# StreamingT2V: Consistent, Dynamic, and Extendable Long Video Generation from Text (Appendix)

This appendix complements our main paper with experiments, in which we further investigate the text-to-video generation quality of StreamingT2V, demonstrate even longer sequences than those assessed in the main paper, and provide additional information on the implementation of StreamingT2V and the experiments carried out.

In Sec. A.1, a user study is conducted on the test set, in which all text-to-video methods under consideration are evaluated by humans to determine the user preferences.

Sec. A.2 supplements our main paper by additional qualitative results of StreamingT2V for very long video generation, and qualitative comparisons with competing methods.

In Sec. A.3, we present ablation studies to show the effectiveness of our proposed components CAM, APM and randomized blending.

In Sec. A.4, implementation and training details, including hyperparameters used in StreamingT2V, and implementation details of our ablated models are provided.

Sec. A.5 provides the prompts that compose our testset.

Finally, in Sec. A.6, the exact definition of the motion aware warp error (MAWE) is provided.

## A.1. User Study

We conduct a user study comparing our StreamingT2V method with prior work using the video results generated for the benchmark of Sec. 5.3 main paper. To remove potential biases, we resize and crop all videos to align them. The user study is structured as a one vs one comparison between our StreamingT2V method and competitors where participants are asked to answer three questions for each pair of videos:

- Which model has better motion?
- Which model has better text alignment?
- Which model has better overall quality?

We accept exactly one of the following three answers for each question: preference for the left model, preference for the right model, or results are considered equal. To ensure fairness, we randomize the order of the videos presented in each comparison, and the sequence of comparisons. Fig. A.1 shows the preference score obtained from the user study as the percentage of votes devoted to the respective answer.

Across all comparisons to competing methods, StreamingT2V is significantly more often preferred than the competing method, which demonstrates that StreamingT2V clearly improves upon state-of-the-art for long video generation. For instance in motion quality, as the results of StreamingT2V are non-stagnating videos, temporal consistent and possess seamless transitions between chunks, $65\%$ of the votes were preferring StreamingT2V, compared to $17\%$ of the votes preferring SEINE.

Competing methods are much more affected by quality degradation over time, which is reflected in the preference for StreamingT2V in terms of *text alignment* and *overall quality*.

## A.2. Qualitative Results

Complementing our visual results shown in the main paper (see Fig 5 main paper) , we present additional qualitative results of StreamingsT2V on our test set on very long video generation, and further qualitative comparisons to prior works on 240 frames.

### A.2.1. Very Long Video Generation

Supplementing our main paper, we show that StreamingT2V can be used for very long video generation. To this end, we generate and show videos consisting of 1200 frames, thus spanning 2 minutes, which is 5 times longer than the ones produced for the experiments in our main paper. Fig. A.2 show these text-to-video results of StreamingT2V for different actions, *e.g. dancing*, *running*, or *camera moving*, and different characters like *bees* or *jellyfish*. We can observe that scene and object features are kept across each video generation (see *e.g.* Fig. A.2(a)&(e)), thanks to our proposed APM module. Our proposed CAM module ensures that generated videos are temporally smooth, with seamless transitions between video chunks, and not stagnating (see *e.g.* Fig. A.2(f)&(k)).

### A.2.2. More Qualitative Evaluations.

The visual comparisons shown in Fig. A.3, A.4, A.5, A.6 demonstrate that StreamingT2V significantly excels the generation quality of all competing methods. StreamingT2V shows non-stagnating videos with good motion

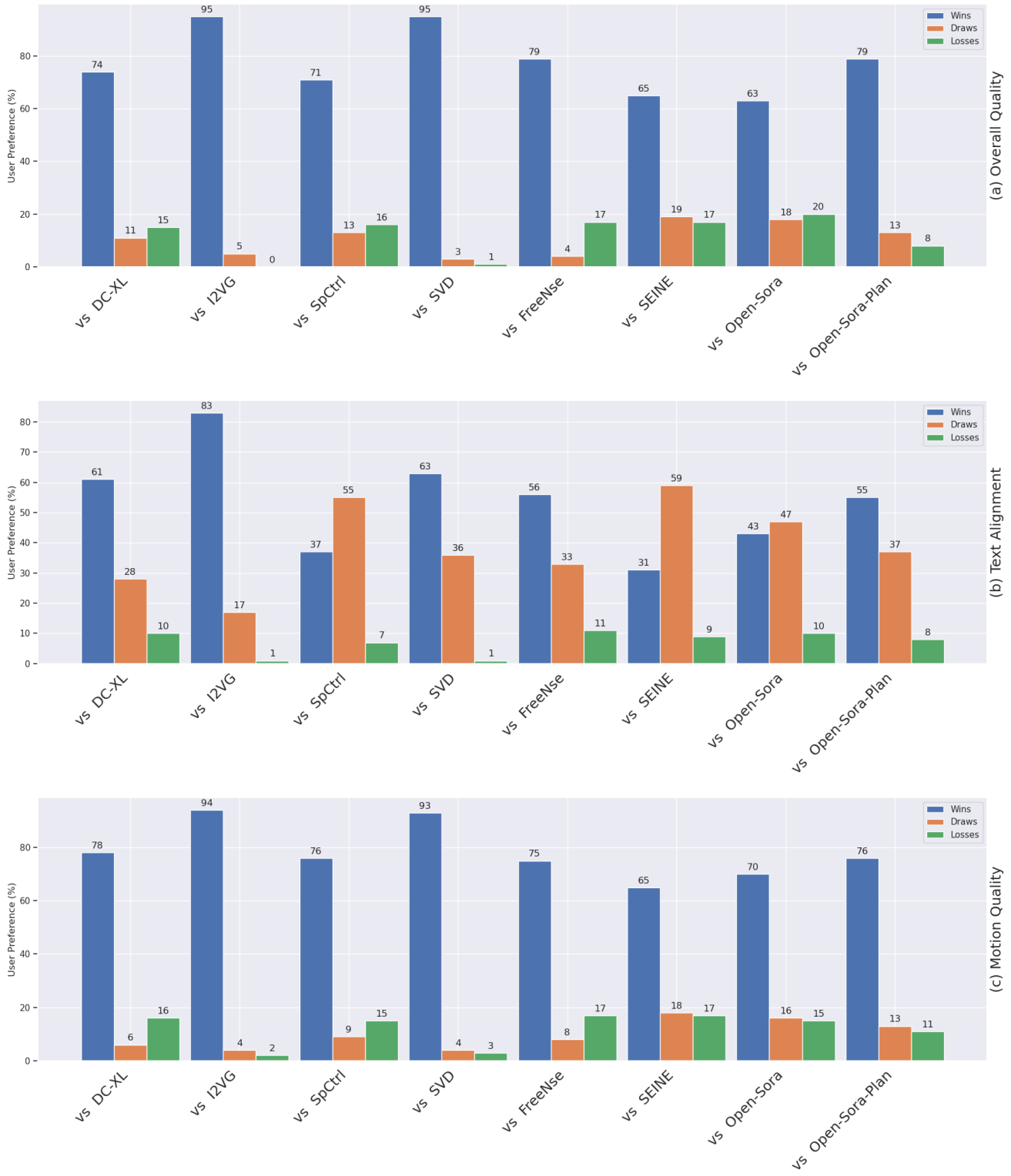



Figure A.1. We conduct a user study, asking humans to assess the test set results (mentioned in Sec. 5.3 of the paper) in a one-to-one evaluation, where for any prompt of the test set and any competing method, the results of the competing method have to be compared with the corresponding results of our StreamingT2V method. For each comparison of our method to a competing method, we report the relative of number votes that prefer StreamingT2V (*i.e.* wins), that prefer the competing method (*i.e.* losses), and that consider results from both methods as equal (*i.e.* draws).

(a) People dancing in room filled with fog and colorful lights

(b) Camera moving in a wide bright ice cave

(c) Marvel at the diversity of bee species

(d) Dive into the depths of the ocean: explore vibrant coral reefs

(e) Venture into the kelp forests: weave through towering underwater forests

(f) Experience the dance of jellyfish: float through mesmerizing swarms of jellyfish

(g) Enter the realm of ice caves: venture into frozen landscapes

(h) Wide shot of battlefield, stormtroopers running at night, smoke, fires and smokes

(i) Witness the wonders of sea caves

(j) Camera moving around vast deserts, where dunes stretch endlessly into the horizon

(k) Enter the fascinating world of bees: explore the intricate workings of a beehive

Figure A.2. Qualitative results of StreamingT2V for different prompts. Each video has 1200 frames.

quality, in particular seamless transitions between chunks and temporal consistency.

Videos generated by DynamiCrafter-XL eventually possess severe image quality degradation. For instance, we observe in Fig. A.3 eventually wrong colors at the beagle's face and the background pattern heavily deteriorates.

The quality degradation also heavily deteriorates the textual alignment (see the result of DynamiCrafter-XL in Fig. A.5). Across all visual results, the method SVD is even more susceptible to these issues.

The methods SparseControl and FreeNoise eventually lead to almost stand-still, and are thus not able to perform the action described in a prompt, *e.g.* "zooming out" in Fig. A.6. Likewise, also SEINE is not following this camera instructions (see Fig. A.6).

OpenSora is mostly not generating any motion, leading either to complete static results (Fig. A.3), or some image warping without motion (Fig. A.5). OpenSoraPlan is loosing initial object details and suffers heavily from quality degradation through the autoregressive process, *e.g.* the dog is hardly recognizable at the of the video generation (see Fig. A.3), showing again that a sophisticated conditioning mechanism is necessary.

I2VGen-XL shows low motion amount, and eventually quality degradation, leading eventually to frames that are weakly aligned to the textual instructions.

We further analyse visually the chunk transitions using an X-T slice visualization in Fig. A.7. We can observe that StreamingT2V leads to smooth transitions. In contrast, we observe that conditioning via CLIP or concatenation may lead to strong inconsistencies between chunks.

## A.3. Ablation Studies

To assess the importance of our proposed components, we conduct several ablation studies on a randomly sampled set of 75 prompts from our validation set that we used during training.

Specifically, we compare CAM against established conditioning approaches in Sec. A.3.1, analyse the impact of our long-term memory APM in Sec. A.3.2, and ablate on our modifications for the video enhancer in Sec. A.3.3.

### A.3.1. Conditional Attention Module.

To analyse the importance of CAM, we compare CAM (w/o APM) with two baselines (baseline details in Sec. A.3.1.1): (i) Connect the features of CAM with the skip-connection of the UNet via zero convolution, followed by addition. We zero-pad the condition frame and concatenate it with a frame-indicating mask to form the input for the modified CAM, which we denote as Add-Cond. (ii) We append the conditional frames and a frame-indicating mask to input of Video-LDM's Unet along the channel dimension, but do not use CAM, which we denote as Conc-Cond. We train our method with CAM and the baselines on the same dataset. Architectural details (including training) of these baselines are provided in the appendix.

We obtain an SCuts score of 0.24, 0.284 and 0.03 for Conc-Cond, Add-Cond and Ours (w/o APM), respectively. This shows that the inconsistencies in the input caused by the masking leads to frequent inconsistencies in the generated videos and that concatenation to the Unet's input is a too weak conditioning. In contrast, our CAM generates consistent videos with a SCuts score that is 88% lower than the baselines.

#### A.3.1.1. Ablation models

For the ablation of CAM, we considered two baselines that we compare with CAM. Here we provide additional implementation details of these baselines.

The ablated model *Add-Cond* applies to the features of CAM (*i.e.* the outputs of the encoder and middle layer of the ControlNet part in Fig 3 from main paper) zero-convolution, and uses addition to fuse it with the features of the skip-connection of the UNet (similar to ControlNet [7]) (see Fig. A.10). We provide here additional details to construct this model. Given a video sample $\mathcal{V} \in \mathbb{R}^{F\times H\times W\times 3}$ with $F = 16$ frames, we construct a mask $M \in \{0,1\}^{F\times H\times W\times 3}$ that indicates which frame we use for conditioning, *i.e.* $M^f[i,j,k] = M^f[i',j',k']$ for all frames $f = 1,\dots,F$ and for all $i,j,k,i',j',k'$. We require that exactly $F - F_{\text{cond}}$ frames are masked, *i.e.*

$$\sum_{f=1}^{F} M^f[i,j,k] = F - F_{\text{cond}}, \text{ for all } i,j,k. \tag{A.1}$$

We concatenate $[\mathcal{V}\odot M, M]$ along the channel dimension and use it as input for the image encoder $\mathcal{E}_{\text{cond}}$, where $\odot$ denotes element-wise multiplication.

During training, we randomly set the mask $M$. During inference, we set the mask for the first 8 frames to zero, and for the last 8 frames to one, so that the model conditions on the last 8 frames of the previous chunk.

For the ablated model *Conc-Cond*, we start from our Video-LDM's UNet, and modify its first convolution. Like for *Add-Cond*, we consider a video $\mathcal{V}$ of length $F = 16$ and a mask $M$ that encodes which frames are overwritten by zeros. Now the Unet takes $[z_t, \mathcal{E}(\mathcal{V})\odot M, M]$ as input, where we concatenate along the channel dimension. As with *Add-Cond*, we randomly set $M$ during training so that the information of 8 frames is used, while during inference, we set it such that the last 8 frames of the previous chunk are used. Here $\mathcal{E}$ denotes the VQ-GAN encoder (see Sec. 3).

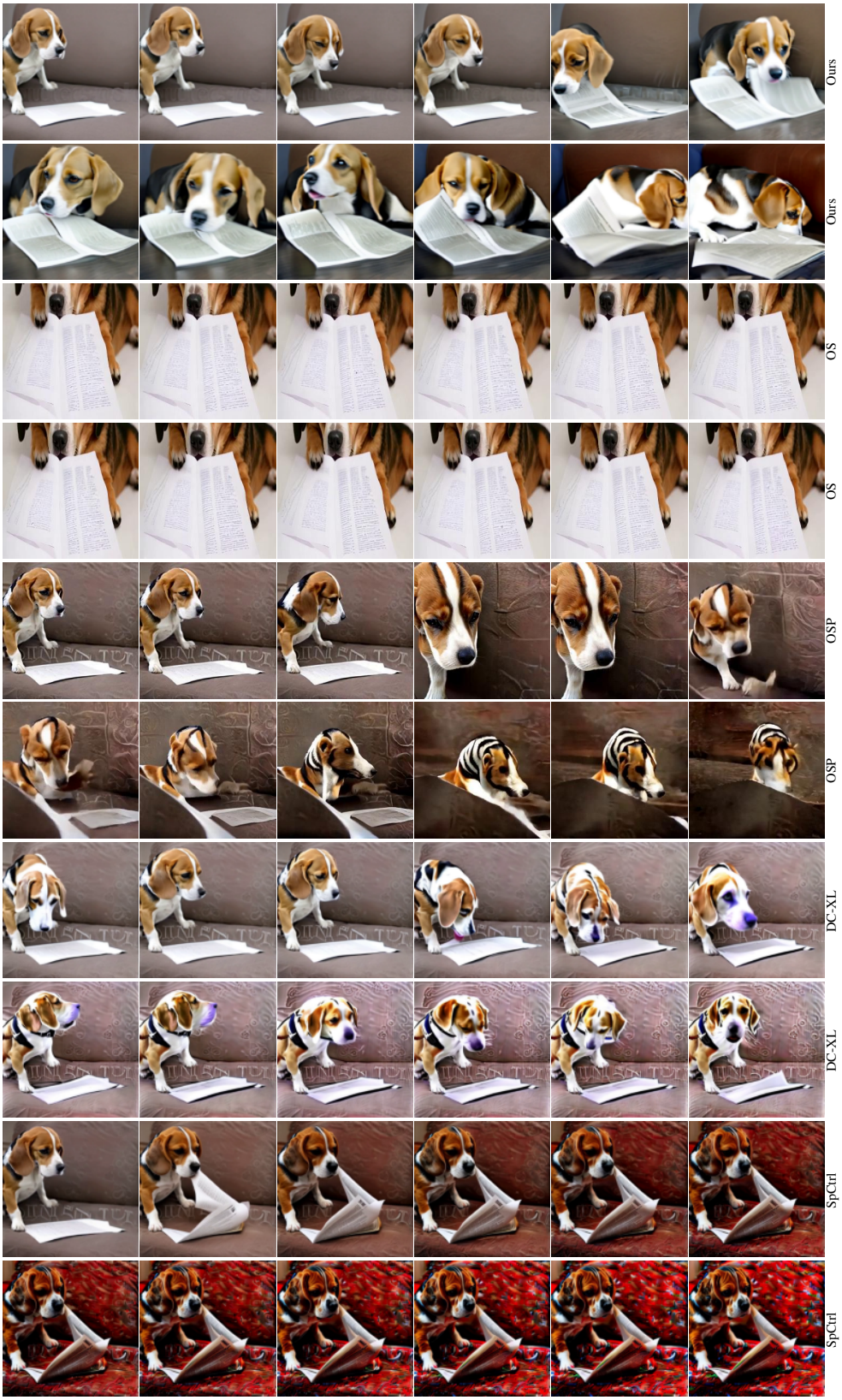



Figure A.3. Video generation for the prompt "*A beagle reading a paper*", using StreamingT2V and competing methods. For each method, the image sequence of its first row is continued by the image in the leftmost column of the following row.

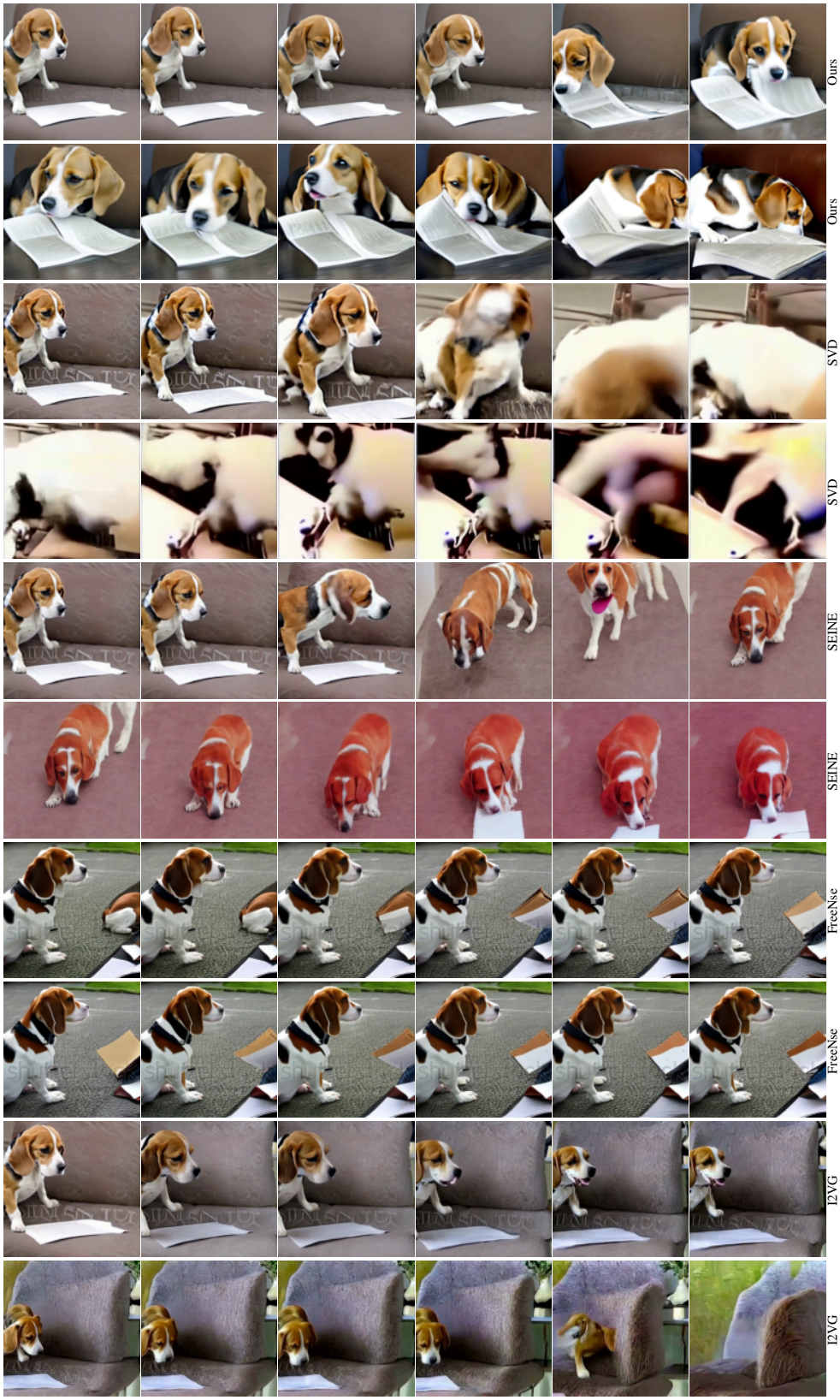



Figure A.4. Video generation for the prompt "*A beagle reading a paper*", using StreamingT2V and competing methods. For each method, the image sequence of its first row is continued by the image in the leftmost column of the following row.

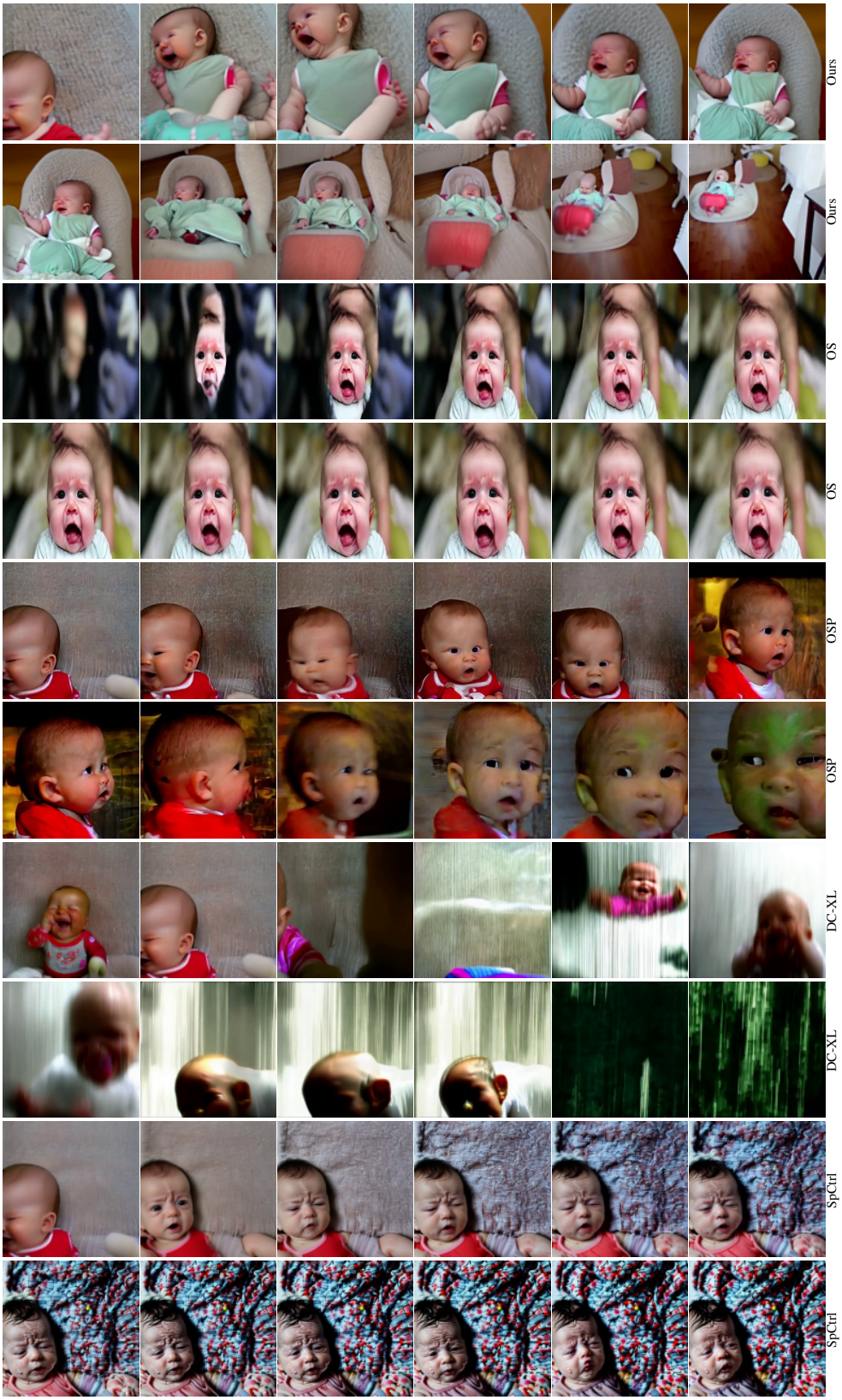



Figure A.5. Video generation for the prompt "*Camera is zooming out and the baby starts to cry*", using StreamingT2V and competing methods. For each method, the image sequence of its first row is continued by the image in the leftmost column of the following row.

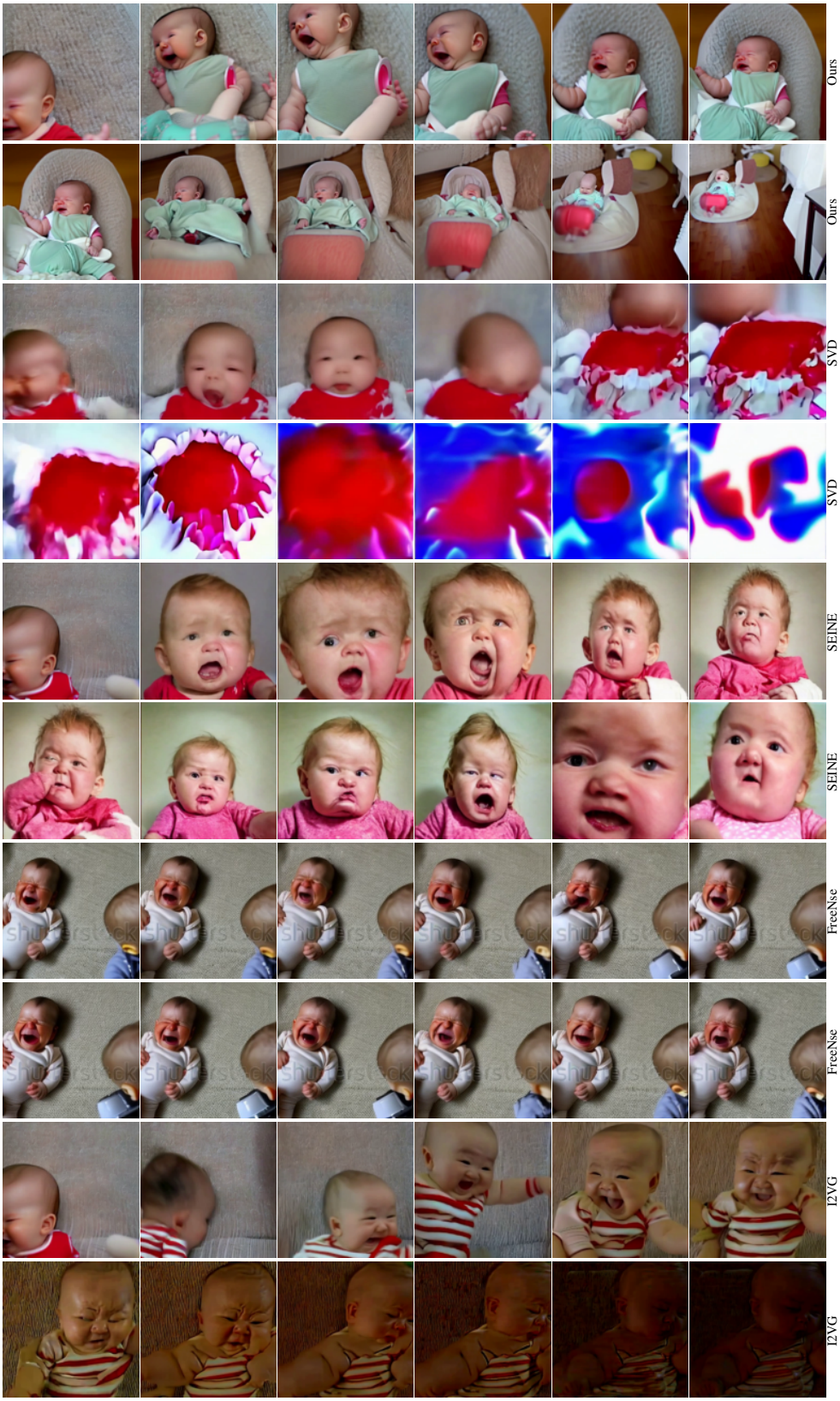
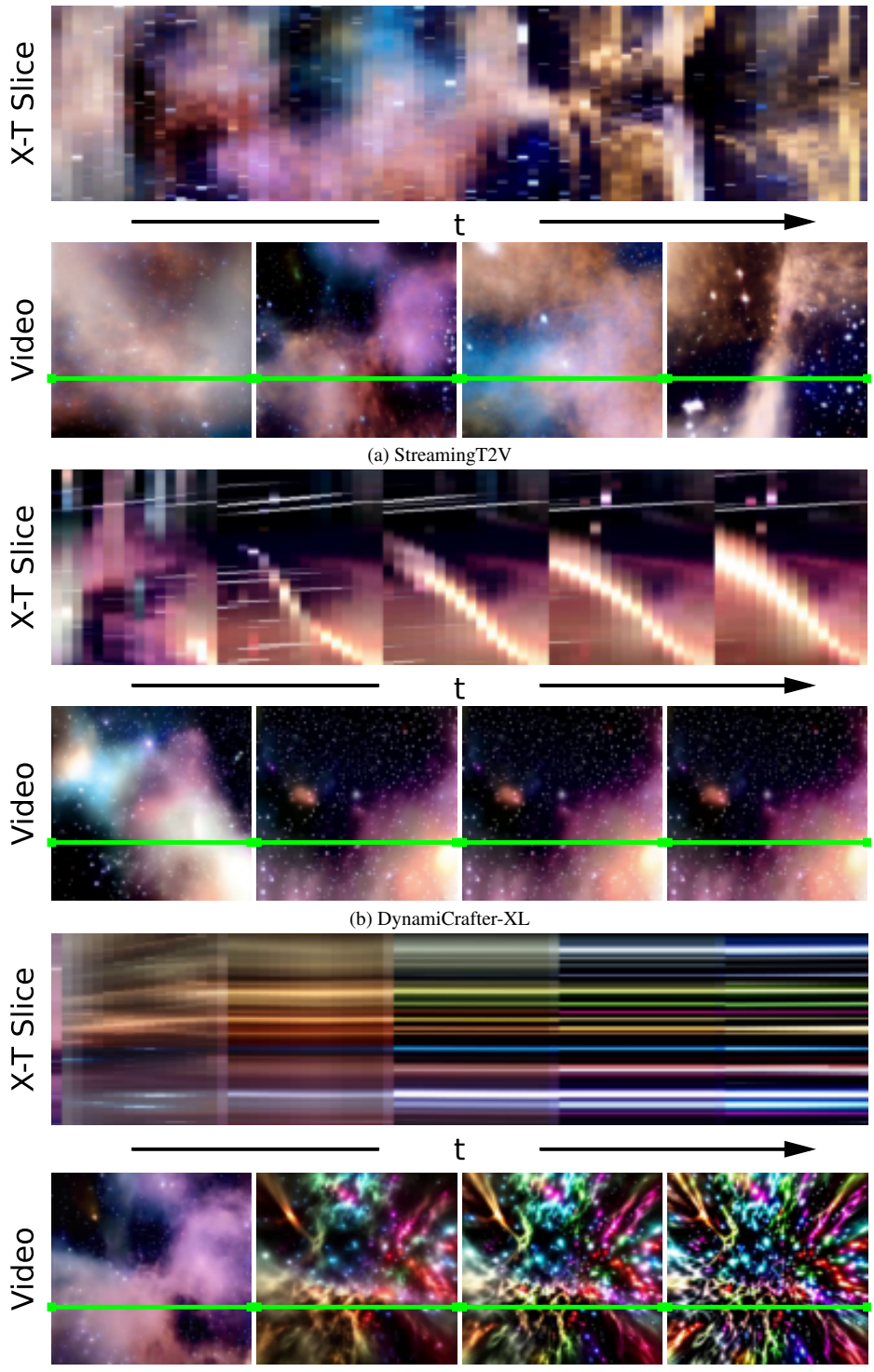



Figure A.6. Video generation for the prompt "*Camera is zooming out and the baby starts to cry*", using StreamingT2V and competing methods. For each method, the image sequence of its first row is continued by the image in the leftmost column of the following row.




(a) StreamingT2V

(b) DynamiCrafter-XL

(c) SparseControl

Figure A.7. Visual comparison of SparseControl, DynamiCrafter-XL and StreamingT2V. All text-to-video results are generated using the same prompt. The X-T slice visualization shows that DynamiCrafter-XL and SparseControl suffer from severe chunk inconsistencies and repetitive motions. In contrast, our method shows seamless transitions and evolving content.

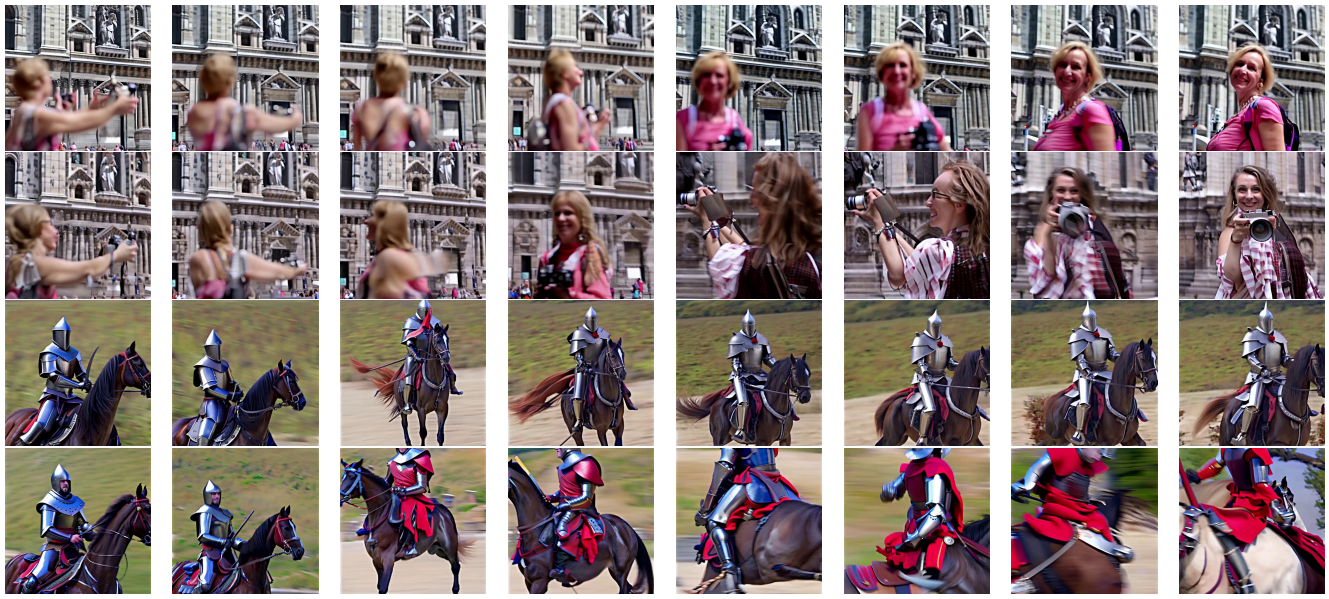


Figure A.8. Ablation study on the APM module. Top row is generated from StreamingT2V, bottom row is generated from StreamingT2V w/o APM.

### A.3.2. Appearance Preservation Module

We analyse the impact of utilizing long-term memory in the context of long video generation.

Fig. **??** and Fig. A.8 show that long-term memory greatly helps keeping the object and scene features across autoregressive generations. Thanks to the usage of long-term information via our proposed APM module, identity and scene features are preserved throughout the video. For instance, the face of the woman in Fig. A.8 (including all its tiny details) are consistent[1] across the video generation. Also, the style of the jacket and the bag are correctly generated throughout the video. Without having access to a long-term memory, these object and scene features are changing over time.

This is also supported quantitatively. We utilize a person re-identification score to measure feature preservation (definition in Sec. A.3.2.1), and obtain scores of 93.42 and 94.95 for Ours w/o APM, and Ours, respectively. Our APM module thus improves the identity/appearance preservation. Also the scene information is better kept, as we observe an image distance score in terms of LPIPS [8] of 0.192 and 0.151 for Ours w/o APM and Ours, respectively. We thus have an improvement in terms of scene preservation of more than 20% when APM is used.

[1]The background appears to have changed. However, please note that the camera is rotating so that a different area behind the two woman becomes visible, so that the background change is correct.

#### A.3.2.1. Measuring Feature Preservation.

We employ a person re-identification score as a proxy to measure feature preservation. To this end, let $P_n = \{p_i^n\}$ be all face patches extracted from frame $n$ using an off-the-shelf head detector [4] and let $F_i^n$ be the corresponding facial feature of $p_i^n$, which we obtain from an off-the-shelf face recognition network [4]. Then, for frame $n$, $n_1 := |P_n|$, $n_2 := |P_{n+1}|$, we define the re-id score re-id$(n)$ for frame $n$ as

$$\text{re-id}(n) := \begin{cases} \max_{i,j} \cos\Theta(F_i^n, F_j^{n+1}), & n_1, n_2 > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{A.2}$$

where $\cos\Theta$ is the cosine similarity. Finally, we obtain the re-ID score of a video by averaging over all frames, where the two consecutive frames have face detections, *i.e.* with $m := |\{n \in \{1, .., N\} : |P_n| > 0\}|$, we compute the weighted sum:

$$\text{re-id} := \frac{1}{m} \sum_{n=1}^{N-1} \text{re-id}(n), \tag{A.3}$$

where $N$ denotes the number of frames.

### A.3.3. Randomized Blending.

We assess our randomized blending approach by comparing against two baselines. (B) enhances each video chunk independently, and (B+S) uses shared noise for consecutive chunks, with an overlap of 8 frames, but not randomized blending. We compute per sequence the standard deviation of the optical flow magnitudes between consecutive frames and average over all frames and sequences, which indicates temporal smoothness. We obtain the scores 8.72, 6.01 and 3.32 for B, B+S, and StreamingT2V, respectively. Thus, noise sharing improves chunk consistency (by 31% vs B), but it is significantly further improved by randomized blending (by 62% vs B).

These findings are supported visually. Fig. A.9 shows ablated results on our randomized blending approach. From the X-T slice visualizations we can see that the randomized blending leads to smooth chunk transitions, confirming our observations and quantitative evaluations. In contrast, when naively concatenating enhanced video chunks, or using shared noise, the resulting videos possess visible inconsistencies between chunks.

## A.4. Implementation detail

We generate $F = 16$ frames, condition on $F_{\text{cond}} = 8$ frames, and display videos with 10 FPS. Training is conducted using an internal dataset. We sample with 3FPS@256x256 16 frames (during CAM training) and 32 frames (during CAM+APM training).

**CAM training**: we freeze the weights of the pre-trained Video-LDM and train the new layers of CAM with batch size 8 and learning rate $5 \cdot 10^{-5}$ for 400K steps.

**CAM+APM training**: After the CAM training, we freeze the CLIP encoder and the temporal layers of the main branch, and train the remaining layers for 1K steps.

The image encoder $\mathcal{E}_{\text{cond}}$ used in CAM is composed of stacked 2D convolutions, layer norms and SiLU activations. For the video enhancer, we diffuse an input video using $T' = 600$ steps.

In order to train the APM module, we randomly sample an anchor frame out of the first 16 frames. For the conditioning and denoising, we use the frames $17 - 24$ and $17 - 32$, respectively. This aligns training with inference, where there is a large time gap between the conditional frames and the anchor frame. In addition, by randomly sampling an anchor frame, the model can leverage the CLIP information only for the extraction of high-level semantic information, as we do not provide a frame index to the model.

### A.4.1. Streaming T2V Stage

For the StreamingT2V stage, we use classifier free guidance [1, 2] from text and the anchor frame. More precisely, let $\epsilon_\theta(x_t, t, \tau, a)$ denote the noise prediction in the StreamingT2V stage for latent code $x_t$ at diffusion step $t$, text $\tau$ and anchor frame $a$. For text guidance and guidance by the anchor frame, we introduce weights $\omega_{\text{text}}$ and $\omega_{\text{anchor}}$, respectively. Let $\tau_{\text{null}}$ and $a_{\text{null}}$ denote the empty string, and the image with all pixel values set to zero, respectively. Then, we obtain the multi-conditioned classifier-free-guided noise prediction $\hat{\epsilon}_\theta$ (similar to DynamiCrafter-XL [6]) from the noise predictor $\epsilon$ via

$$\begin{aligned} \hat{\epsilon}_\theta(x_t, t, \tau, a) = \epsilon_\theta(x_t, t, \tau_{\text{null}}, a_{\text{null}}) \\ + \omega_{\text{text}}\big(\epsilon_\theta(x_t, t, \tau, a_{\text{null}}) \\ - \epsilon_\theta(x_t, t, \tau_{\text{null}}, a_{\text{null}})\big) \\ + \omega_{\text{anchor}}\big(\epsilon_\theta(x_t, t, \tau, a) \\ - \epsilon_\theta(x_t, t, \tau, a_{\text{null}})\big). \end{aligned} \tag{A.4}$$

We then use $\hat{\epsilon}_\theta$ for denoising. In our experiments, we set $\omega_{\text{text}} = \omega_{\text{anchor}} = 7.5$. During training, we randomly replace $\tau$ with $\tau_{\text{null}}$ with 5% likelihood, the anchor frame $a$ with $a_{\text{null}}$ with 5% likelihood, and we replace at the same time $\tau$ with $\tau_{\text{null}}$ and $a$ with $a_{\text{null}}$ with 5% likelihood.

Additional hyperparameters for the architecture, training and inference of the Streaming T2V stage are presented in Tab. A.7, where *Per-Pixel Temporal Attention* refers to the attention module used in CAM (see Fig. 3)

## A.5. Test set prompts

1. A camel resting on the snow field.
2. Camera following a pack of crows flying in the sky.

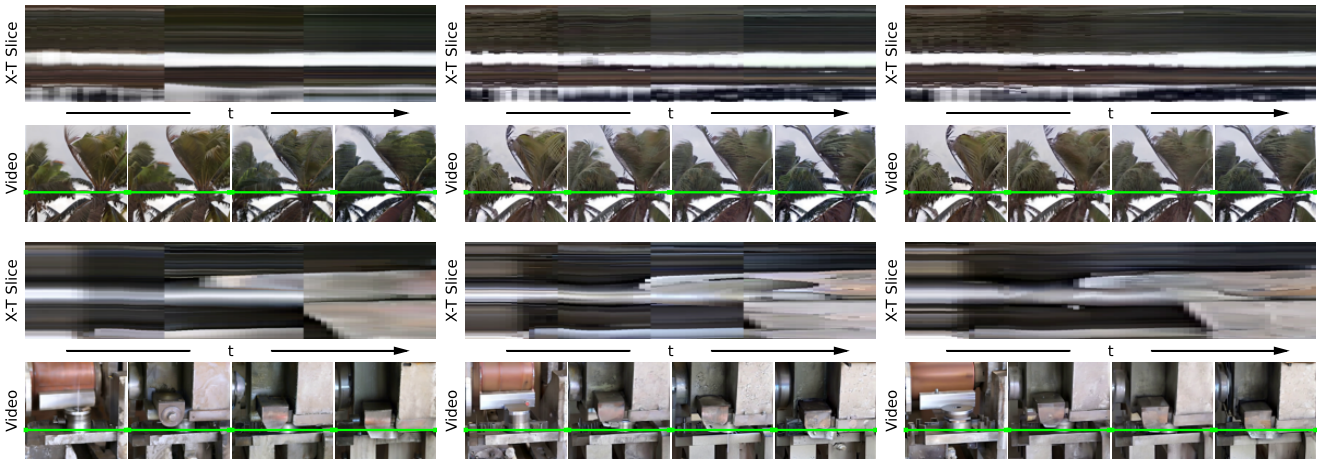



Figure A.9. Ablation study on our video enhancer improvements. The X-T slice visualization shows that randomized blending leads to smooth chunk transitions, while both baselines have clearly visible, severe inconsistencies between chunks.

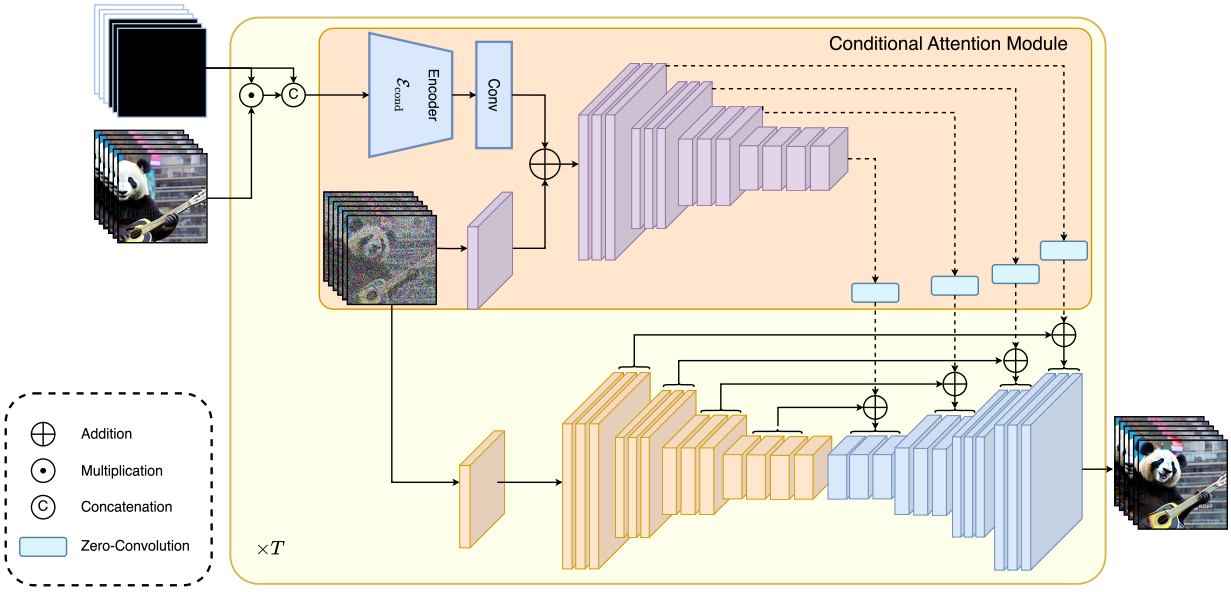



Figure A.10. Illustration of the Add-Cond baseline, which is used in Sec. A.3.1.

3. A knight riding on a horse through the countryside.
4. A gorilla eats a banana in Central Park.
5. Men walking in the rain.
6. Ants, beetles and centipede nest.
7. A squirrel on a table full of big nuts.
8. Close flyover over a large wheat field in the early morning sunlight.
9. A squirrel watches with sweet eyes into the camera.
10. Santa Claus is dancing.
11. Chemical reaction.
12. Camera moving in a wide bright ice cave, cyan.
13. Prague, Czech Republic. Heavy rain on the street.
14. Time-lapse of stormclouds during thunderstorm.
15. People dancing in room filled with fog and colorful lights.
16. Big celebration with fireworks.
17. Aerial view of a large city.
18. Wide shot of battlefield, stormtroopers running at night, fires and smokes and explosions in background.
19. Explosion.
20. Drone flythrough of a tropical jungle with many birds.
21. A camel running on the snow field.
22. Fishes swimming in ocean camera moving.
23. A squirrel in Antarctica, on a pile of hazelnuts cinematic.

| | |
|---|---|
| **Per-Pixel Temporal Attention** | |
| Sequence length Q | 16 |
| Sequence length K,V | 8 |
| Token dimensions | 320, 640, 1280 |
| **Appearance Preservation Module** | |
| CLIP Image Embedding Dim | 1024 |
| CLIP Image Embedding Tokens | 1 |
| MLP hidden layers | 1 |
| MLP inner dim | 1280 |
| MLP output tokens | 16 |
| MLP output dim | 1024 |
| 1D Conv input tokens | 93 |
| 1D Conv output tokens | 77 |
| 1D Conv output dim | 1024 |
| Cross attention sequence length | 77 |
| **Training** | |
| Parametrization | $\epsilon$ |
| **Diffusion Setup** | |
| Diffusion steps | 1000 |
| Noise scheduler | Linear |
| $\beta_0$ | 0.0085 |
| $\beta_T$ | 0.0120 |
| **Sampling Parameters** | |
| Sampler | DDIM |
| Steps | 50 |
| $\eta$ | 1.0 |
| $\omega_{\text{text}}$ | 7.5 |
| $\omega_{\text{anchor}}$ | 7.5 |

Table A.7. Hyperparameters of Streaming T2V Stage. Additional architectural hyperparameters are provided by the Modelsope report [5].

24. Fluids mixing and changing colors, closeup.
25. A horse eating grass on a lawn.
26. The fire in the car is extinguished by heavy rain.
27. Camera is zooming out and the baby starts to cry.
28. Flying through nebulas and stars.
29. A kitten resting on a ball of wool.
30. A musk ox grazing on beautiful wildflowers.
31. A hummingbird flutters among colorful flowers, its wings beating rapidly.
32. A knight riding a horse, pointing with his lance to the sky.
33. steampunk robot looking at the camera.
34. Drone fly to a mansion in a tropical forest.
35. Top-down footage of a dirt road in forest.
36. Camera moving closely over beautiful roses blooming time-lapse.
37. A tiger eating raw meat on the street.
38. A beagle looking in the Louvre at a painting.
39. A beagle reading a paper.
40. A panda playing guitar on Times Square.
41. A young girl making selfies with her phone in a crowded street.
42. Aerial: flying above a breathtaking limestone structure on a serene and exotic island.
43. Aerial: Hovering above a picturesque mountain range on a peaceful and idyllic island getaway.
44. A time-lapse sequence illustrating the stages of growth in a flourishing field of corn.
45. Documenting the growth cycle of vibrant lavender flowers in a mesmerizing time-lapse.
46. Around the lively streets of Corso Como, a fearless urban rabbit hopped playfully, seemingly unfazed by the fashionable surroundings.
47. Beside the Duomo's majestic spires, a fearless falcon soared, riding the currents of air above the iconic cathedral.
48. A graceful heron stood poised near the reflecting pools of the Duomo, adding a touch of tranquility to the vibrant surroundings.
49. A woman with a camera in hand joyfully skipped along the perimeter of the Duomo, capturing the essence of the moment.
50. Beside the ancient amphitheater of Taormina, a group of friends enjoyed a leisurely picnic, taking in the breathtaking views.

## A.6. MAWE Definition

For MAWE, we measure the motion amount using OFS (optical flow score), which computes for a video the mean of the squared magnitudes of all optical flow vectors between any two consecutive frames. Furthermore, for a video $\mathcal{V}$, we consider the mean warp error [3] $W(\mathcal{V})$, which measures the average squared L2 pixel distance from a frame to its warped subsequent frame, excluding occluded regions. Finally, MAWE is defined as:

$$\text{MAWE}(\mathcal{V}) := \frac{W(\mathcal{V})}{\text{OFS}(\mathcal{V})}, \tag{A.5}$$

which we found to be well-aligned with human perception.

For MAWE, we measure the motion amount using OFS (optical flow score), which computes for a video the mean of the squared magnitudes of all optical flow vectors between any two consecutive frames. Furthermore, for a video $\mathcal{V}$, we consider the mean warp error [3] $W(\mathcal{V})$, which measures the average squared L2 pixel distance from a frame to its warped subsequent frame, excluding occluded regions. Finally, MAWE is defined as:

$$\text{MAWE}(\mathcal{V}) := \frac{W(\mathcal{V})}{\text{OFS}(\mathcal{V})}, \tag{A.6}$$

which we found to be well-aligned with human perception.

## References


[1] 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. 11

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 11

[3] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 13

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 11

[5] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 13

[6] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 11

[7] 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. 4

[8] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 10