

Supplementary material of TIME RIPPLE: Accelerating vDiTs by Understanding the Spatio-Temporal Correlations in Latent Space

Anonymous CVPR submission

Paper ID NA

1. Attention Reuse Algorithm

Algorithm 1: Attention reuse algorithm

Input : Input query Q_i , input key K_i , input value V_i , timestep i

Parameter: Start timestep i_{min} , end timestep i_{max} , maximum threshold θ_{max} , minimum threshold θ_{min} , video latent shape $(B, H, T' \cdot H' \cdot W', C)$, window size W

Output : Attention output O_i

```

1 if  $i < i_{min}$  then
2    $A_i = Q_i \cdot K_i^T$ ;
3    $O_i = \text{Projection}(A_i, V_i)$ ;
4 else
5   if  $i > i_{max}$  then
6      $\theta \leftarrow \theta_{max}$ ;
7   else
8      $\theta \leftarrow (i - i_{min}) \cdot \frac{\theta_{max} - \theta_{min}}{i_{max} - i_{min}} + \theta_{min}$ ;
9   end
10  for each axis  $a \in \{T', H', W'\}$  do
11    for  $X \in \{Q_i, K_i\}$  do
12      std:  $s \leftarrow \text{std}(X, W, a)$ ;
13       $mask_X^a \leftarrow (s < \theta)$ ;
14    end
15  end
16   $mask \leftarrow mask_X^a$  for
17     $X \in \{Q_i, K_i\}, a \in \{T', H', W'\}$ ;
18   $A_{Sparse} \leftarrow \text{SparseAttention}(Q_i, K_i, mask)$ ;
19   $O_i \leftarrow \text{ReusableProjection}(A_i, V_i, mask)$ ;
20 end
21 return  $O_i$ 

```

Algo. 1 presents our proposed attention reuse algorithm, which aims to accelerate attention computation in diffusion models by exploiting local spatio-temporal redundancy at each denoising timestep.

Algorithm Setup. The algorithm takes the queries, Q_i ,

and keys, K_i , at timestep, i , as input along with the following hyper-parameters:

- i_{min}, i_{max} : These two timesteps define the range during which threshold, θ , in Sec. 3.3 is linearly increased.
- $\theta_{min}, \theta_{max}$: The minimum and maximum thresholds for reuse.
- $(B, H, T' \cdot H' \cdot W', C)$: Flattened latent feature shape, where T' , H' , and W' represent the temporal and spatial dimensions in latent space.
- W' : The window size used to divide the variable.

The output of our attention reuse algorithm is the attention output, O_i , at timestep i .

Step-by-Step Procedure. The overall procedure of our attention reuse algorithm consists of six stages:

① At each denoising timestep i , if $i < i_{min}$, the attention computation performs canonical self-attention without reuse. Specifically, the full attention map A_i is calculated as the dot product, $Q_i \cdot K_i^T$, and the final output O_i is derived via the standard linear projection, $\text{Softmax}(\frac{A_i}{\sqrt{d}}) \cdot V_i$.

② When i falls within the adaptive reuse window $[i_{min}, i_{max}]$, we first calculate the reuse threshold θ using the equation as follows,

$$\theta = (i - i_{min}) \cdot \frac{\theta_{max} - \theta_{min}}{i_{max} - i_{min}} + \theta_{min}.$$

If i surpasses i_{end} , the threshold, θ , is clamped to the maximum value θ_{max} .

③ To recover the spatio-temporal structure, the query and key tensors Q_i and K_i are reshaped from a flattened sequence back into the latent video grid with shape (B, H, T', H', W', C) , where T' , H' , and W' denote temporal and spatial dimensions.

④ Next, we find the potential reuse pattern along each axis $a \in \{T', H', W'\}$. Specifically, we iterate over each dimension. For each dimension, a , we slice both Q_i and K_i into multiple blocks with the same dimension of W . Within each block, we then compute the standard deviation, s , along the dimension a . If s is lower than the threshold, θ , all tokens within this window W can reuse the partial atten-

tion score of the first token, i.e., only the first token requires computation.

⑤ We then aggregate all reuse masks obtained along each axis and form a final binary mask. This mask identifies the tokens that can safely reuse partial attention scores.

⑥ Finally, we compute the partial attention scores for those tokens that cannot reuse other tokens' results via a sparse dot product, SparseAttention. The final output, O_i , is computed by a projection operation, ReusableProjection, which applies token reuse for those reusable tokens.

2. Experimental Configuration

We summarize the key hyperparameters and implementation details used in our experiments. All results are reported under consistent evaluation settings unless otherwise specified.

vDiT Model Configurations

We evaluate our method on three state-of-the-art video diffusion models: HunyuanVideo-T2V [5], Wan2.1-T2V-14B[6], CogVideoX1.5-5B-T2V [8] and , Open-Sora-Plan v1.2.0 [4]. Below we describe their architecture-related configurations.

HunyuanVideo is evaluated using 50 denoising steps. The input latent resolution is $33 \times 34 \times 60$ with a channel dimension of 128. The model adopts 24 attention heads. After decoding, the evaluation resolution is 133 frames $\times 544 \times 960$.

Wan2.1 is configured with 50 denoising steps. The input latent resolution is $21 \times 30 \times 52$ with a channel dimension of 128 and a setup of 12 attention heads. Its decoded output resolution is 81 frames $\times 480 \times 832$.

Open-Sora-Plan uses 50 denoising steps, with an input latent resolution of $8 \times 30 \times 40$ and a channel dimension of 96. It also employs 24 attention heads. The evaluation resolution is 29 frames $\times 480 \times 640$.

CogVideoX operates with 50 denoising steps. The input latent resolution is $11 \times 48 \times 85$ with a channel dimension of 128 and 16 attention heads. Its decoded output resolution is 129 frames $\times 768 \times 1360$.

Other Methods Settings

PAB [9] Since the 3D attention computation in PAB does not distinguish between temporal and spatial attention, we evaluate the configuration PAB59, where self-attention results are reused across 5 time steps and cross-attention results are reused across 9 time steps, while keeping full attention computation for the first 10 time steps to preserve generation consistency.

SVG [7] We adopt the configuration described in the original paper, using a sparsity ratio of 0.25 within each attention layer. Full attention computation is retained for the

first 10 time steps, as well as for the first 2.5% layer of each time step.

MINFERENCE [3] We follow the same configuration as Sparse Videogen but apply only the temporal masking strategy for comparison.

Δ -DiT [1] We preserve full attention for the 5 preceding and 5 succeeding time steps. The method is applied to the first 1/3 of layers during the first half of timesteps, and to the last 1/3 of layers during the second half of timesteps.

Evaluation Setting

All models are evaluated on 900 text prompts sampled from the validation split of the VBench benchmark [2]. For each prompt, we generate 5 video samples and compute the VBench score based on the average performance across these samples. In addition to the VBench score, we report several standard quantitative metrics, including PSNR, SSIM, and LPIPS.

All inference experiments are conducted on a single NVIDIA H100 GPU with 80GB of memory. To ensure reliable runtime measurements, we warm up the model with 10 iterations and then measure the inference time over the following 50 iterations, reporting the average runtime as the final result.

FLOPs Estimation

We estimate the theoretical cost of Transformer components to analyze the effect of attention reuse.

The dot products QK^\top and AV each require:

$$2BN^2d = 2BN^2H_{\text{total}}, \quad (120)$$

while softmax is negligible. Four linear projections (Q , K , V , and output) each with cost $2BNH_{\text{total}}^2$ sum to:

$$8BNH_{\text{total}}^2. \quad (123)$$

Thus, total FLOPs for a self-attention layer: (124)

$$\text{FLOPs}_{\text{self}} = 8BNH_{\text{total}}^2 + 2 \cdot 2BN^2H_{\text{total}} = 8BNH_{\text{total}}^2 + 4BN^2H_{\text{total}}. \quad (125)$$

With attention reuse, only a fraction α of tokens is active: (126)

$$\text{FLOPs}_{\text{sparse}} = 8BNH_{\text{total}}^2 + 4\alpha BN^2H_{\text{total}}. \quad (127)$$

We assume reused positions incur negligible cost; α is empirically estimated. (128)

Cross-attention FLOPs (query length N_q , key/value length N_{kv}): (130)

$$\text{FLOPs}_{\text{cross}} = 4B(N_q + N_{kv})H_{\text{total}}^2 + 4BN_qN_{kv}H_{\text{total}}. \quad (132)$$

Each MLP block contains two projections: (133)

$$\text{FLOPs}_{\text{mlp}} = 16BNH_{\text{total}}^2. \quad (134)$$

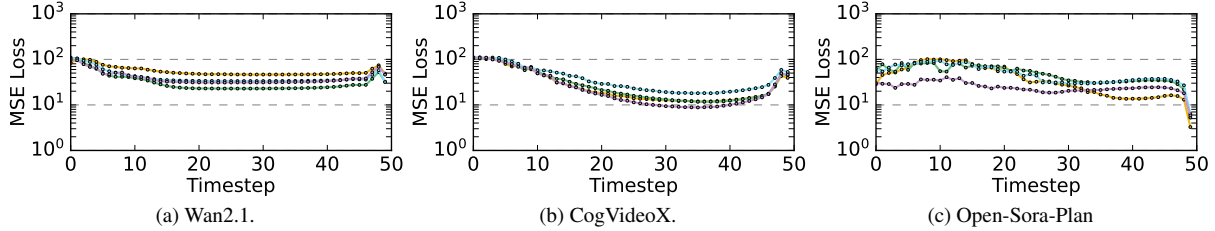


Fig. 1. Sensitivity of our method to the input prompt.

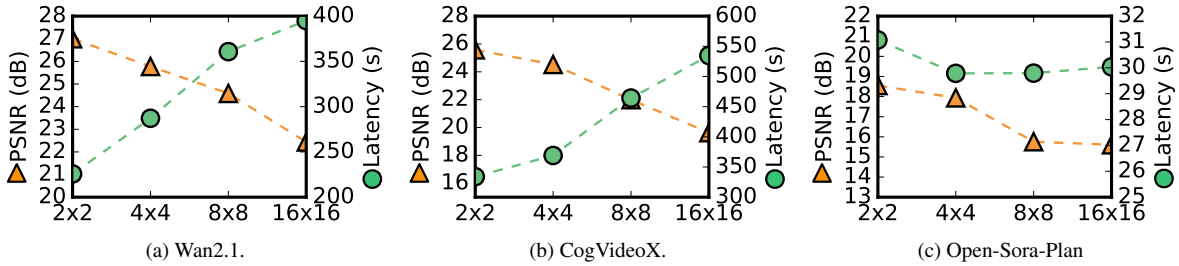


Fig. 2. Sensitivity of our method to the reuse window size.

3. Additional Sensitivity Analysis

In this section, we provide additional sensitivity analyses to support findings presented in the main paper. Specifically, we first show that the threshold setting is only sensitive to the timestep and insensitive to the input prompt in Sec. 3.1. Next, we verify the generality of our proposed reuse mechanism across different video generation models by varying different reuse window sizes in Sec. 3.2. Lastly, we show the sensitivity analysis of the individual channel-wise reuse threshold setting in Sec. 3.3.

3.1. Sensitivity to Input Prompts

In the main paper, we show that the threshold setting is independent of the specific input prompt for HunyuanVideo. Fig. 1 further gives the trends on other vDiT models, i.e., Wan2.1 [6], CogVideoX [8] and Open-Sora-Plan [4]. The experimental setup is the same as HunyuanVideo: we allow one specific timestep to reuse all tokens from the previous timestep and compare the MSE loss of the generated videos against the original generated videos. Overall, we show that the MSE curves of different prompts remain tightly clustered. This demonstrates that our dynamic threshold is “input prompt”-invariant, however, it is a function of timestep.

3.2. Sensitivity to Reuse Window Size

In the main paper, we show the trade-off of the reuse mechanism by varying the window size for HunyuanVideo. Here, we show the additional results on other vDiT models. Specifically, we conducted the sensitivity analysis on Wan2.1 [6], CogVideoX [8] and Open-Sora-Plan [4].

Fig. 2 shows the trade-off between latency (speedup)

Table 1. Sensitivity study of generation quality to dynamic thresholds.

| Model | Methods | Quality Metrics | | | |
|----------------|--|-------------------|----------------------|-----------------|--------------------|
| | | VBench \uparrow | PSNR (dB) \uparrow | SSIM \uparrow | LPIPS \downarrow |
| Wan2.1 | TIMERIPPLE _{80%} | 81.03 | 27.00 | 0.862 | 0.070 |
| | TIMERIPPLE _{80%} (channel wise) | 79.66 | 25.58 | 0.792 | 0.167 |
| CogVideoX | TIMERIPPLE _{80%} | 81.16 | 25.58 | 0.846 | 0.134 |
| | TIMERIPPLE _{80%} (channel wise) | 80.23 | 23.47 | 0.776 | 0.219 |
| Open-Sora-Plan | TIMERIPPLE _{80%} | 73.18 | 18.55 | 0.706 | 0.226 |
| | TIMERIPPLE _{80%} (channel wise) | 72.60 | 17.36 | 0.702 | 0.265 |

and generated quality (PSNR) across different models as the window size is varied. Across all vDiT models, as the window size increases, not only does the potential for reuse decrease, but the image quality also degrades noticeably, as shown in Fig. 2a and Fig. 2b. Although we find the latency decreases as the window size increases in Fig. 2c, the generated video quality also decreases significantly. Thus, we find that a window size of 2 offers a favorable trade-off, maintaining high efficiency without noticeable quality degradation caused by excessive token reuse.

3.3. Sensitivity to Channel-wise Reuse Threshold

In Table 4 of the main paper, we present the results of individually setting reuse thresholds for the channel dimension on HunyuanVideo. The goal of this experiment is to investigate whether adaptive, channel-specific thresholds could yield better results than a uniform setting. Here, we further show the results of other models.

Tbl. 1 shows the quantitative results of quality sensitivity to dynamic thresholding. Similar to what we configure for HunyuanVideo, we configure the threshold τ to be $\tau = \alpha \cdot \frac{1}{D} \sum_{i=1}^D |\Delta_i|$, where α is a base coefficient and Δ_i denotes the channel-wise difference. This way, we can set

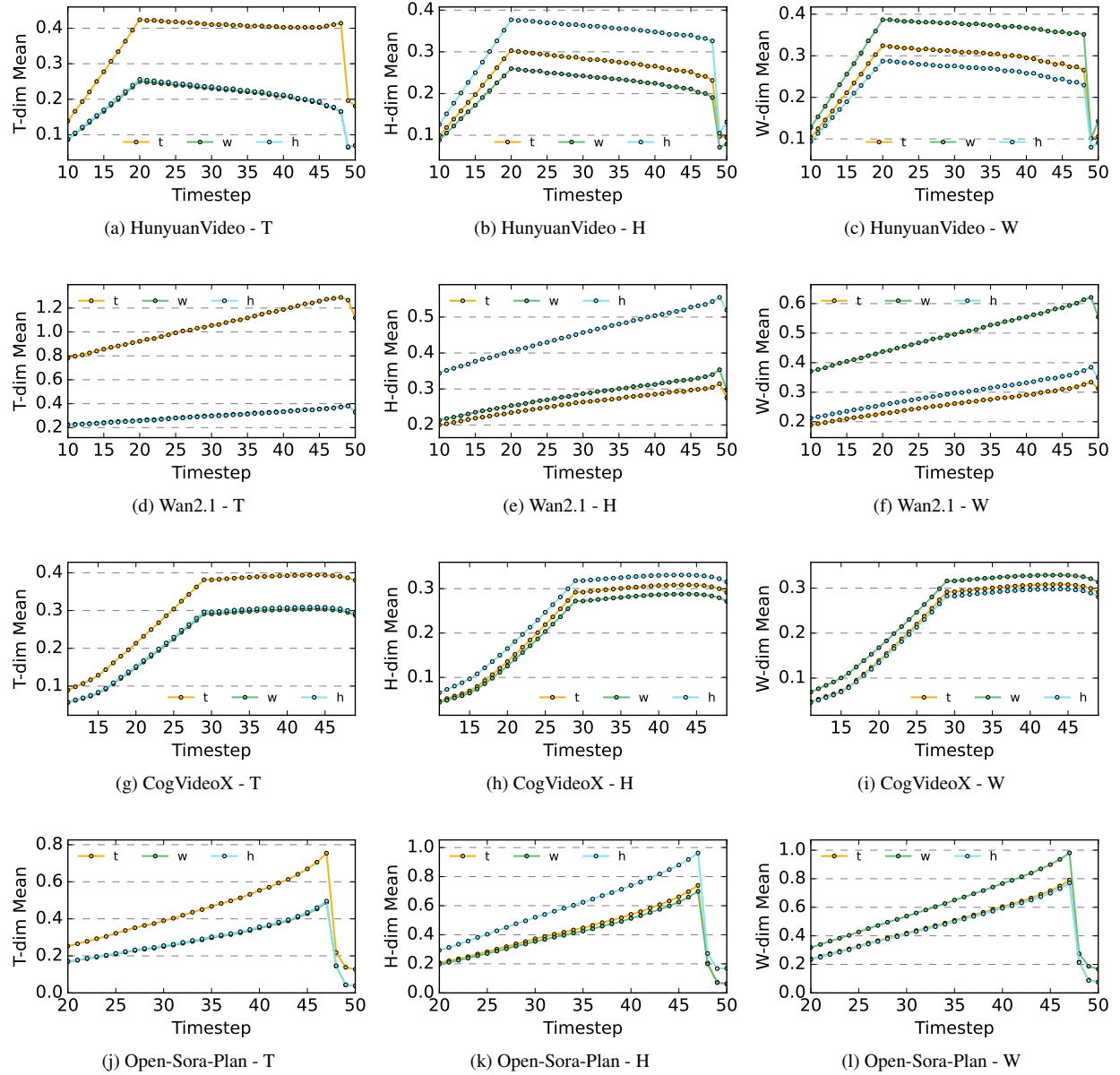


Fig. 3. The impact of varying adaptive threshold coefficients on different dimensions for four models. Each row represents a model, illustrating its performance across three distinct dimensions. The straight line represents the fitted trend.

different thresholds for x , y , and t dimensions (we denote it as “channel-wise”). Overall, we find that setting dedicated thresholds based on this strategy leads to lower generative quality. This further confirms that a uniform threshold setting is good enough for maintaining generation quality.

Lastly, Fig. 3 shows the actual values of dynamic thresholds across different timesteps. For instance, Fig. 3a shows the thresholds of different reuse directions in time-related channels (we call it “t-channel group”). Recall, the first 16 channels encode time-related information in HunyuanVideo. From the result, we can see that the reuse threshold

of a specific dimension is much higher when this dimension is the same as the corresponding channel group. Other two channels’ thresholds are often much lower. This means that, for t-channel group, it is better to reuse in x- or y-direction.

4. Full Performance Metrics

The remaining part of the supplementary material shows the detailed experimental results. To meet the supplementary material size constraints, all images are compressed using the JPEG format. Red circles are added to highlight regions where artifacts differ across methods.

Table 2. Individual VBench scores for HunyuanVideo model.

| Metric | Original | Δ -DiT | PAB _{5,9} | MINFERENCE | SVG _{70%} | TIMERIPPLE _{85%} | TIMERIPPLE _{75%} | TIMERIPPLE _{75%} +SVG _{70%} |
|------------------------|----------|---------------|--------------------|------------|--------------------|---------------------------|---------------------------|---|
| Subject Consistency | 0.9164 | 0.9143 | 0.9142 | 0.9216 | 0.9109 | 0.9316 | 0.9168 | 0.9128 |
| Motion Smoothness | 0.9901 | 0.9909 | 0.9915 | 0.9882 | 0.9886 | 0.9908 | 0.9904 | 0.9896 |
| Dynamic Degree | 0.8056 | 0.8056 | 0.8115 | 0.7857 | 0.7778 | 0.8571 | 0.8194 | 0.7639 |
| Aesthetic Quality | 0.6234 | 0.6206 | 0.6225 | 0.6133 | 0.6181 | 0.6093 | 0.6227 | 0.6176 |
| Imaging Quality | 0.6250 | 0.6290 | 0.5994 | 0.6412 | 0.6152 | 0.6181 | 0.6155 | 0.6072 |
| Overall Consistency | 0.2676 | 0.2674 | 0.2660 | 0.2581 | 0.2715 | 0.2536 | 0.2673 | 0.2710 |
| Background Consistency | 0.9633 | 0.9633 | 0.9585 | 0.9523 | 0.9583 | 0.9521 | 0.9618 | 0.9604 |
| Object Class | 0.6305 | 0.6915 | 0.6385 | 0.7109 | 0.7112 | 0.6523 | 0.6361 | 0.6867 |
| Multiple Objects | 0.5297 | 0.5427 | 0.3111 | 0.5078 | 0.5168 | 0.5352 | 0.5221 | 0.5328 |
| Color | 0.8790 | 0.8575 | 0.8836 | 0.7517 | 0.8690 | 0.8995 | 0.8878 | 0.8639 |
| Spatial Relationship | 0.6583 | 0.6767 | 0.6470 | 0.7378 | 0.6574 | 0.6580 | 0.6487 | 0.6322 |
| Scene | 0.2885 | 0.2929 | 0.1322 | 0.3309 | 0.3009 | 0.2831 | 0.2943 | 0.2958 |
| Temporal Style | 0.2367 | 0.2371 | 0.2333 | 0.2371 | 0.2421 | 0.2306 | 0.2367 | 0.2408 |
| Human Action | 0.8900 | 0.8800 | 0.8200 | 0.9000 | 0.9200 | 0.9000 | 0.8800 | 0.9400 |
| Temporal Flickering | 0.9861 | 0.9861 | 0.9883 | 0.9867 | 0.9852 | 0.9883 | 0.9863 | 0.9858 |
| Appearance Style | 0.1908 | 0.1898 | 0.1896 | 0.1922 | 0.1942 | 0.1882 | 0.1912 | 0.1938 |
| Quality Score | 0.8371 | 0.8373 | 0.8336 | 0.8343 | 0.8294 | 0.8395 | 0.8365 | 0.8286 |
| Semantic Score | 0.6657 | 0.6727 | 0.6107 | 0.6715 | 0.6812 | 0.6641 | 0.6651 | 0.6777 |
| Total Score | 0.8028 | 0.8043 | 0.7890 | 0.8018 | 0.7997 | 0.8044 | 0.8023 | 0.7984 |

Table 3. Individual VBench scores for Wan2.1 model.

| Metric | Original | Δ -DiT | PAB _{5,9} | MINFERENCE | SVG _{70%} | TIMERIPPLE _{85%} |
|------------------------|----------|---------------|--------------------|------------|--------------------|---------------------------|
| Subject Consistency | 0.9528 | 0.9512 | 0.9311 | 0.9424 | 0.9493 | 0.9522 |
| Motion Smoothness | 0.9771 | 0.9809 | 0.9806 | 0.9793 | 0.9803 | 0.9778 |
| Dynamic Degree | 0.7143 | 0.7857 | 0.5714 | 0.7857 | 0.7143 | 0.7143 |
| Aesthetic Quality | 0.6248 | 0.6087 | 0.6118 | 0.6218 | 0.6083 | 0.6263 |
| Imaging Quality | 0.6504 | 0.6641 | 0.6422 | 0.6640 | 0.6558 | 0.6517 |
| Overall Consistency | 0.2356 | 0.2420 | 0.2335 | 0.2427 | 0.2431 | 0.2351 |
| Background Consistency | 0.9750 | 0.9584 | 0.9551 | 0.9533 | 0.9708 | 0.9717 |
| Object Class | 0.9375 | 0.9492 | 0.9219 | 0.9063 | 0.9375 | 0.9258 |
| Multiple Objects | 0.5156 | 0.4531 | 0.5547 | 0.6172 | 0.6367 | 0.5273 |
| Color | 0.9727 | 0.9109 | 0.9063 | 0.9678 | 0.9844 | 0.9766 |
| Spatial Relationship | 0.8557 | 0.7290 | 0.7479 | 0.8256 | 0.8386 | 0.8219 |
| Scene | 0.2390 | 0.1544 | 0.1838 | 0.1728 | 0.2794 | 0.2132 |
| Temporal Style | 0.2349 | 0.2318 | 0.2252 | 0.2358 | 0.2353 | 0.2360 |
| Human Action | 0.7500 | 0.7000 | 0.8000 | 0.7000 | 0.7500 | 0.7500 |
| Temporal Flickering | 0.9895 | 0.9874 | 0.9849 | 0.9854 | 0.9892 | 0.9894 |
| Appearance Style | 0.2161 | 0.2184 | 0.2163 | 0.2195 | 0.2188 | 0.2157 |
| Quality Score | 0.8377 | 0.8402 | 0.8153 | 0.8379 | 0.8360 | 0.8376 |
| Semantic Score | 0.7079 | 0.6662 | 0.6857 | 0.7011 | 0.7296 | 0.7011 |
| Total Score | 0.8117 | 0.8054 | 0.7894 | 0.8105 | 0.8147 | 0.8103 |

Table 4. Individual VBench scores for CogVideoX model.

| Metric | Original | Δ -DiT | PAB _{5,9} | MINFERENCE | SVG _{70%} | TIMERIPPLE _{85%} |
|------------------------|----------|---------------|--------------------|------------|--------------------|---------------------------|
| Subject Consistency | 0.9547 | 0.9517 | 0.9650 | 0.8991 | 0.9422 | 0.9446 |
| Motion Smoothness | 0.9813 | 0.9882 | 0.9867 | 0.9612 | 0.9776 | 0.9745 |
| Dynamic Degree | 0.7361 | 0.5694 | 0.6500 | 0.7857 | 0.8571 | 0.8571 |
| Aesthetic Quality | 0.5936 | 0.5436 | 0.2719 | 0.4929 | 0.5716 | 0.5992 |
| Imaging Quality | 0.6636 | 0.5810 | 0.2417 | 0.5517 | 0.6712 | 0.6719 |
| Overall Consistency | 0.2568 | 0.2486 | 0.0375 | 0.2299 | 0.2310 | 0.2409 |
| Background Consistency | 0.9678 | 0.9656 | 0.9611 | 0.9609 | 0.9614 | 0.9612 |
| Object Class | 0.8299 | 0.7801 | 0.7764 | 0.7109 | 0.7930 | 0.8086 |
| Multiple Objects | 0.5945 | 0.4977 | 0.3555 | 0.3672 | 0.6172 | 0.5664 |
| Color | 0.8961 | 0.8866 | 0.9286 | 0.9029 | 0.9336 | 0.9297 |
| Spatial Relationship | 0.7039 | 0.6152 | 0.6559 | 0.6340 | 0.7208 | 0.7015 |
| Scene | 0.3525 | 0.2333 | 0.3547 | 0.2463 | 0.4338 | 0.4191 |
| Temporal Style | 0.2312 | 0.2174 | 0.0373 | 0.2157 | 0.2249 | 0.2340 |
| Human Action | 0.8500 | 0.8200 | 0.8000 | 0.5500 | 0.8500 | 0.9000 |
| Temporal Flickering | 0.9773 | 0.9793 | 0.9832 | 0.9777 | 0.9684 | 0.9694 |
| Appearance Style | 0.2321 | 0.2268 | 0.1999 | 0.2266 | 0.2279 | 0.2230 |
| Quality Score | 0.8326 | 0.8028 | 0.7173 | 0.7818 | 0.8305 | 0.8340 |
| Semantic Score | 0.7174 | 0.6619 | 0.5392 | 0.6091 | 0.7214 | 0.7223 |
| Total Score | 0.8095 | 0.7746 | 0.6817 | 0.7473 | 0.8087 | 0.8116 |

Table 5. Individual VBench scores for Open-Sora-Plan model.

| Metric | Original | Δ -DiT | PAB _{5,9} | MINFERENCE | SVG _{70%} | TIMERIPPLE _{85%} |
|------------------------|----------|---------------|--------------------|------------|--------------------|---------------------------|
| Subject Consistency | 0.9264 | 0.9123 | 0.9231 | 0.9478 | 0.9534 | 0.9269 |
| Motion Smoothness | 0.9894 | 0.9473 | 0.9901 | 0.9916 | 0.9932 | 0.9892 |
| Dynamic Degree | 0.5417 | 0.5459 | 0.5694 | 0.3571 | 0.3889 | 0.5972 |
| Aesthetic Quality | 0.5051 | 0.521 | 0.5089 | 0.4893 | 0.5117 | 0.5190 |
| Imaging Quality | 0.5744 | 0.5570 | 0.5599 | 0.5484 | 0.5844 | 0.5912 |
| Overall Consistency | 0.2146 | 0.2383 | 0.2118 | 0.2010 | 0.2068 | 0.2129 |
| Background Consistency | 0.9701 | 0.9257 | 0.9669 | 0.9696 | 0.9724 | 0.9591 |
| Object Class | 0.4328 | 0.5095 | 0.4644 | 0.3438 | 0.4509 | 0.4905 |
| Multiple Objects | 0.1928 | 0.3251 | 0.1631 | 0.1289 | 0.2889 | 0.2012 |
| Color | 0.7213 | 0.5791 | 0.7596 | 0.8132 | 0.7358 | 0.7029 |
| Spatial Relationship | 0.4040 | 0.4018 | 0.4247 | 0.4803 | 0.4638 | 0.4158 |
| Scene | 0.0218 | 0.1524 | 0.0182 | 0.0000 | 0.0574 | 0.0073 |
| Temporal Style | 0.1940 | 0.1420 | 0.1854 | 0.1944 | 0.1915 | 0.1903 |
| Human Action | 0.3000 | 0.5356 | 0.3200 | 0.2000 | 0.3200 | 0.3600 |
| Temporal Flickering | 0.9881 | 0.6397 | 0.9904 | 0.9893 | 0.9949 | 0.9884 |
| Appearance Style | 0.2285 | 0.1482 | 0.2267 | 0.2308 | 0.2277 | 0.2279 |
| Quality Score | 0.7944 | 0.7696 | 0.7955 | 0.7784 | 0.7954 | 0.8013 |
| Semantic Score | 0.4444 | 0.4323 | 0.4487 | 0.4289 | 0.4689 | 0.4538 |
| Total Score | 0.7244 | 0.7021 | 0.7261 | 0.7085 | 0.7301 | 0.7318 |

Prompt: “a boat accelerating to gain speed”



Fig. 4. The qualitative comparison of TIMERIPPLE against other methods on HunyuanVideo

Prompt: “a bear catching a salmon in its powerful jaws”

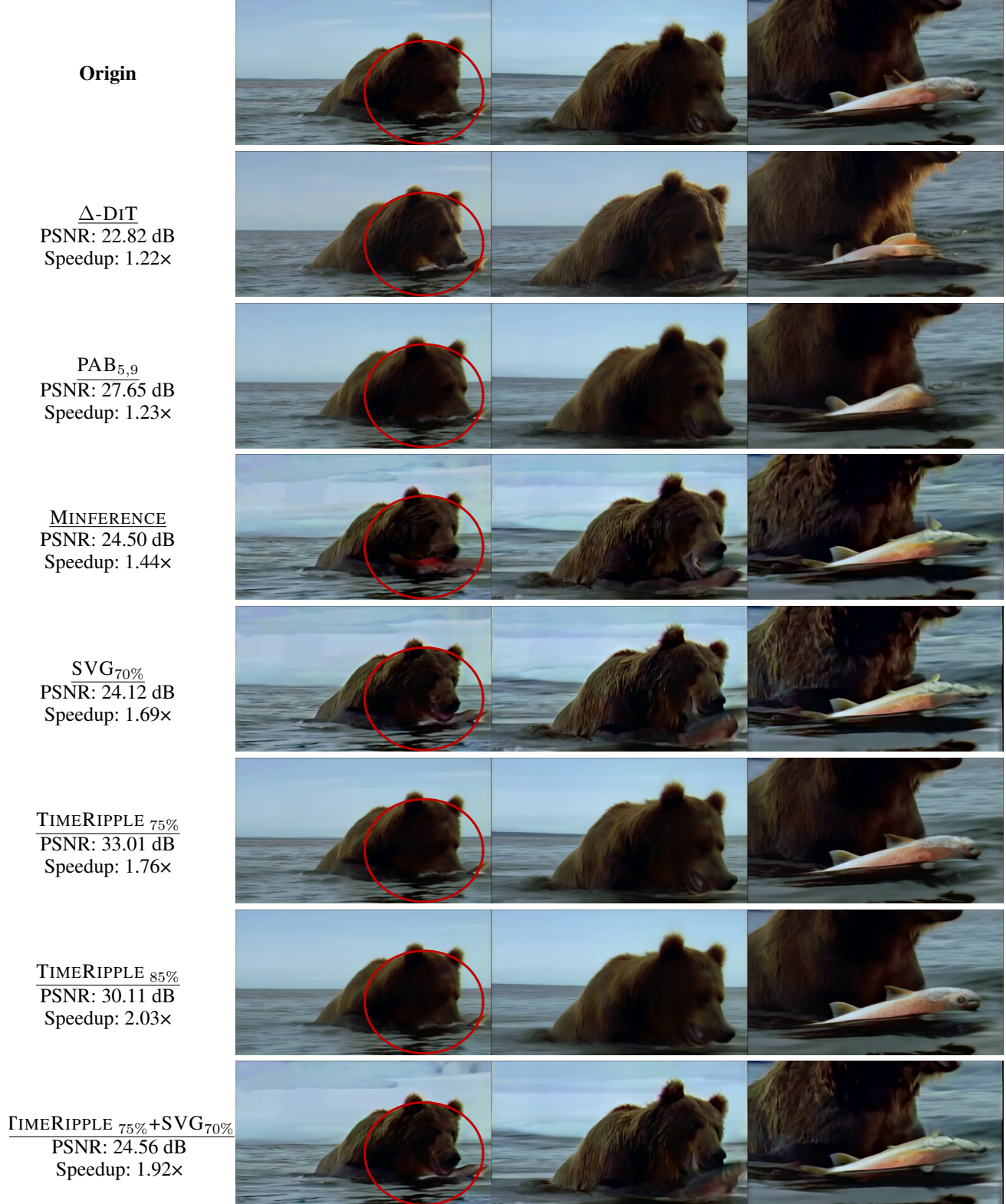


Fig. 5. The qualitative comparison of TIMERIPPLE against other methods on HunyuanVideo

Prompt: “A boat sailing leisurely along the Seine River with the Eiffel Tower in background, tilt down”

Origin



Δ -DiT
PSNR: 26.37 dB
Speedup: 1.22x



PAB_{5.9}
PSNR: 25.25 dB
Speedup: 1.23x



MINFERENCE
PSNR: 18.93 dB
Speedup: 1.44x



SVG_{70%}
PSNR: 19.83 dB
Speedup: 1.69x



TIMERIPPLE_{75%}
PSNR: 30.59 dB
Speedup: 1.76x



TIMERIPPLE_{85%}
PSNR: 27.64 dB
Speedup: 2.03x



TIMERIPPLE_{75%}+SVG_{70%}
PSNR: 20.02 dB
Speedup: 1.92x



Fig. 6. The qualitative comparison of TIMERIPPLE against other methods on HunyuanVideo

Prompt: “Aerial panoramic video from a drone of a fantasy land.”

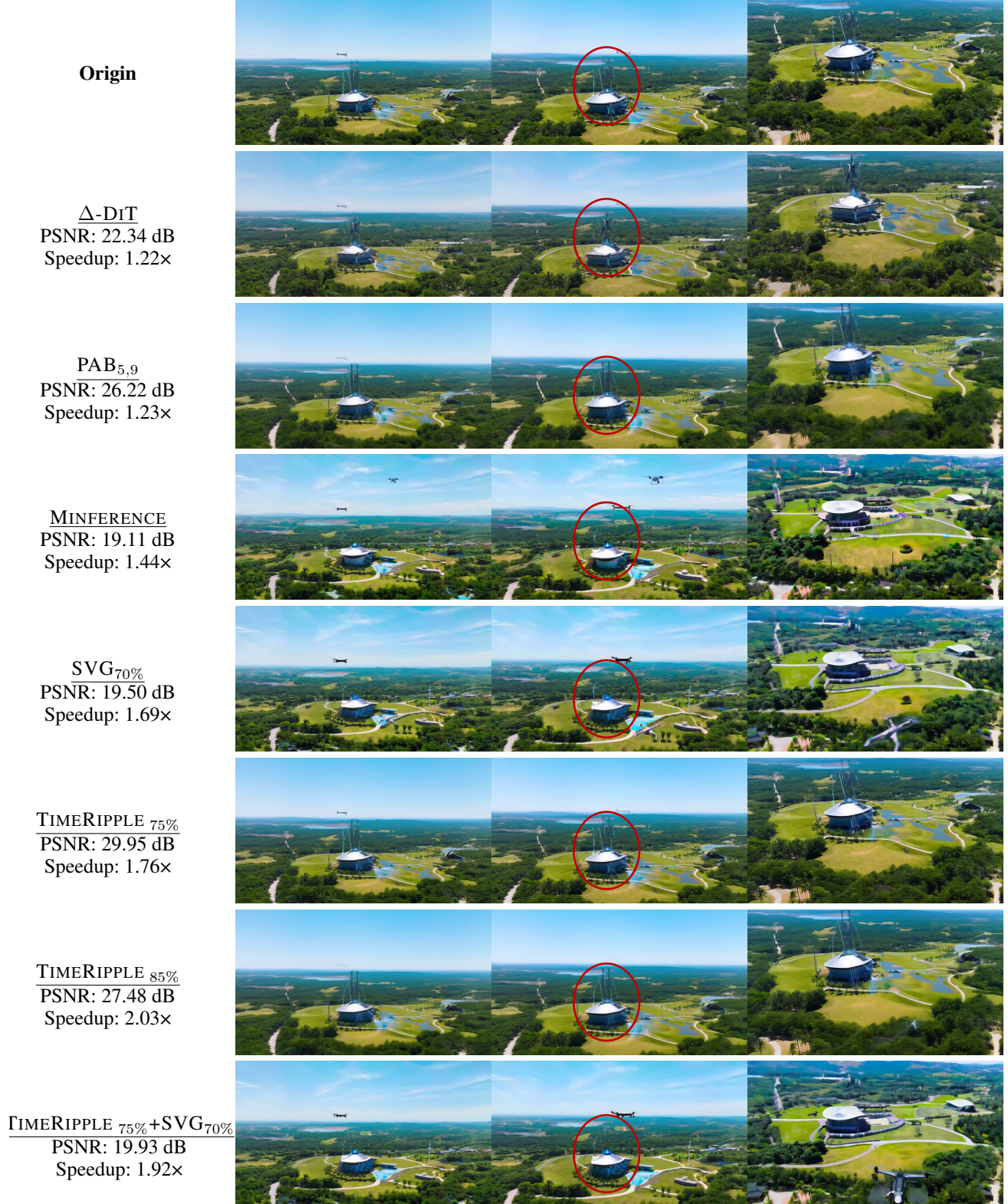


Fig. 7. The qualitative comparison of TIMERIPPLE against other methods on HunyuanVideo

Prompt: “A beautiful coastal beach in spring, waves lapping on sand by Vincent van Gogh”

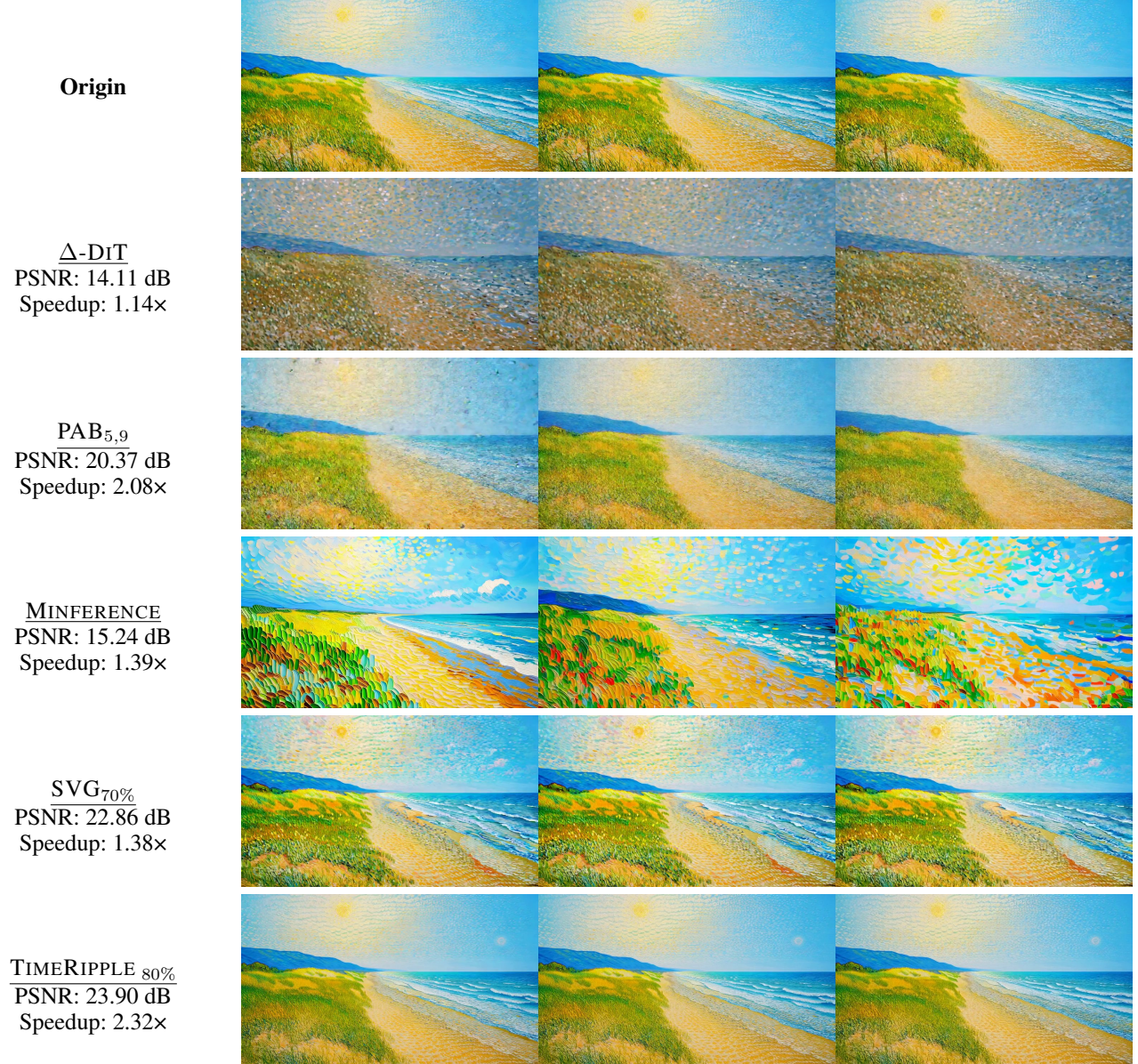


Fig. 8. The qualitative comparison of TIMERIPPLE against other methods on Wan2.1

Prompt: “a train speeding down the tracks”

Origin

Δ -DiT
PSNR: 16.33 dB
Speedup: 1.14×

PAB_{5,9}
PSNR: 22.16 dB
Speedup: 2.08×

MINFERENCE
PSNR: 19.68 dB
Speedup: 1.39×

SVG_{70%}
PSNR: 20.43 dB
Speedup: 1.38×

TIMERIPPLE_{80%}
PSNR: 25.79 dB
Speedup: 2.32×



Fig. 9. The qualitative comparison of TIMERIPPLE against other methods on Wan2.1

Prompt: "A person is skateboarding"

Origin

Δ -DiT
PSNR: 14.18 dB
Speedup: 1.14x

PAB_{5,9}
PSNR: 19.29 dB
Speedup: 2.07x

MINFERENCE
PSNR: 16.87 dB
Speedup: 1.39x

SVG_{70%}
PSNR: 17.05 dB
Speedup: 1.38x

TIMERIPPLE_{80%}
PSNR: 21.80 dB
Speedup: 2.32x



Fig. 10. The qualitative comparison of TIMERIPPLE against other methods on Wan2.1

Prompt: “Aerial panoramic video from a drone of a fantasy land.”

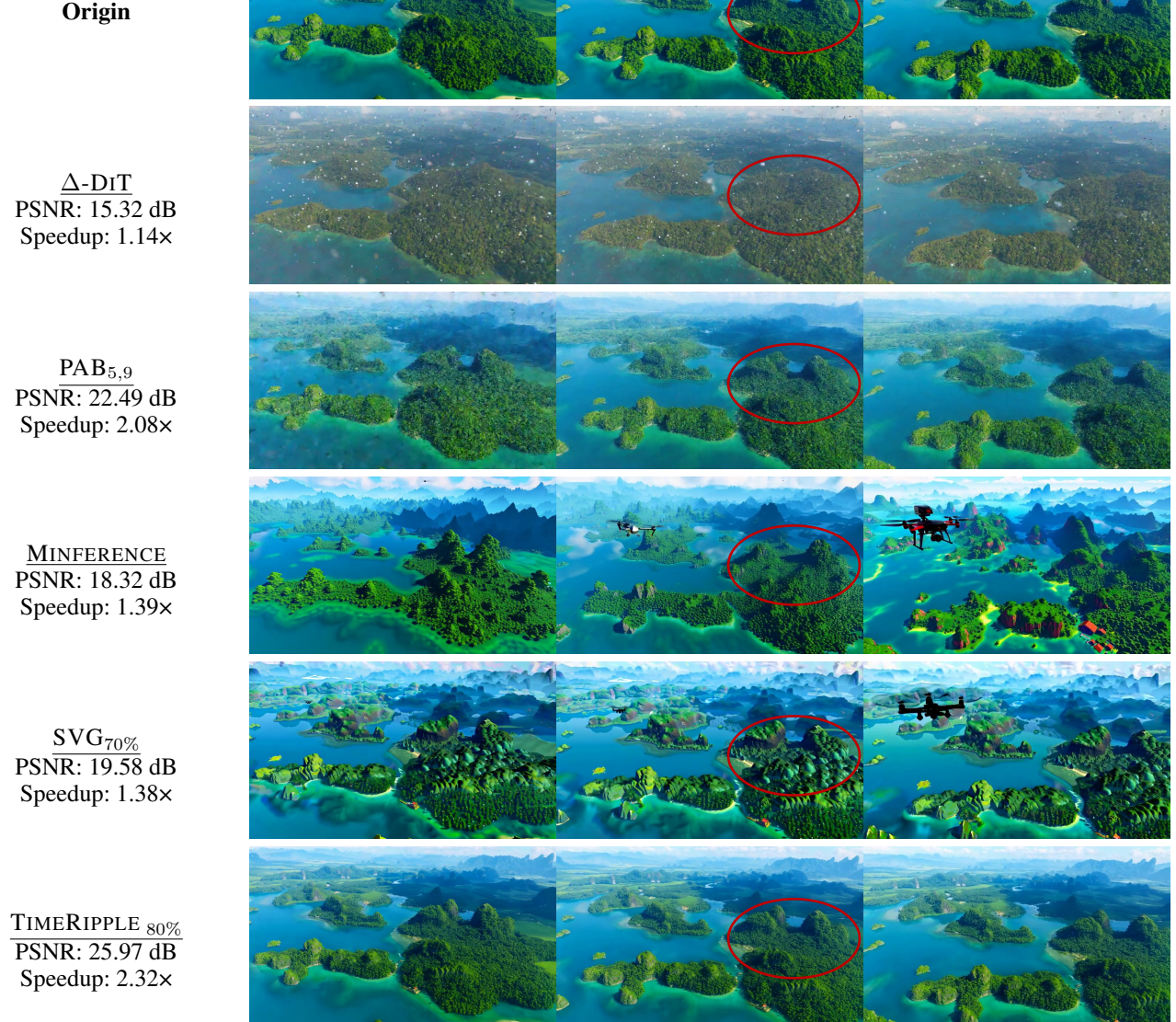


Fig. 11. The qualitative comparison of TIMERIPPLE against other methods on Wan2.1

Prompt: “A beautiful coastal beach in spring, waves lapping on sand by Vincent van Gogh”

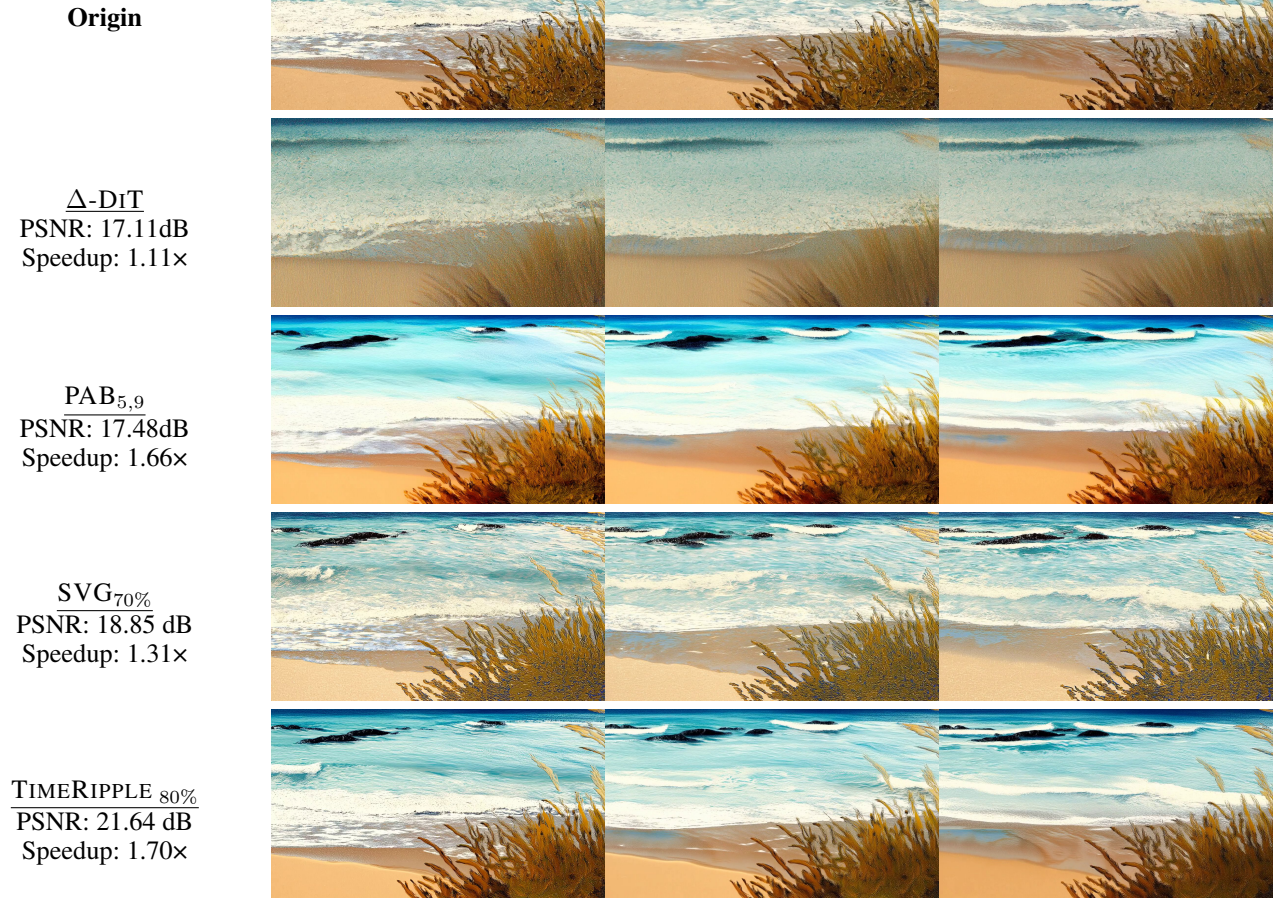


Fig. 12. The qualitative comparison of TIMERIPPLE against other methods on CogVideoX

Prompt: “A person is skateboarding”

Origin



Δ -DiT
PSNR: 15.43 dB
Speedup: 1.11×



PAB_{5,9}
PSNR: 16.33 dB
Speedup: 1.66×



SVG_{70%}
PSNR: 25.29 dB
Speedup: 1.31×



TIMERIPPLE_{80%}
PSNR: 27.08 dB
Speedup: 1.70×



Fig. 13. The qualitative comparison of TIMERIPPLE against other methods on CogVideoX

Prompt: “a train speeding down the tracks”

Origin

Δ -DiT
PSNR: 19.43 dB
Speedup: 1.11×

$\text{PAB}_{5,9}$
PSNR: 18.72 dB
Speedup: 1.66×

$\text{SVG}_{70\%}$
PSNR: 22.58 dB
Speedup: 1.31×

$\text{TIMERIPPLE}_{80\%}$
PSNR: 26.95 dB
Speedup: 1.70×

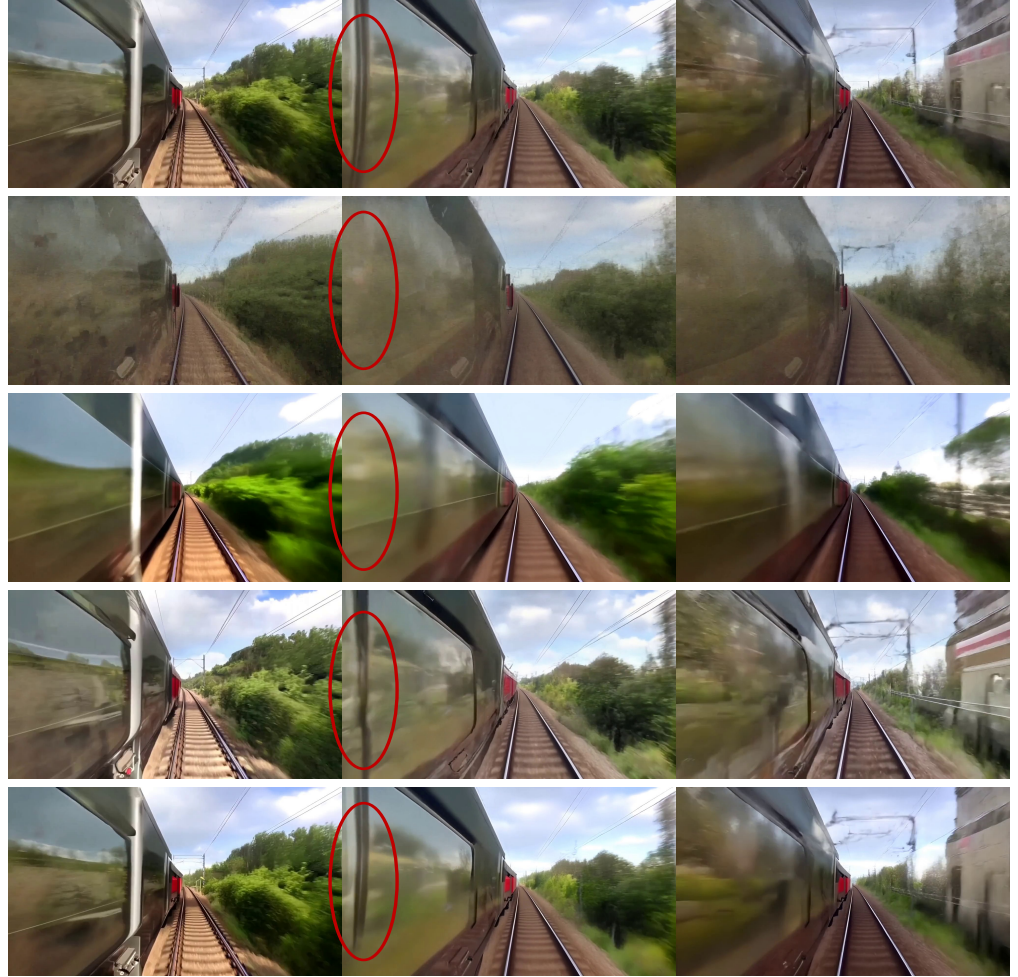


Fig. 14. The qualitative comparison of TIMERIPPLE against other methods on CogVideoX

Prompt: “Aerial panoramic video from a drone of a fantasy land.”

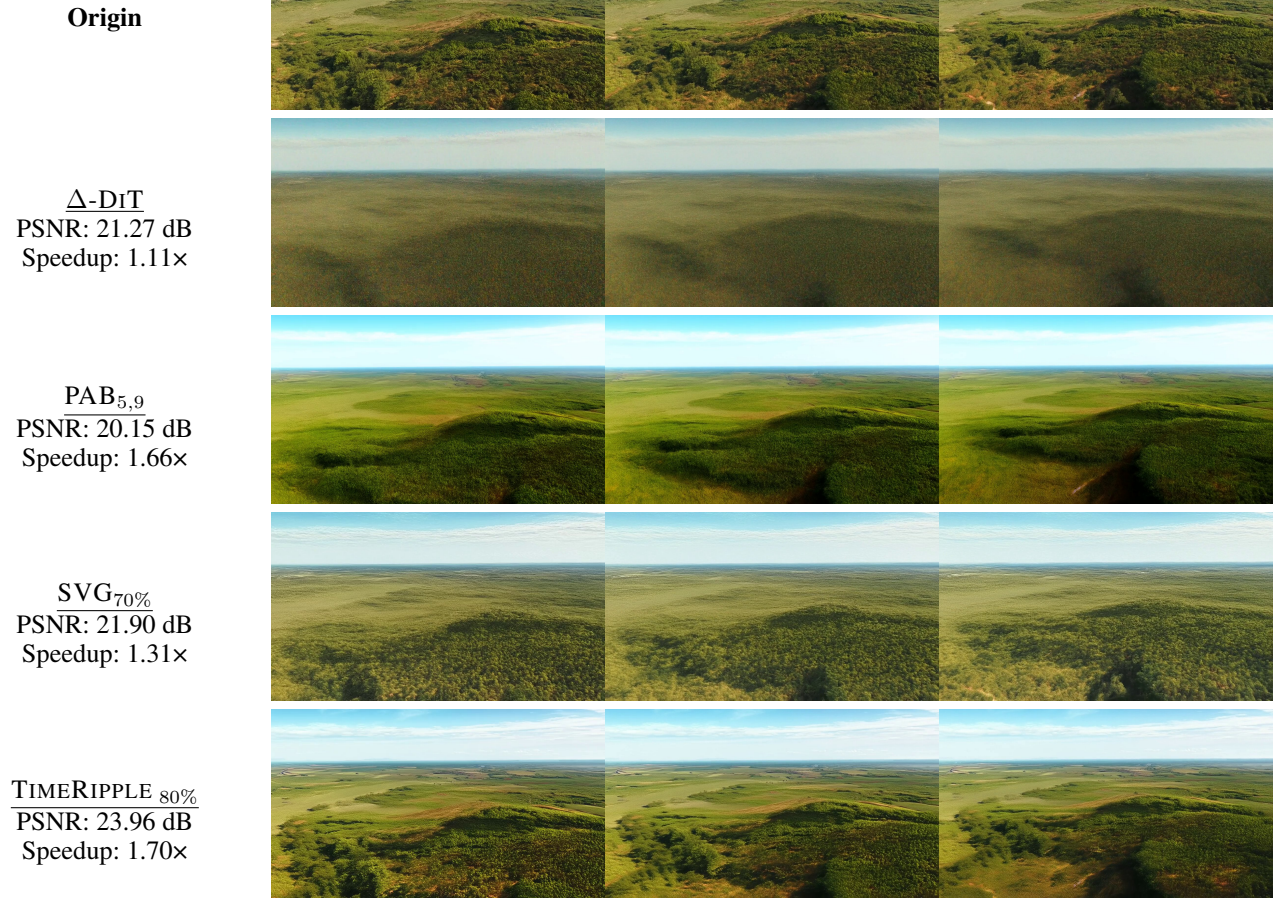


Fig. 15. The qualitative comparison of TIMERIPPLE against other methods on CogVideoX

References

- [1] Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Delta-dit: A training-free acceleration method tailored for diffusion transformers. *arXiv preprint arXiv:2406.01125*, 2024. 2
- [2] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 2
- hui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024. 2
- [4] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. 2, 3
- [5] Tencent. Tencent launches and open-sources Hunyuan video-generation model, 2024. 2
- [6] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wenten Wang, Wenting Shen, Wenyuan Yu, Xi-anzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 3
- [7] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025. 2
- [8] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 3
- [9] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024. 2
- [3] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qian-