3×
StreamDiffusion	72.5	2.0×
SDEdit	48423	1345×
T2V-Zero	4687	130×
PNP	56306	1564×
DMT	68977	1916×

D. Additional Quantitative Results on VBench

In our quantitative evaluations, we utilize a subset of metrics from the VBench benchmark [8] to assess diverse aspects of video quality and consistency beyond CLIP Score and Warp Error. The specific VBench metrics employed in our work are Subject Consistency, Background Consistency, and Motion Smoothness. Below, we detail their computation as described in the VBench framework. For the results we follow the same experimental setup provided in Section 4 of the main paper.

D.1. Subject Consistency

This metric assesses if a primary subject's appearance remains consistent throughout the video. To compute this, VBench extracts a DINO feature vector [3], d_t, for each frame t, as DINO features are sensitive to identity variations. The final score, S_{subject}, is the average cosine similarity of each frame to both the first frame and its immediately preceding frame, indicating how well the subject's identity is maintained:

$$S_{\text{subject}} = \frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{2}(\langle d_1, d_t \rangle + \langle d_{t-1}, d_t \rangle) \quad (12)$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity. A higher S_{subject} score indicates better consistency.

D.2. Background Consistency

This metric evaluates the temporal stability of the background scene. It uses the CLIP ViT-L/14 image encoder [38] to extract a feature vector, c_t, for each frame. The

Table 13. VBench benchmark results. Best results are in bold (higher is better).

Method	Motion Smoothness	Subject Consistency	Background Consistency
SmoothDiffusion-VE (Ours)	**0.963**	**0.941**	**0.942**
DMT [52]	0.955	0.936	0.932
StreamV2V [26]	0.938	0.920	0.915
PnP [48]	0.928	0.909	0.917
SDEdit [33]	0.917	0.902	0.905
StreamDiffusion [21]	0.904	0.890	0.896
TokenFlow [11]	0.944	0.939	0.922

background consistency score, $S_{\text{background}}$, is then computed similarly by averaging the cosine similarity of each frame to the first and previous frames. A higher $S_{\text{background}}$ score signifies a more stable background throughout the video.

$$S_{\text{background}} = \frac{1}{T-1}\sum_{t=2}^{T}\frac{1}{2}(\langle c_1, c_t\rangle + \langle c_{t-1}, c_t\rangle) \tag{13}$$

D.3. Motion Smoothness

This metric assesses whether the motion in the video is smooth and physically plausible. It operates by dropping odd-numbered frames from a sequence and then tasking a video frame interpolation model to reconstruct them. The score is based on the Mean Absolute Error (MAE) between the original odd frames (f_o) and the reconstructed ones ($\hat{f}_o$). A lower MAE results in a higher final score, S_{motion}, as defined below, where E_{max} is the maximum possible pixel error (e.g., 255).

$$S_{\text{motion}} = 1 - \frac{\text{MAE}(f_o, \hat{f}_o)}{E_{max}} \tag{14}$$

A higher S_{motion} score indicates the motion was predictable and smooth.

D.4. VBench Performance Analysis

The results presented in Table 13 quantitatively demonstrate the superior performance of **SmoothDiffusion-VE** across all three evaluated VBench metrics. Our method consistently produces videos with greater temporal coherence compared to all competing approaches.

For **Motion Smoothness**, our method achieves the highest score of 0.963, indicating more fluid and physically plausible motion compared to the next-best method, DMT [52] (0.955), and other strong baselines like TokenFlow [11] (0.944). This is a critical factor in producing quality video edits that are free from unnatural jitter.

In **Subject Consistency**, SmoothDiffusion-VE again leads with a score of 0.941, surpassing the next-best method, TokenFlow [11] (0.939). This demonstrates its enhanced ability to maintain the identity and appearance of the primary subject throughout the video sequence, a significant challenge where other methods often falter. Similarly, our method scores highest in **Background Consistency** (0.942), which translates to more stable scenes with fewer flickering artifacts compared to all other methods.

Collectively, these VBench results validate that SmoothDiffusion-VE generates more temporally coherent and high-quality video edits compared to competing methods.

E. Thresholding

This section describes two widely used global thresholding methods, Yen's thresholding [53] and Otsu's thresholding [35], which can be applied to separate foreground from background in optical flow maps for our SmoothDiffusion-VE.

E.1. Yen's Thresholding

Yen's thresholding [53] is based on an entropy-driven criterion that aims to maximize the separation between foreground and background classes in a grayscale image. Let f be an image with intensity levels in the set $\{0, 1, \ldots, L-1\}$, and let $p(i)$ be the normalized histogram of f at intensity i, so that

$$\sum_{i=0}^{L-1} p(i) \ = \ 1 \tag{15}$$

We define:

$$P(t) = \sum_{i=0}^{t} p(i), \quad m(t) = \sum_{i=0}^{t} i\, p(i), \quad m_G = \sum_{i=0}^{L-1} i\, p(i) \tag{16}$$

which represent the cumulative distribution, partial mean, and global mean, respectively. For a threshold t, the foreground class is all pixels with intensities in $\{t+1, \ldots, L-1\}$, and the background class is those in $\{0, \ldots, t\}$. We denote these sets by Ω_1 (foreground) and Ω_2 (background). Yen's thresholding computes two entropies,

$$H_b(t) \ = \ -\sum_{i=0}^{t} \frac{p(i)}{P(t)} \ \ln\!\Big(\frac{p(i)}{P(t)}\Big) \tag{17}$$

$$H_f(t) \ = \ -\sum_{i=t+1}^{L-1} \frac{p(i)}{1-P(t)} \ \ln\!\Big(\frac{p(i)}{1-P(t)}\Big) \tag{18}$$

and sums them to form the Yen criterion $H_b(t)+H_f(t)$ The optimal threshold t^* maximizes this sum:

$$t^* \ = \ \arg\max_t \Big[H_b(t) + H_f(t)\Big] \tag{19}$$

Because each potential threshold $t \in \{0, \ldots, L-1\}$ only requires a constant-time computation of $H_b(t)$ and $H_f(t)$ once $p(i)$ is known cumulatively, the entire search is $\mathcal{O}(L)$, meaning it requires a single pass through all L intensity levels. Yen's thresholding often works well for multi-modal or skewed histograms, which can occur in high-motion scenarios where flow distributions are more varied.

E.2. Otsu's Thresholding

Otsu's thresholding [35] aims to minimize the intra-class variance (or equivalently, maximize the between-class variance) between a foreground set Ω_1 and a background set Ω_2. As with Yen's method, let $p(i)$ be the normalized histogram, and define:

$$P(t) = \sum_{i=0}^{t} p(i), \quad m(t) = \sum_{i=0}^{t} i\, p(i), \quad m_G = \sum_{i=0}^{L-1} i\, p(i) \tag{20}$$

Here, $P(t)$ is the fraction of pixels in the background, while $\mu_1(t) = m(t)/P(t)$ and $\mu_2(t) = (m_G - m(t))/(1 - P(t))$ give the mean intensities of background and foreground, respectively. The between-class variance is:

$$\sigma^2_{\text{between}}(t) = P(t)\,\big[\mu_1(t) - m_G\big]^2 + \big[1 - P(t)\big]\,\big[\mu_2(t) - m_G\big]^2 \tag{21}$$

Otsu's threshold t^* is chosen to maximize this variance:

$$t^* = \arg\max_t \ \sigma^2_{\text{between}}(t) \tag{22}$$

As before, $\mathcal{O}(L)$ computations suffice to evaluate $\sigma^2_{\text{between}}(t)$ for all t, because $P(t)$ and $m(t)$ can be maintained cumulatively. Otsu's method typically performs best on bimodal histograms, which correspond to well-separated foreground and background intensities. In high-motion scenarios where flow distributions can be complex or multimodal, Otsu's approach may struggle if classes overlap.

E.3. Complexity and Usage

Both Yen's and Otsu's thresholding are widely used for generating a foreground-background mask from optical flow. Each method requires $\mathcal{O}(L)$ time, where L is the number of possible intensity levels. For simple, clearly separated (bimodal) histograms, Otsu's method often yields good results. However, if the histogram is skewed or multi-modal (as can occur in busy or high-motion videos), Yen's thresholding may be more robust to variations and noise. We select Yen's thresholding by default for our motion-guided attention as the computational cost is negligible compared to the diffusion process, but Otsu's thresholding could also be selected.

F. Super-Resolution

We did not enable super-resolution by default and did not use it for our experimental results or user studies. But, we are able to integrate super-resolution with our method for fast video upscaling. We used an off-the-shelf deep learning super resolution model optimized for GPU deployments [34].

The model preserves image quality and achieves a 2.83 milliseconds (ms) inference time per frame for upscaling videos from 512×512 to 1024×1024 resolution on an Nvidia GeForce RTX 4090 GPU. Our proposed method in combination with the AI-enhanced video upscaling achieves 2× video resolutions with a total inference time of 38.8 ms/frame, 2.8 ms/frame for super-resolution and 36 ms/frame for our SmoothDiffusion-VE.

G. Stream Batch

We leverage the Stream Batch from StreamDiffusion [21] similar to StreamV2V [26] to enable faster processing.

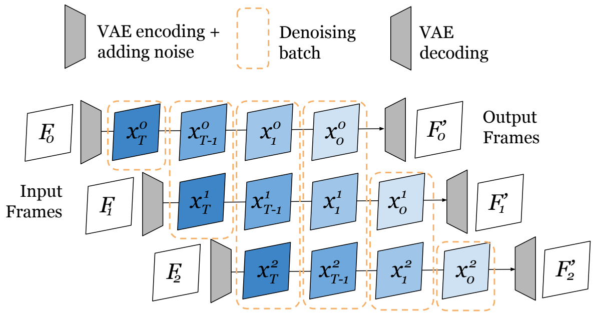

Figure 5. **Stream Batch.** The Stream Batch approach enables the joint denoising of multiple intermediates x across varying timesteps and frames. Jointly denoised groups are indicated in the figure above by an orange box, where x_t^i indicates the noisy latents from frame i at timestep t, F_i represents the input frame i, and F'_i represents the edited output frame i.

H. SDEdit Qualitative Comparisons

We also provide a direct comparison of our SmoothDiffusion-VE with SDEdit [33] in Figure 6. In the PS1 graphics edit, where a wolf is transformed into an orange fox, our method more accurately follows the prompt while maintaining greater temporal consistency. In contrast, SDEdit [33] produces inconsistent results across frames, particularly in column 3. For the pencil sketch edit, SmoothDiffusion-VE is yet again more consistent, while SDEdit [33] introduces unintended modifications, such as transforming the leaves in the background to a pencil. Furthermore, the pencil sketch video generated with SDEdit [33] is inconsistent across all frames. For the watercolor edit, SDEdit [33] exhibits some consistency, but fails to transform the wolf into a watercolor aesthetic, whereas our SmoothDiffusion-VE is able to successfully transform the wolf into a watercolor style while maintaining temporal consistency.

I. More Qualitative Results on Long Videos

The streaming architecture of SmoothDiffusion-VE, enabled by our Adaptive Feature Cache and temporal embedding, allows for the consistent editing of arbitrarily long videos. We demonstrate this on a 60-second, 1500-frame video from the LongVideoBench dataset [50]. As shown in Figure 7, our method maintains high temporal consistency for both the pencil sketch and claymation styles. The stylistic textures and the subject's core identity are preserved across changes in expression and pose, avoiding the degradation common in long-form video editing.

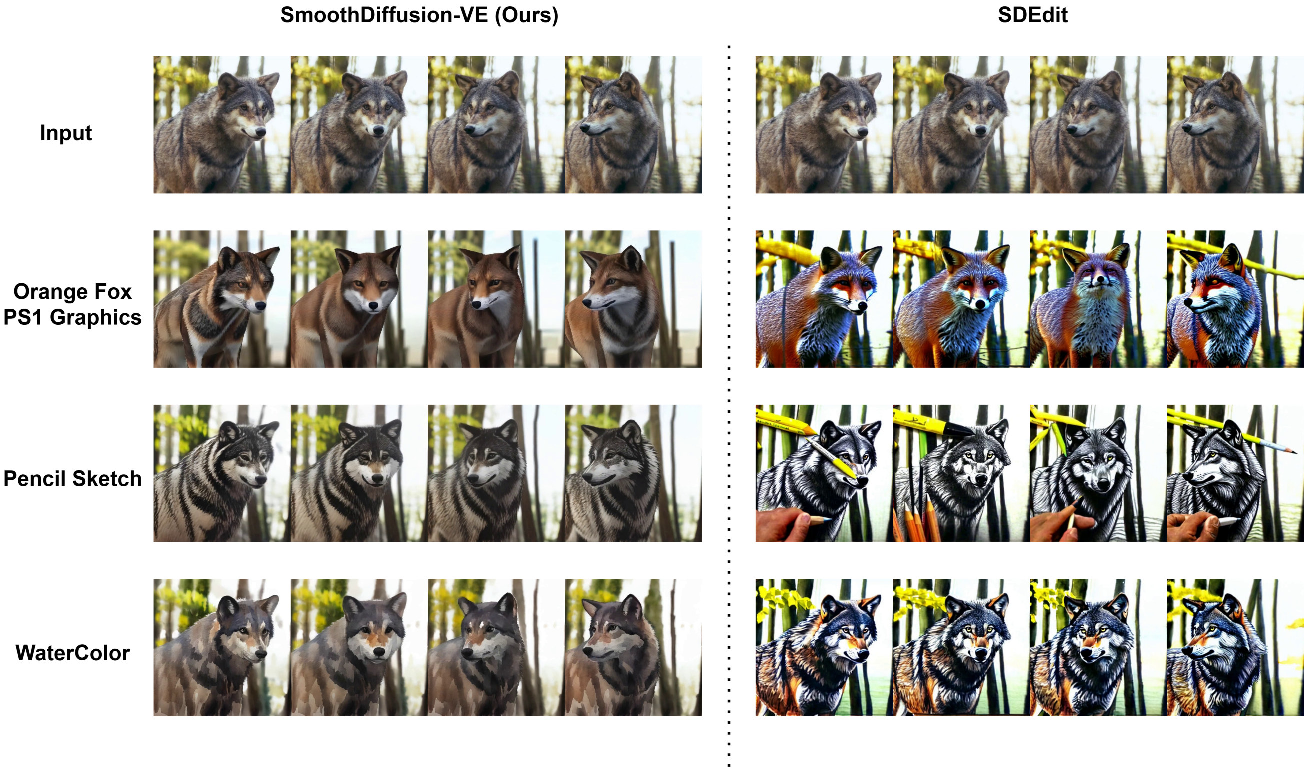

Figure 6. **SmoothDiffusion-VE Versus SDEdit Video Edit Quality Comparisons.** Video edits using our proposed SmoothDiffusion-VE produce higher quality results than SDEdit [33].

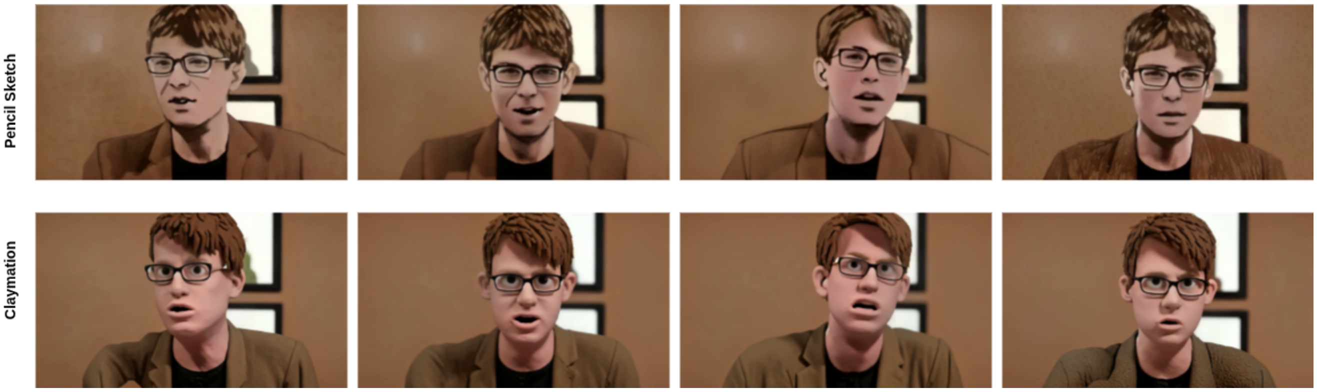

Figure 7. **Long Video Editing on a 1500-frame sequence.** Our method maintains stylistic consistency for both the pencil sketch and claymation edits across changes in the subject's expression and pose.