

A Pseudocode: Lexicographic Minimax Path Selection

Below we present the full pseudocode implementation of the lexicographic minimax path selection algorithm, as introduced in Section 3.3. The algorithm leverages dynamic programming to efficiently compute an optimal caching path from source s to target t under a step budget constraint B . Unlike standard shortest-path methods, our approach handles the non-additive and non-Markovian nature of error accumulation by minimizing edge weights in a lexicographically ordered manner.

Algorithm 1 Lexicographic Minimax Path Selection

```

1: Input: Directed acyclic graph  $G = (V, E)$ , start node  $s$ , end node  $t$ , step limit  $B$ 
2: Output: Lexicographic Minimax Path  $P^*$ 
3: Initialization:
4:    $dp[v][k]$ : maximum edge weight on any  $k$ -step path to  $v$ 
5:    $paths[v][k], edges[v][k]$ : corresponding node and edge sequences
6:    $dp[s][0] \leftarrow 0, paths[s][0] \leftarrow [[s]], edges[s][0] \leftarrow [[]]$ 
7: Main Loop:
8: for  $k = 0$  to  $B - 1$  do
9:   for each node  $v$  with  $dp[v][k] < \infty$  do
10:    for each neighbor  $u$  of  $v$  do
11:       $w \leftarrow$  weight of edge  $(v, u)$ 
12:       $m \leftarrow \max(dp[v][k], w)$ 
13:      if  $m < dp[u][k + 1]$  then
14:         $dp[u][k + 1] \leftarrow m$ 
15:        Update  $paths[u][k + 1], edges[u][k + 1]$  from  $v$ 
16:      else if  $m = dp[u][k + 1]$  then
17:        Append new paths and edges from  $v$  to  $paths[u][k + 1], edges[u][k + 1]$ 
18:      end if
19:    end for
20:  end for
21: end for
22: Final Selection:
23:  $P^* \leftarrow \min(\text{zip}(paths[t][B], edges[t][B]), \text{key} = \lambda(p, e) : \text{sorted}(e, \text{reverse}=\text{True}))$ 

```

B Experiment Settings

B.1 Models

In this paper, we introduce LeMiCa, a novel caching technique designed to accelerate and enhance a range of state-of-the-art video synthesis models, including Open-Sora 1.2 [52], Latte [23], and CogVideoX [44]. Open-Sora 1.2 integrates 2D/3D VAEs and ST-DiT blocks for efficient video compression and generation. Latte leverages spatio-temporal tokenization and Transformer layers to model video distributions in the latent space. CogVideoX employs a 3D VAE and expert Transformers with adaptive LayerNorm for modality fusion and high-fidelity generation. In our experiments, we adopt the CogVideoX-2B variant.

B.2 Details of the Compared Methods

PAB introduces a pyramid-style broadcasting mechanism to reduce redundant attention computations in diffusion models. By observing a U-shaped pattern in attention differences across steps, PAB applies adaptive broadcast strategies based on the variance of different attention types (e.g., spatial, temporal, cross-modal). Stable attention outputs are efficiently reused in later steps, reducing computation. All experiments use PAB’s default parameter settings.

TeaCache is a training-free, architecture-agnostic caching method that exploits the correlation between timestep embedding changes and model output differences across adjacent steps. By

introducing a unified threshold-based strategy, TeaCache decides when to activate caching through an accumulated error-based discriminator. Since this method operates solely along the temporal dimension without modifying specific model components, it offers strong generalization and broad applicability.

B.3 Model Forward Steps

Model Forward Steps. In this work, we control the acceleration efficiency of LeMiCa via the Model Forward Steps B . Smaller values of B reduce the denoising time, leading to higher speed-up ratios. We consider two variants: LeMiCa-slow, which emphasizes visual fidelity, and LeMiCa-fast, which prioritizes inference efficiency. The corresponding B values for each variant across different models are listed in Table 4.

Table 4: Model forward steps B under different configurations.

Model	Configuration	Model Forward Steps B
Open-Sora 1.2	Original	30
	LeMiCa-slow	19
	LeMiCa-fast	11
Latte	Original	50
	LeMiCa-slow	27
	LeMiCa-fast	14
CogVideoX	Original	50
	LeMiCa-slow	27
	LeMiCa-fast	16

C More Visual Results

We present additional visual comparisons across three foundational models: Open-Sora [52], Latte [23], and CogVideoX [44]. Results are grouped into two settings: fidelity-focused and speed-focused.

C.1 Fidelity-Focused

We perform frame-by-frame comparisons to assess fine-grained differences in quality (LeMiCa-slow vs. TeaCache-slow). Since this setting uses relatively low acceleration ratios, artifacts are less obvious in real-time playback. To address this, we extract representative frames that highlight detail preservation, object integrity, and temporal consistency. As shown in Figures 8, 9, 10, 11, and 12, our method consistently produces more coherent results across all baselines.

C.2 Speed-Focused

To evaluate robustness under aggressive acceleration, we compare videos generated with higher speed-up ratios (LeMiCa-fast vs. TeaCache-fast). This setting is designed to prioritize generation speed without significantly compromising visual quality. Under such conditions, baseline methods are more prone to issues such as flickering, object drift, and reduced temporal consistency. In contrast, our method maintains strong temporal and semantic coherence, even at high generation speeds.

As part of the supplementary material, we include the following video files: **Speed-Focused Open-Sora.mp4**, **Speed-Focused Latte.mp4**, and **Speed-Focused CogVideoX.mp4**.

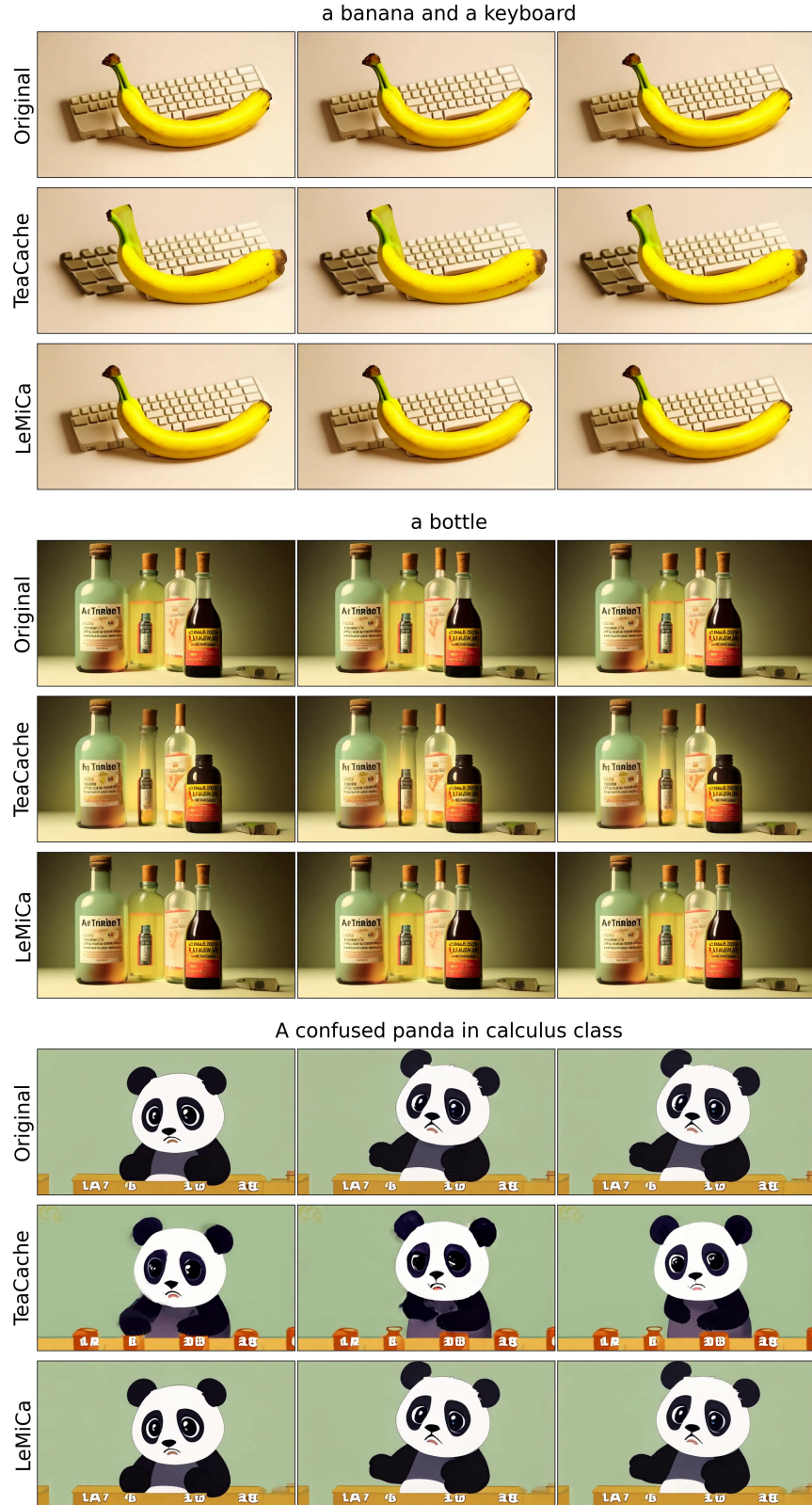


Figure 8: More visual results on Open-Sora (Part I).

a teddy bear on the right of a potted plant, front view



A tranquil tableau of a picturesque barn



A tranquil tableau of an apple

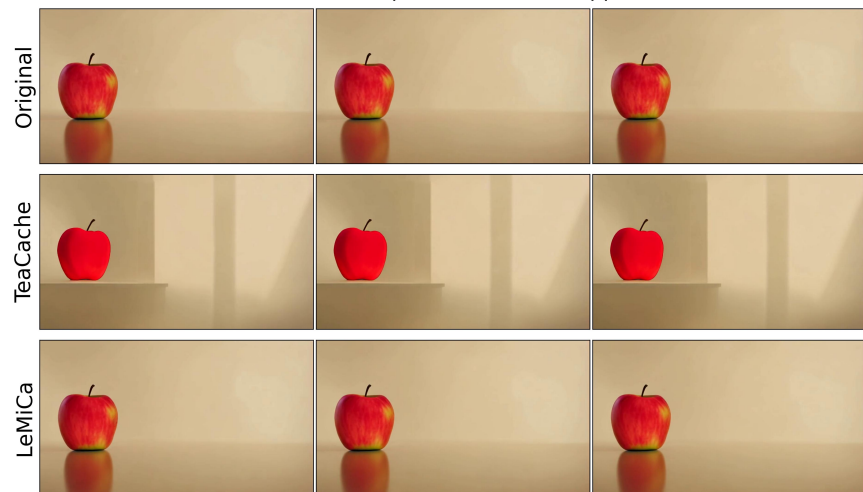


Figure 9: More visual results on Open-Sora (Part II).

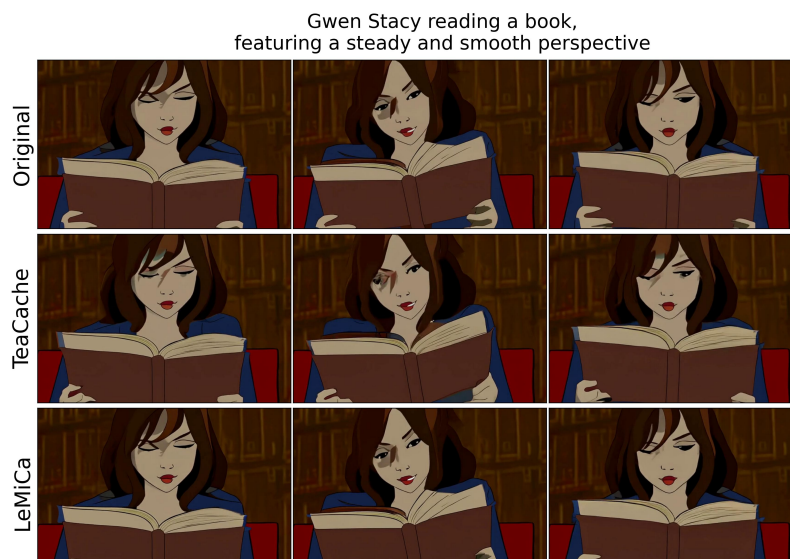
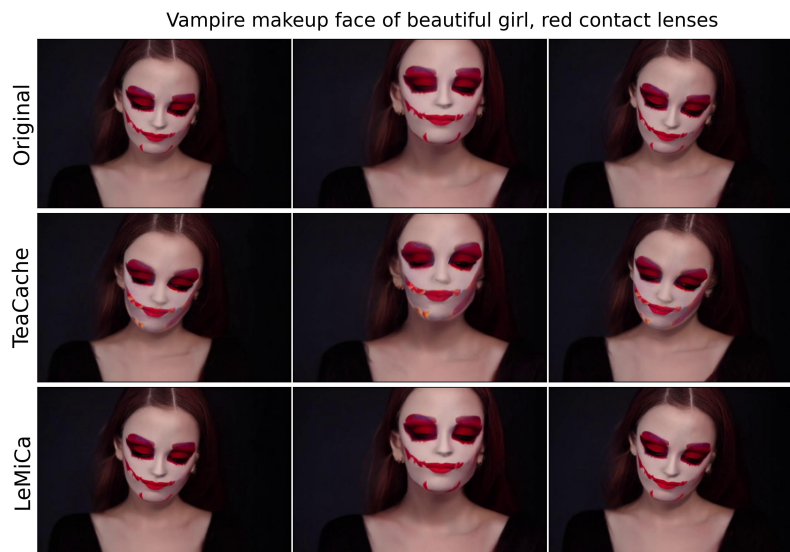


Figure 10: More visual results on CogVideoX.

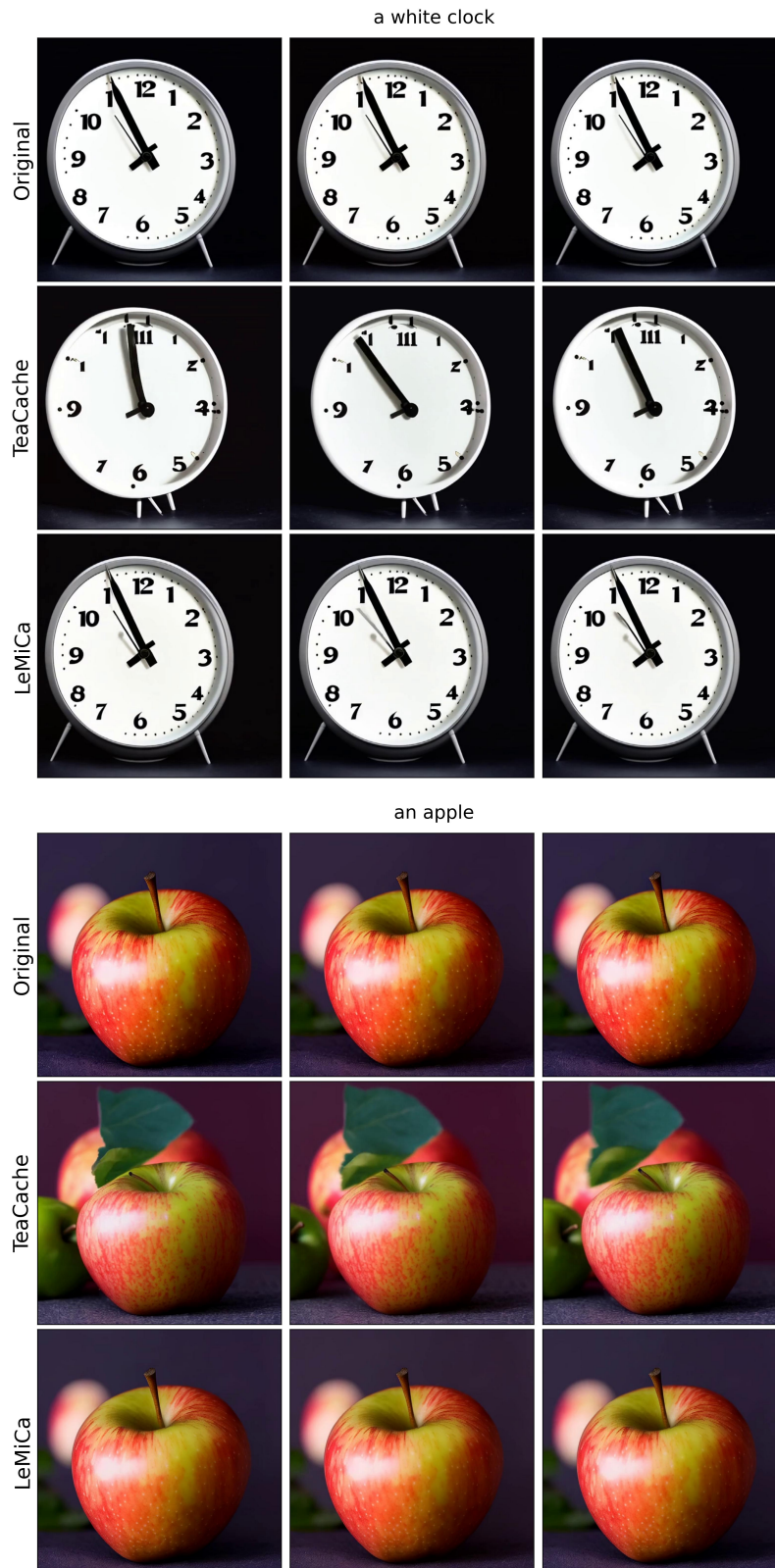


Figure 11: More visual results on Latte (Part I).

An astronaut is riding a horse in the space in a photorealistic style



A boat sailing leisurely along the Seine River with the Eiffel Tower in background, surrealism style

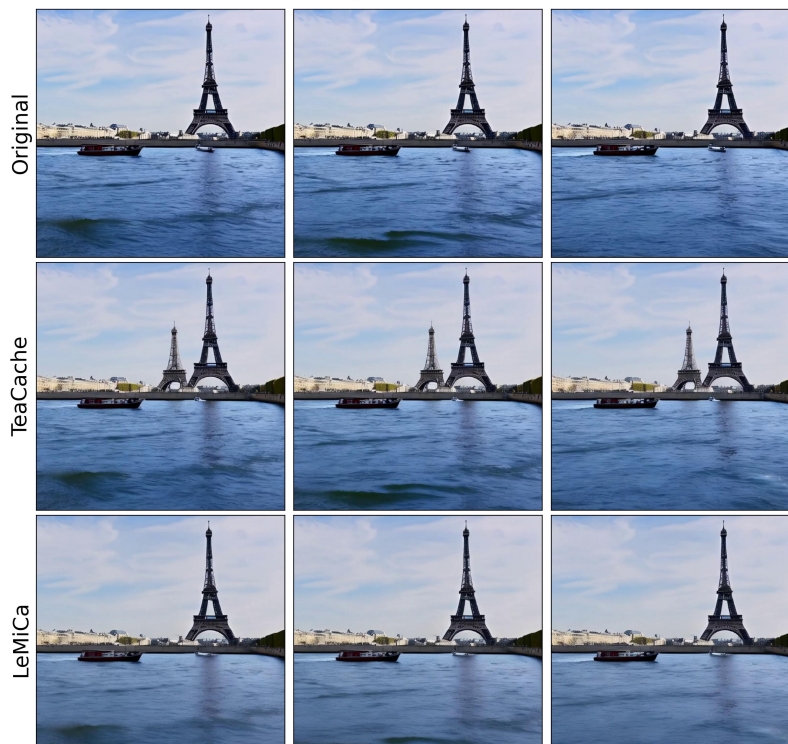


Figure 12: More visual results on Latte (Part II).

D Limitation

Although our method achieves strong performance in both acceleration and video fidelity, it still has certain limitations. First, when the original video quality is low, particularly in scenarios involving complex motion dynamics, it struggles to consistently generate satisfactory results. This reflects a dependency on the representational capacity of the underlying diffusion model. Second, under high acceleration ratios, some degree of quality degradation remains inevitable due to the significantly reduced number of model forward steps. We believe that continued progress in foundational video generation models will help alleviate these issues. Moreover, since our approach focuses solely on temporal step scheduling and is agnostic to model architecture, it can be quickly adapted to future, more powerful diffusion models.

E Social Impact

Diffusion-based video generation models are often limited by high inference time and computational cost. Our method alleviates this by significantly improving efficiency without requiring additional training. This enables broader access to high-quality video synthesis, particularly in resource-constrained settings. By reducing computation during the inference process, our approach also lowers energy use and carbon emissions, contributing to more sustainable AI development. Furthermore, we will release our code to support future research.