LeMiCa: Lexicographic Minimax Path Caching for Efficient Diffusion-Based Video Generation

Huanlin Gao[1,2*] **Ping Chen**[1,2*] **Fuyuan Shi**[1,2] **Chao Tan**[1,2] **Zhaoxiang Liu**[1,2]
Fang Zhao[1,2†] **Kai Wang**[1,2] **Shiguo Lian**[1,2†]
Data Science & Artificial Intelligence Research Institute, China Unicom[1]
Unicom Data Intelligence, China Unicom[2]
{gaohl51, chenp181, shify15, tanc10, liuzx178, zhaof50, wangk115, liansg}@chinaunicom.cn
https://unicomai.github.io/LeMiCa

Abstract

We present LeMiCa, a training-free and efficient acceleration framework for diffusion-based video generation. While existing caching strategies primarily focus on reducing local heuristic errors, they often overlook the accumulation of global errors, leading to noticeable content degradation between accelerated and original videos. To address this issue, we formulate cache scheduling as a directed graph with error-weighted edges and introduce a Lexicographic Minimax Path Optimization strategy that explicitly bounds the worst-case path error. This approach substantially improves the consistency of global content and style across generated frames. Extensive experiments on multiple text-to-video benchmarks demonstrate that LeMiCa delivers dual improvements in both inference speed and generation quality. Notably, our method achieves a 2.9× speedup on the Latte model and reaches an LPIPS score of 0.05 on Open-Sora, outperforming prior caching techniques. Importantly, these gains come with minimal perceptual quality degradation, making LeMiCa a robust and generalizable paradigm for accelerating diffusion-based video generation. We believe this approach can serve as a strong foundation for future research on efficient and reliable video synthesis. Our code is available at https://github.com/UnicomAI/LeMiCa

1 Introduction

Diffusion models [10, 38] have made significant advancements in video generation [24, 53, 45], particularly with DiT-based architectures [29], which greatly enhance visual quality. However, these methods are often hindered by high memory usage, substantial computational costs, and long inference latencies, limiting their use in interactive applications. This has led to increased interest in more efficient and cost-effective generation strategies.

Existing approaches such as model distillation [39, 30, 42], pruning [7, 27], and quantization [34, 37, 8, 18] have been widely adopted to accelerate inference. While effective, these methods require careful architectural design and retraining on large datasets, incurring high costs. Caching mechanisms [35, 26], in contrast, offer a retraining-free alternative for accelerating diffusion model inference. The core idea is to reuse model outputs from specific timesteps during sampling to reduce redundant computations and speed up the process [20, 44]. Selecting optimal cache timesteps, while balancing video quality and inference speed, remains an open problem in video generation.

Ideally, a lossless video acceleration method should meet two essential criteria: **(i) High visual quality** and **(ii) Consistency between accelerated and original videos**. However, existing cache-

[*]Equal contribution
[†]Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

based methods [20, 51] maintain a certain level of visual quality, but they often introduce content deviations and loss of high-frequency details, increasing the risk of uncontrolled degradation.

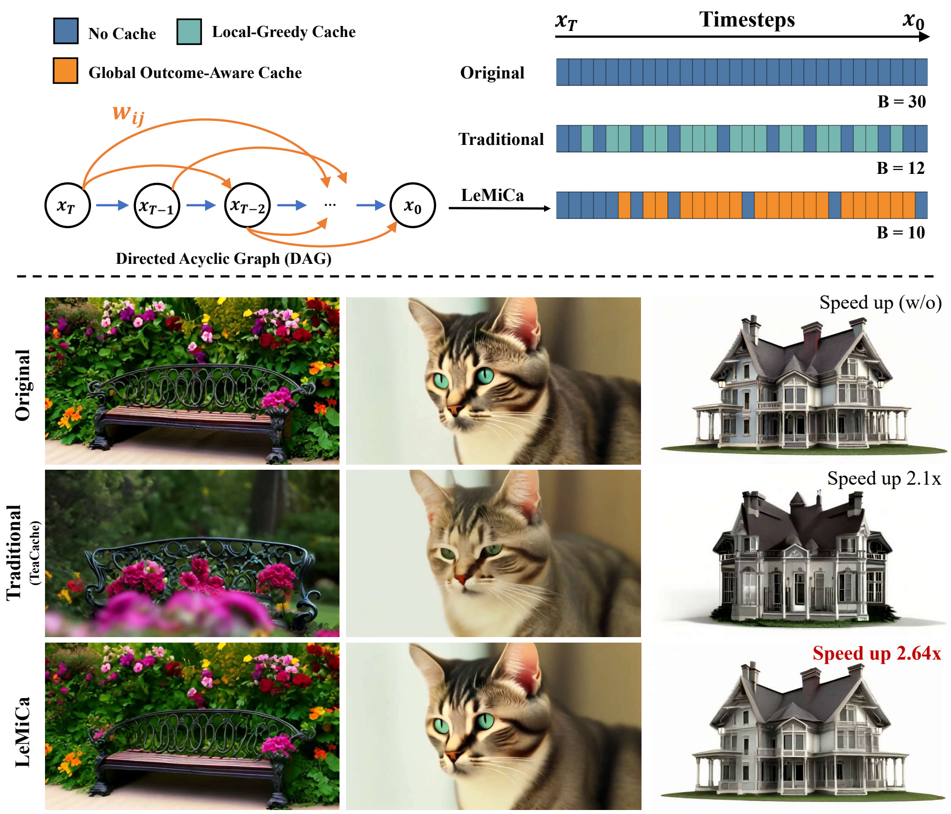


Figure 1: Comparison between our globally controlled cache mechanism (**LeMiCa**) and traditional local greedy cache methods. **Top**: The second row shows the traditional Local-Greedy approach, which uses local error estimation and fixed thresholds for caching decisions. It assumes uniform denoising contributions across time steps and ignores temporal heterogeneity and error propagation. Our method (third row) introduces a *Global Outcome-Aware Cache*, evaluating cache segment impacts through multiple prompts along a fixed sampling path, creating a static directed acyclic graph (DAG). We then use *Lexicographic MiniMax Path Optimization* (LeMiCa) to find the optimal cache path under a fixed inference budget (B, model forward steps). **Bottom**: LeMiCa outperforms traditional methods (e.g., TeaCache) in maintaining structural consistency with faster inference and better control over cache errors and distortions.

Upon further analysis, we identify two key limitations. First, representative methods [44, 20] typically compute local errors between adjacent timesteps and apply fixed thresholds to decide whether to cache. However, the diffusion denoising process exhibits significant *temporal heterogeneity*, with varying noise levels and semantic richness across timesteps. Applying a uniform threshold throughout the process may disrupt semantic alignment and introduce inconsistencies in decision-making, leading to inaccurate caching behavior. Second, these methods mainly focus on minimizing local differences between consecutive steps—what we refer to as *Local-Greedy error*. While this may reduce short-term discrepancies, it overlooks how small errors accumulate over time, potentially resulting in a dual loss in both video quality and content consistency. These issues are evident in TeaCache (a state-of-the-art Local-Greedy method), as shown in Figure 1, particularly with the three frames in the top and second rows, where caching introduces noticeable content deviations and visual quality degradation.

To address these limitations, we propose **Le**xicographic **Mi**nimax **Ca**ching (**LeMiCa**), a static caching framework that is model-agnostic and architecture-independent. Instead of using local greedy strategies, LeMiCa treats cache scheduling as a global path planning problem. This is based on the observation that well-trained diffusion models remain stable along a fixed sampling path.

LeMiCa takes a global view of error by introducing the *Global Outcome-Aware error*, which quantifies the impact of each cache segment on the final output, effectively eliminating temporal heterogeneity and mitigating error propagation. Based on this metric, LeMiCa constructs a *Directed Acyclic Graph* (DAG), where each edge represents a possible cache segment and is weighted by its global impact on output quality. This graph is generated offline using multiple prompts and full sampling trajectories.

We then apply *lexicographic minimax optimization* to identify the path that minimizes worst-case degradation. Among all feasible paths under a fixed budget, the one with the smallest maximum error is selected. If multiple paths have the same maximum error, the next largest error is compared, and so on. This strategy explicitly constrains the worst-case error, effectively preventing global degradation caused by locally unstable cache decisions, and significantly improving content consistency and video quality in accelerated generation.

In summary, the contributions of this paper are:

- We propose **LeMiCa**, a novel, training-free cache scheduling framework that formulates the generation process as a globally optimized DAG traversal task, offering a principled alternative to heuristic and locally greedy approaches.
- We conduct an in-depth analysis of the cache optimization problem and appropriately introduce the Lexicographic Minimax Path Optimization strategy to solve the graph under a fixed cache budget, effectively suppressing error peaks and enhancing global consistency.
- Experiments show that, compared to existing cache techniques, ours achieves dual improvements in inference speed and generation quality across various base models, such as a 2.9X speedup on Latte and an LPIPS of 0.05 on Open-Sora.

2 Related Work

Diffusion Model Acceleration. Diffusion models exhibit strong versatility across domains, but their iterative nature incurs high computational costs, positioning inference acceleration as a central research challenge. Current efforts to accelerate diffusion model sampling focus primarily on reducing sampling steps via schedulers. Denoising Diffusion Implicit Models (DDIM) [38] represents one of the earliest attempts to accelerate sampling by extending the original Denoising Diffusion Probabilistic Model (DDPM) [10] to non-Markovian settings. The Efficient Denoising Model (EDM) [13] introduces a design framework that optimizes specific aspects of the diffusion process. Concurrently, there is growing attention to more efficient and accurate methods for solving stochastic differential equations (SDEs) and ordinary differential equations (ODEs) [40, 12, 21, 3]. Other approaches introduce knowledge distillation [9], training a student model to condense the multi-step outputs of the original diffusion model into fewer steps [22], including Progressive Distillation [30], Consistency Distillation [39, 14, 6, 42, 52], Adversarial Diffusion Distillation [32, 31], and Score Distillation Sampling [47, 46]. Additionally, methods such as quantization [17, 36, 34], pruning [7, 27], optimization [19], and parallelism [50, 15, 5, 4] have been proposed and applied to various diffusion-based generative tasks. However, these methods often require large amounts of computational resources and data for training or intricate engineering designs, which increases the complexity of their application.

Cache in Diffusion Models. Caching mechanisms [35] have recently attracted attention as a retraining-free alternative for accelerating diffusion model inference [44, 25]. The core idea is to reuse model outputs from certain timesteps during sampling to reduce redundant computations [33]. DeepCache [26] accelerates the Unet structure using manually set rules. T-GATE [49] and Δ-DiT [2] apply this idea to DiT-based networks [29], achieving advanced image generation acceleration [54, 16]. With the breakthrough of Sora [28] in video generation, researchers have extended this acceleration concept from image generation to video generation. In this context, PAB [51] observed a U-shaped pattern in attention differences across timesteps in the diffusion process, and based on this, proposed a strategy to cache and broadcast intermediate features at various timestep intervals. FasterCache [23] realized the significant redundancy in conditional generation (CFG) and further enhanced inference speed by utilizing a dynamic feature-based caching mechanism. TeaCache [20] leverages the correlation between timestep embeddings and model outputs, incorporating threshold-based indicators and polynomial fitting to guide caching. Although these methods have improved the efficiency of diffusion-based generation, the core challenge remains in how to accelerate inference while maintaining content consistency and preserving details.

3 Method

3.1 Background: Denoising Diffusion Models

Denoising Diffusion Models achieve generative modeling by simulating the gradual noising and denoising process of data. The core of these models consists of two key stages: **diffusion** and **denoising**. During the forward diffusion process, the model starts from a real sample $x_0 \sim q(x)$ and gradually adds Gaussian noise over T timesteps. The noised sample x_t at timestep t is given by:

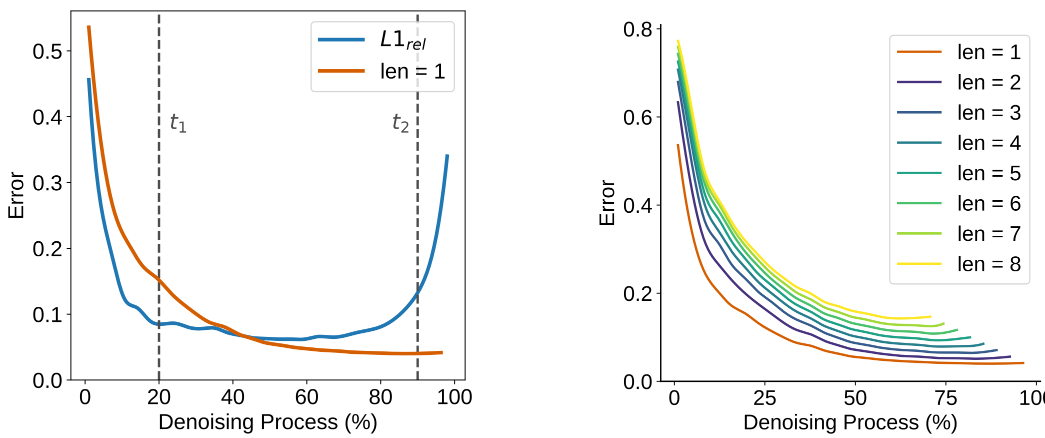
$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, z_t, \quad z_t \sim \mathcal{N}(0, I), \quad t = 1, \ldots, T, \tag{1}$$

where α_t controls the noise strength at each step. As t increases, the samples converge to a standard normal distribution $\mathcal{N}(0, I)$. In the reverse denoising process, the model reconstructs the original data distribution by iteratively denoising through a neural network. The conditional probability for each step is modeled as:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{2}$$

where μ_θ and Σ_θ are learned mean and covariance functions. Due to the multi-step nature of denoising, diffusion models typically incur significant computational overhead during generation.

3.2 Rethinking Cache in Diffusion Sampling



(a) Local vs. Global Cache Error Control

(b) Segment-wise Error Trends

Figure 2: Rethinking cache reuse in denoising diffusion via error estimation. (a) The traditional Local-Greedy ($L1_{rel}$) strategy uses fixed thresholds on local output differences between adjacent timesteps to decide when to cache. This assumes uniform temporal sensitivity, which can be misleading—for instance, caching at t_2 yields lower final error than t_1, despite t_1 seeming smoother locally. This highlights the role of temporal heterogeneity. (b) Our Global Outcome-Aware (*segment-wise error*) strategy estimates final output error when caching outputs over segments of length len, starting from timestep i. The plot shows that early caches cause greater error, supporting an outcome-sensitive, trajectory-aware strategy over fixed local heuristics.

Traditional cache reuse in diffusion sampling typically adopts a *Local-Greedy* strategy (Figure 2a), where caching is based on local differences between adjacent model outputs, often measured by the relative L1 distance [20]:

$$\mathrm{L1}_{\mathrm{rel}}(O, t) = \frac{\|O_t - O_{t+1}\|_1}{\|O_{t+1}\|_1} \tag{3}$$

where O_t is the output at timestep t. High local differences prompt full inference; low differences lead to cache reuse. This step-wise strategy assumes uniform importance across timesteps.

However, diffusion processes are inherently *temporally heterogeneous*—early steps shape global structure, while later steps refine details. Thus, as illustrated in Figure 2a, a seemingly minor change at an early step (e.g., t_1) can have a larger impact on the final output than a larger change at a later step (e.g., t_2). Local metrics fail to account for this asymmetric error propagation, motivating a rethinking of cache strategies.

To address this, we propose a *Global Outcome-Aware* view that considers the *long-term impact* of cache reuse over time. Specifically, we define a *cache segment* (i, j) means full inference is performed at timesteps i and j, while all intermediate steps $t \in (i, j)$ reuse cached outputs:

$$\mathrm{L1}_{\mathrm{glob}}(i \to j) = \frac{1}{N} \left\| x_0^{\mathrm{cache}(i \to j)} - x_0^{\mathrm{original}} \right\|_1 \tag{4}$$

Here, x_0^{original} is the output with no caching, and $x_0^{\mathrm{cache}(i \to j)}$ is the output with segment-level cache. As shown in Figure 2b, the global error depends not just on segment length but also on its *temporal position*—early caches induce amplified downstream errors, while later caches are less disruptive.

These findings reveal two key insights: (1) Global error propagation is *non-uniform* and *time-dependent*, invalidating fixed-threshold heuristics; (2) The *position* of the cache segment matters more than its length. Building on these insights, we formulate cache planning as a graph-based constrained path optimization problem over the sampling trajectory.

3.3 Lexicographic Minimax Path Caching

Based on the rethinking of cache in Sec 3.2, we propose **LeMiCa**, a method that integrates sparse directed graph construction with optimal graph search under peak error control.

Graph Construction. We construct a directed acyclic graph, as shown in Figure 1, where each edge represents a candidate cache segment along the original sampling trajectory. To reduce complexity, we impose a maximum skip length based on the prior that long-range reuse typically leads to large errors, thus avoiding full graph construction. Edge weights are evaluated by replaying cached segments using intermediate states from a full denoising pass. To ensure generality, we build a static graph by averaging edge errors across diverse prompts and noise seeds.

Graph Optimization. Given a directed acyclic graph G with globally error-weighted edges, we frame the caching problem as selecting a path from source s to target t that includes exactly B full computation steps and an arbitrary number of cached segments. This budget-constrained formulation allows flexible reuse while bounding computational cost.

As shown in Figure 2b, early-stage cache errors amplify exponentially during denoising, while late-stage errors remain more localized. This asymmetric error propagation renders traditional shortest-path heuristics— which minimize only additive cost—suboptimal, as they fail to control the dominant sources of degradation.

To better address this imbalance, we adopt a *lexicographic minimax* criterion that explicitly minimizes the highest cache error along the path, followed by the second highest, and so on. Unlike training-based approaches such as ShortDF [3], which directly seek the shortest error path, our formulation—commonly used in control systems for robust worst-case optimization—offers improved stability in error-sensitive settings. Formally, the optimization problem is defined as:

$$\min_{P \in \mathcal{P}_{s \to t}^{(B)}} \text{LexMax}\left(\text{sort\_desc}\left(\{w(e) \mid e \in P_{\text{cache}}\}\right)\right) \tag{5}$$

Here, $\mathcal{P}_{s \to t}^{(B)}$ denotes the set of all paths from s to t with exactly B full steps, and $P_{\text{cache}} \subset P$ are the cached segments within a given path P. The operator LexMax lexicographically minimizes the sorted error vector, ensuring worst-case robustness. The detailed algorithm pseudocode is provided in the Appendix (Section A).

Table 1: Comparison of inference efficiency and visual quality across different models and acceleration strategies on a single GPU.

Method	Efficiency			Visual Quality			
	FLOPs (P)↓	Speedup↑	Latency (s)↓	VBench↑	LPIPS↓	SSIM↑	PSNR↑
Open-Sora 1.2 (51 frames, 480P)							
Original	3.15	1×	26.54	79.22%	—	—	—
Δ-DiT	3.09	1.03×	25.87	78.21%	0.569	0.481	11.91
T-GATE	2.75	1.19×	22.22	77.61%	0.350	0.676	15.50
PAB	2.50	1.43×	18.52	76.95%	0.174	0.822	23.58
TeaCache-slow	2.40	1.50×	17.58	79.20%	0.134	0.837	23.50
TeaCache-fast	1.64	2.10×	12.63	78.24%	0.252	0.743	19.03
LeMiCa-slow	2.30	1.52×	17.43	**79.26%**	**0.050**	**0.923**	**31.32**
LeMiCa-fast	**1.45**	**2.44**×	**10.86**	78.34%	0.187	0.798	21.76
Latte (16 frames, 512×512)							
Original	3.36	1×	11.18	77.40%	—	—	—
Δ-DiT	3.36	1.02×	10.85	52.00%	0.851	0.108	8.65
T-GATE	2.99	1.13×	9.88	75.42%	0.261	0.693	19.55
PAB	2.52	1.36×	8.21	73.13%	0.390	0.642	17.16
TeaCache-slow	1.94	1.65×	6.76	77.40%	0.195	0.775	21.52
TeaCache-fast	1.15	2.60×	4.30	76.09%	0.318	0.674	18.04
LeMiCa-slow	1.88	1.69×	6.60	**77.45%**	**0.091**	**0.865**	**27.65**
LeMiCa-fast	**1.00**	**2.93**×	**3.81**	76.75%	0.273	0.70	19.43
CogVideoX (49 frames, 480P)							
Original	12.45	1×	43.08	77.13%	—	—	—
PAB	9.26	1.43×	32.07	75.95%	0.064	0.916	29.85
TeaCache-slow	6.93	1.70×	25.34	76.79%	0.053	0.928	31.07
TeaCache-fast	4.53	2.45×	17.58	76.06%	0.176	0.804	22.95
LeMiCa-slow	6.91	1.72×	25.02	**76.89%**	**0.023**	**0.958**	**35.93**
LeMiCa-fast	**4.26**	**2.61**×	**16.48**	76.20%	0.132	0.846	25.59

4 Experiments

4.1 Experimental Setup

Metrics For fair comparison, we follow prior works and report both efficiency and visual quality metrics. Efficiency is measured by FLOPs and latency. Visual quality is evaluated using VBench [11] (human preference), LPIPS [48] (perceptual similarity), SSIM [43] (structural consistency), and PSNR (pixel-level accuracy).

Baselines and Compared Methods We evaluate our method on representative diffusion-based video models: Open-Sora [53], Latte [24], and CogVideoX [45]. Baselines include Δ-DiT [2], T-GATE [49], PAB [51], and TeaCache [20]. Among them, T-GATE and Δ-DiT are designed for images, while PAB and TeaCache target video. Accordingly, we compare against PAB and TeaCache on CogVideoX, and against all four baselines on Open-Sora and Latte.

Implementation Details Experiments are conducted on NVIDIA H100 GPUs using PyTorch. To construct the DAG for Global Outcome-Aware error modeling, we sample 70 prompts (10 per attribute) from T2V-CompBench [41], following standard practice [41, 20]. The DAG construction and forward inference use distinct datasets to ensure fair and robust evaluation. Sampling is repeated 10 times with different seeds, and results are averaged to reduce bias.

4.2 Comparison with State-of-the-Art Methods

Quantitative Comparison Table 1 compares LeMiCa with baselines across four metrics: VBench, LPIPS, SSIM, and PSNR. LeMiCa includes two variants: LeMiCa-slow (fidelity-focused) and

LeMiCa-fast (speed-focused). It consistently outperforms training-free acceleration baselines across models, schedulers, resolutions, and video lengths. LeMiCa-slow achieves the best reconstruction quality, reducing LPIPS from 0.134 to 0.05 on Open-Sora and from 0.195 to 0.091 on Latte—over 2× improvement vs. TeaCache-slow. LeMiCa-fast improves inference speed from 2.60× to 2.93× on Latte compared to TeaCache-fast, while preserving visual quality. Unlike prior methods relying on online greedy strategies, LeMiCa precomputes its caching policy, eliminating runtime overhead. Overall, LeMiCa provides efficient video generation with minimal perceptual quality degradation.

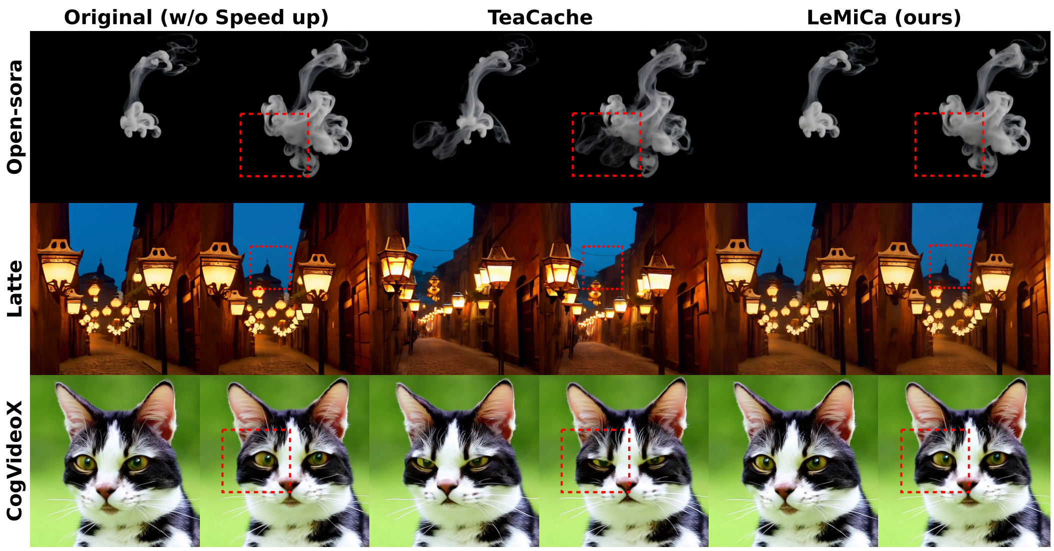


Figure 3: Visual comparison under the fidelity-focused setting (LeMiCa-slow vs. TeaCache-slow) across different models. Differences are highlighted in red boxes.

Visualization We compare video acceleration methods from both quality and speed perspectives. As shown in Fig. 3, under the fidelity-focused setting, LeMiCa excels in preserving content consistency and fine details, as highlighted in red boxes. This demonstrates its ability to maintain high-quality visuals even when prioritizing fidelity. In contrast, Fig. 4 illustrates that under the speed-focused se-

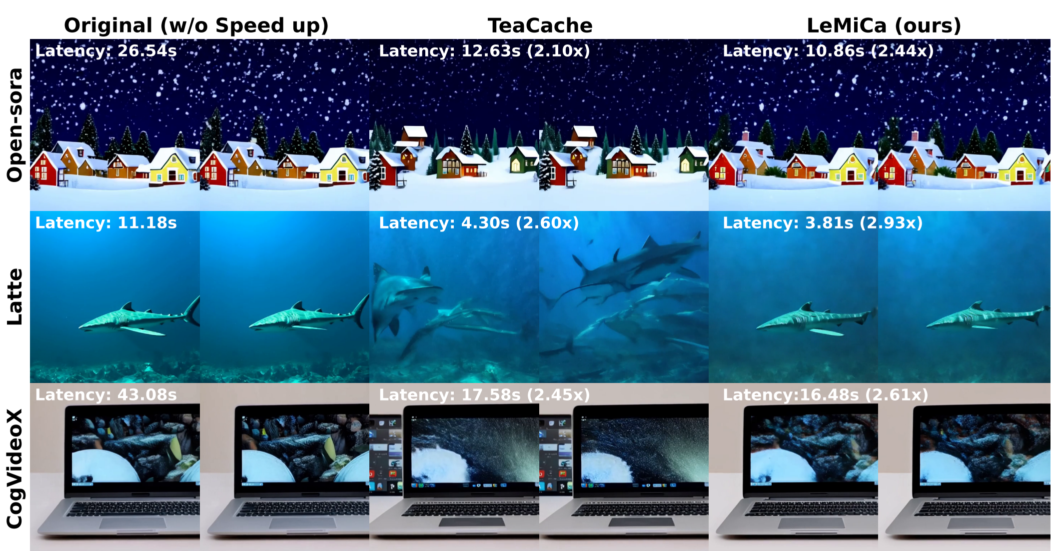


Figure 4: Visual comparison under a speed-focused setting (LeMiCa-fast vs. TeaCache-fast). LeMiCa-fast better preserves content consistency and video quality under a high speedup (>2×).

tting, LeMiCa-fast significantly outperforms TeaCache-fast, achieving superior acceleration rates while still maintaining competitive performance. These results highlight LeMiCa's ability to balance quality and speed across different configurations. Additional qualitative examples can be found in the Appendix (Section E).

4.3 Ablation Studies

Acceleration vs. Performance trade-off Figure 5 presents the quality-latency trade-off between our proposed LeMiCa and TeaCache. To ensure comparable computational budgets, LeMiCa is configured with 19, 12, 9, and 7 inference steps (i.e., inference budget B), corresponding to TeaCache thresholds of 0.1, 0.2, 0.3, and 0.5, respectively. Across all latency regimes, LeMiCa consistently achieves a superior quality-efficiency balance, outperforming TeaCache on all reference-based metrics. Importantly, under extreme acceleration (latencies below 8 seconds), LeMiCa maintains robust and high-quality performance.

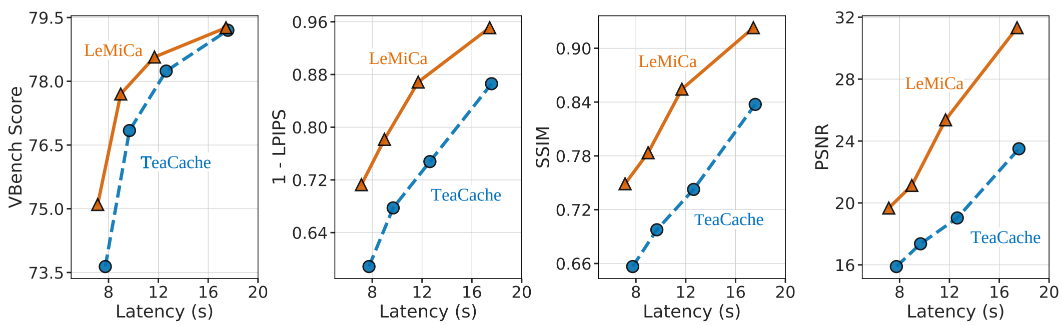


Figure 5: Quality-latency trade-off comparison between LeMiCa and TeaCache.

Sample Requirements for Graph Construction To investigate how many samples LeMiCa requires to offline construct the DAG, we randomly select $n \in \{1, 5, 10, 20\}$ from the original 350 samples (70 prompts × 5 seeds), and compute the optimal caching path under the lexicographic minimax criterion. Each setting is repeated 20 times to reduce randomness. Importantly, distinct datasets are used for DAG construction and forward inference to guarantee fairness and robustness in evaluation. Video quality is then evaluated on 50 selected VBench prompts, with average results reported. Table 2 shows that LeMiCa achieves strong performance with a single sample (e.g., PSNR 24.51), rapidly approaching the upper bound with 10 samples and essentially saturating at 20 samples across all metrics. This demonstrates LeMiCa's ability to construct high-quality cache paths with minimal samples and underscores the robustness of its static caching strategy across varying prompts and seeds.

Table 2: Impact of sample size on cache path graph quality.

Number of Samples	VBench↑	LPIPS↓	SSIM↑	PSNR↑
1	78.58	0.164	0.838	24.51
5	78.70	0.161	0.843	24.57
10	78.95	0.158	0.844	24.56
20	79.16	0.152	0.843	24.60
350	79.27	0.143	0.851	24.67

Table 3: Impact of different path strategies on video reconstruction quality.

Path Strategy	VBench↑	LPIPS↓	SSIM↑	PSNR↑
Original	79.24	-	-	-
Shortest Path	76.04	0.203	0.809	22.90
MiniMax Path	79.27	0.143	0.851	24.67

Trajectory Robustness Since the cache mechanism is inherently tied to the original denoising trajectory, it is essential to assess whether a training-free cache method remains effective when the trajectory changes. To this end, we vary the trajectory scale parameter in the sampling schedule from its default value of 1.0 to several alternative values (0.5, 0.75, 1.25, 1.5), introducing different diffusion paths during inference. As shown in Figure 6, the left panel illustrates the effect of trajectory scaling on the denoising paths, while the right panel demonstrates that LeMiCa consistently outperforms the current state-of-the-art method, TeaCache, across all trajectories in terms of LPIPS. These results confirm that our method remains effective even under varying denoising paths.

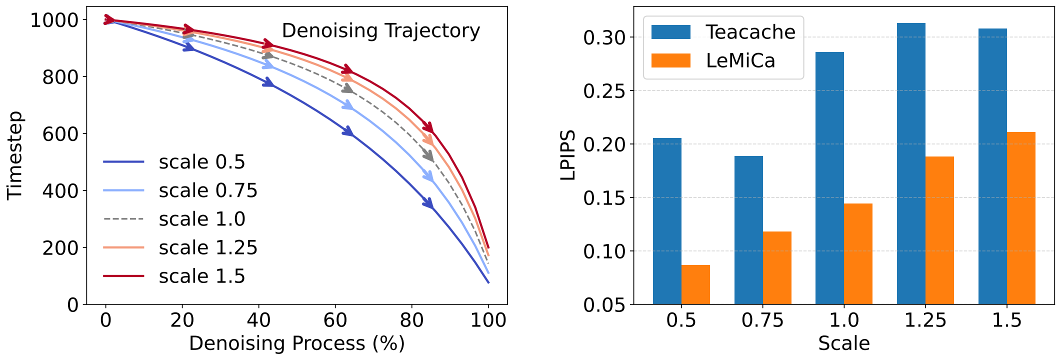


Figure 6: Performance comparison between LeMiCa and TeaCache across different denoising trajectories. Left: Denoising step trajectories under different scale settings. Right: LPIPS performance across various denoising trajectories.

Shortest Path vs. Lexicographic MiniMax Path We compare the performance of the *Shortest Path* strategy and the *Lexicographic MiniMax Path* strategy in video reconstruction tasks. As shown in Table 3, the *MiniMax Path* strategy consistently outperforms the baseline *Shortest Path* strategy in both VBench scores and reconstruction metrics. This observation is consistent with our analysis: the edge errors cached during the sampling process are not independent and thus cannot be simply accumulated linearly.

Performance at different resolutions and lengths Our method incorporates Dynamic Sequence Parallelism (DSP) [51]to support high-resolution long-video generation across multiple GPUs. To assess its sampling acceleration performance across varying video sizes, we conducted tests on videos with different lengths and resolutions. As shown in Figure 7, our method maintains stable acceleration even as video resolution and frame count increase, highlighting its potential for handling longer and higher-resolution videos.

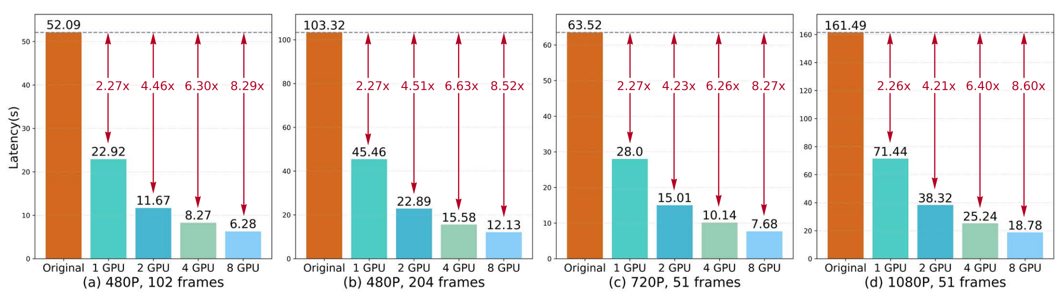


Figure 7: LeMiCa inference efficiency under various video durations and resolutions.

5 Conclusion

We propose **LeMiCa**, a general and efficient caching framework for accelerating diffusion-based video generation. Unlike locally greedy strategies, LeMiCa formulates cache scheduling as a global path optimization problem using lexicographic minimax over a static DAG, effectively constraining worst-case degradation. With the introduction of the Global Outcome-Aware error, our method captures the long-term impact of caching decisions, mitigating temporal heterogeneity and error accumulation. Extensive experiments demonstrate that LeMiCa consistently improves both efficiency and visual quality across diverse diffusion models. More broadly, LeMiCa offers a new perspective on structured caching in generative modeling, which may inspire future research in other domains such as 3D, multi-view, or multi-modal generation where controllable acceleration remains an open challenge.

References

[1] Zechen Bai, Hai Ci, and Mike Zheng Shou. Impossible videos. *arXiv preprint arXiv:2503.14378*, 2025.

[2] 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.

[3] Ping Chen, Xingpeng Zhang, Zhaoxiang Liu, Huan Hu, Xiang Liu, Kai Wang, Min Wang, Yanlin Qian, and Shiguo Lian. Optimizing for the shortest path in denoising diffusion model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[4] Zigeng Chen, Xinyin Ma, Gongfan Fang, Zhenxiong Tan, and Xinchao Wang. Asyncdiff: Parallelizing diffusion models by asynchronous denoising. *arXiv preprint arXiv:2406.06911*, 2024.

[5] Jiarui Fang, Jinzhe Pan, Xibo Sun, Aoyu Li, and Jiannan Wang. xdit: an inference engine for diffusion transformers (dits) with massive parallelism. *arXiv preprint arXiv:2411.01738*, 2024.

[6] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] 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.

[12] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

[13] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[14] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

[15] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7193, 2024.

[16] Senmao Li, Taihang Hu, Fahad Shahbaz Khan, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster diffusion: Rethinking the role of unet encoder in diffusion models. *arXiv e-prints*, pages arXiv–2312, 2023.

[17] Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. *Advances in neural information processing systems*, 36:76680–76691, 2023.

[18] Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. In *International Conference on Machine Learning*, pages 21915–21936. PMLR, 2023.

[20] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *CoRR*, abs/2411.19108, 2024.

[21] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[22] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[23] Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024.

[24] Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

[25] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37:133282–133304, 2024.

[26] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. *arXiv preprint arXiv:2312.00858*, 2023.

[27] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[28] OpenAI. Sora, 2024. https://openai.com/index/sora/.

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[30] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[31] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[32] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[33] Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

[34] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1972–1981, 2023.

[35] Alan Jay Smith. Cache memories. *ACM Computing Surveys (CSUR)*, 14(3):473–530, 1982.

[36] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in neural information processing systems*, 36:48686–48698, 2023.

[37] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[39] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

[40] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[41] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. *arXiv preprint arXiv:2407.14505*, 2024.

[42] Cunzheng Wang, Ziyuan Guo, Yuxuan Duan, Huaxia Li, Nemo Chen, Xu Tang, and Yao Hu. Target-driven distillation: Consistency distillation with target timestep selection and decoupled guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7619–7627, 2025.

[43] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002.

[44] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. *arXiv preprint arXiv:2312.03209*, 2023.

[45] 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.

[46] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[47] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[49] Wentian Zhang, Haozhe Liu, Jinheng Xie, Francesco Faccio, Mike Zheng Shou, and Jürgen Schmidhuber. Cross-attention makes inference cumbersome in text-to-image diffusion models. *arXiv preprint arXiv:2404.02747*, 2024.

[50] Xuanlei Zhao, Shenggan Cheng, Chang Chen, Zangwei Zheng, Ziming Liu, Zheming Yang, and Yang You. Dsp: Dynamic sequence parallelism for multi-dimensional transformers. *arXiv preprint arXiv:2403.10266*, 2024.

[51] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024.

[52] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024.

[53] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. https://github.com/hpcaitech/Open-Sora.

[54] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *CoRR*, abs/2410.05317, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not involve theoretical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We would release the code and the code is submitted in the supplemental material

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We don't report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4.1 Implementation Details

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research conform to the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss it in Appendix

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We have no new models/datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in our paper is cited or marked in the code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Would release the code

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Doesn't cover large language models (LLMs)

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

LeMiCa: Lexicographic Minimax Path Caching for Efficient Diffusion-Based Video Generation

Appendix

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

Input: Directed acyclic graph $G = (V, E)$, start node s, end node t, step limit B
Output: Lexicographic Minimax Path P^*
Initialization:
  $dp[v][k]$: maximum edge weight on any k-step path to v
  $paths[v][k], edges[v][k]$: corresponding node and edge sequences
  $dp[s][0] \leftarrow 0$, $paths[s][0] \leftarrow [[s]]$, $edges[s][0] \leftarrow [[\,]]$
Main Loop:
for $k = 0$ to $B - 1$ **do**
  for each node v with $dp[v][k] < \infty$ **do**
    for each neighbor u of v **do**
      $w \leftarrow$ weight of edge (v, u)
      $m \leftarrow \max(dp[v][k], w)$
      if $m < dp[u][k+1]$ **then**
        $dp[u][k+1] \leftarrow m$
        Update $paths[u][k+1]$, $edges[u][k+1]$ from v
      else if $m = dp[u][k+1]$ **then**
        Append new paths and edges from v to $paths[u][k+1]$, $edges[u][k+1]$
      end if
    end for
  end for
end for
Final Selection:
$P^* \leftarrow \min\big(\text{zip}(paths[t][B], edges[t][B]), \text{key} = \lambda(p, e) : \text{sorted}(e, \text{reverse=True})\big)$

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 [53], Latte [24], and CogVideoX [45]. 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 OOD Generalization Analysis

We analyze the out-of-distribution (OOD) generalization ability of LeMiCa through two evaluation setups: VBench and IP-VBench.

OOD Evaluation on VBench. LeMiCa is evaluated on VBench [11], which follows a distribution distinct from T2V-CompBench [41] used for DAG construction. We quantify the distributional shift by computing prompt-level distances using text embeddings and PCA (see Table 5).

Table 5: Distributional distance analysis between VBench and T2V-CompBench.

Attribute	**Description**	**VBench**
Distance	VBench vs. T2V-CompBench	0.61
Radius	1 Std. Deviation of T2V-CompBench	0.39
Distance / Radius	Ratio of distance to radius	1.58
OOD Status*	Is it OOD?	✓

Despite this evident OOD setting, LeMiCa consistently maintains strong acceleration and visual quality (Table 1), demonstrating robustness to unseen prompt distributions.

OOD Evaluation on IP-VBench. We further test LeMiCa on IP-VBench [1], which contains intentionally unrealistic and semantically diverse prompts across four domains: *Physical*, *Biological*, *Social*, and *Geographical*. These prompts differ significantly from training data, with Distance/Radius nearly doubling compared to T2V-CompBench (see Table 6).

Across all domains, LeMiCa substantially outperforms TeaCache in LPIPS, SSIM, and PSNR, underscoring its strong generalization and robustness under severe OOD conditions.

Table 6: Quantitative OOD performance on IP-VBench across four semantic domains.

Method	Domain	LPIPS (↓)	SSIM (↑)	PSNR (↑)	Distance/Radius	OOD Status*
TeaCache	Physical	0.093	0.911	26.7	2.09	✓
	Biological	0.171	0.839	24.0	1.90	✓
	Social	0.144	0.842	24.9	1.93	✓
	Geographical	0.072	0.914	29.8	2.13	✓
	Overall	**0.120**	**0.877**	**26.4**	**2.01**	✓
LeMiCa	Physical	0.039	0.954	34.6	2.09	✓
	Biological	0.054	0.905	31.4	1.90	✓
	Social	0.040	0.946	33.1	1.93	✓
	Geographical	0.038	0.945	34.9	2.13	✓
	Overall	**0.042**	**0.938**	**33.5**	**2.01**	✓

D Offline Cost

The graph construction in LeMiCa is an entirely offline, three-stage process. First, Edge Weight Estimation estimates reconstruction errors by running full-generation passes on approximately 20 sampled prompts; this task is fully parallelizable (leveraging 8 GPUs in our experiments) and only needs to be run once per model configuration. The subsequent stages, Graph Construction (fusing jump edges into a sparse DAG) and Path Optimization (employing a lexicographic minimax search to find acceleration paths), are both highly efficient, each completing in under 1 second. As detailed in Table 7, these offline procedures incur negligible overhead, yet this low-cost offline computation yields up to **2.44**× acceleration during inference generation.

Table 7: Offline cost analysis of LeMiCa on OpenSora.

Stage	Description	Time Cost (Ref)	Affects Inference
Edge Weight Estimation	Full-generation error estimation	∼3.18 min / prompt	No
Graph Construction	Build sparse DAG	<1 sec	No
Path Optimization	Minimax search for jump paths	<1 sec	No
Inference Acceleration	Execute jump paths with caching	Up to 2.44× faster	Yes

E More Visual Results

We present additional visual comparisons across three foundational models: Open-Sora [53], Latte [24], and CogVideoX [45]. Results are grouped into two settings: fidelity-focused and speed-focused.

E.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.

E.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**.

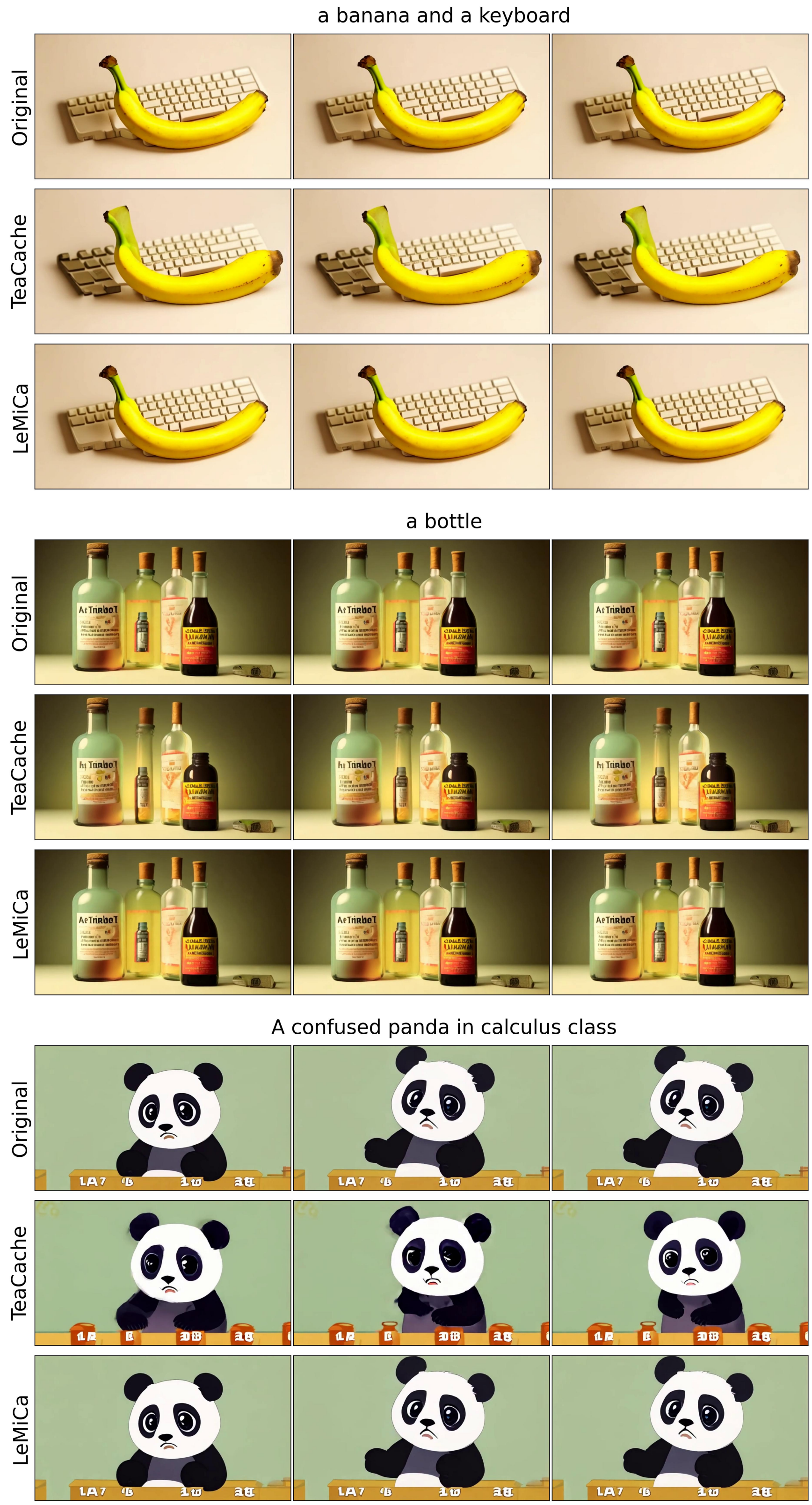


Figure 8: More visual results on Open-Sora (Part I).

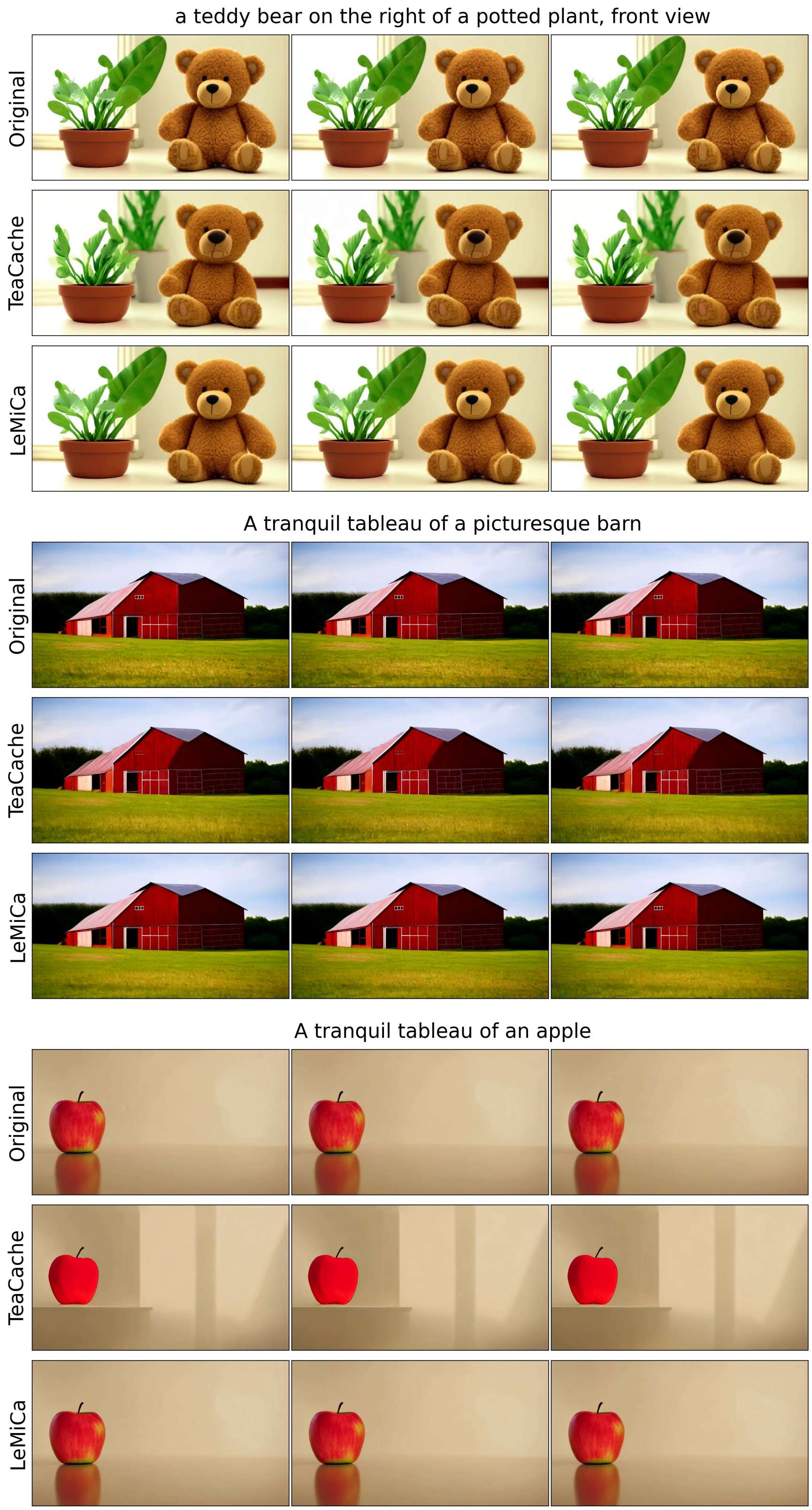


Figure 9: More visual results on Open-Sora (Part II).

Figure 10: More visual results on CogVideoX.

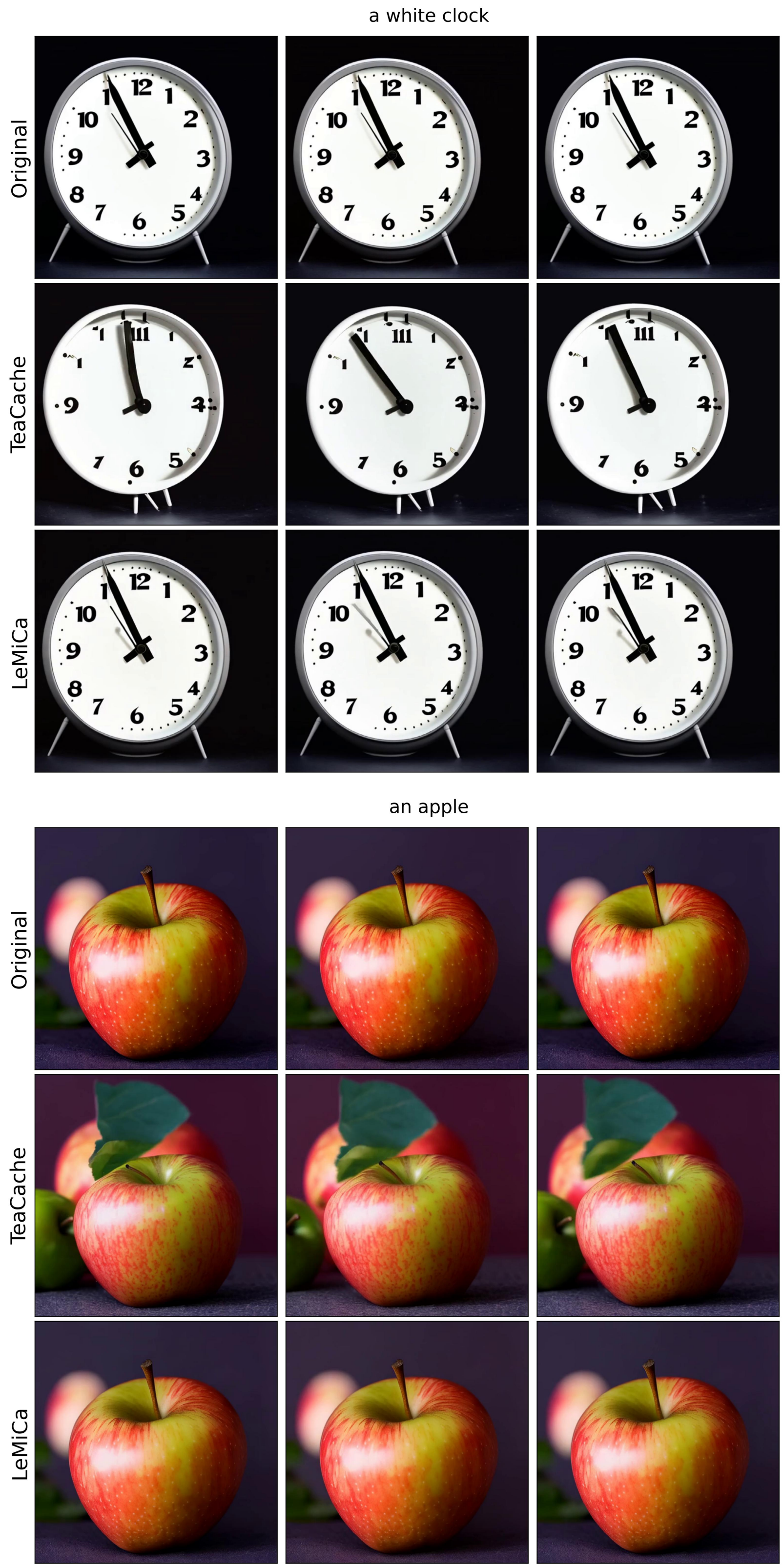


Figure 11: More visual results on Latte (Part I).

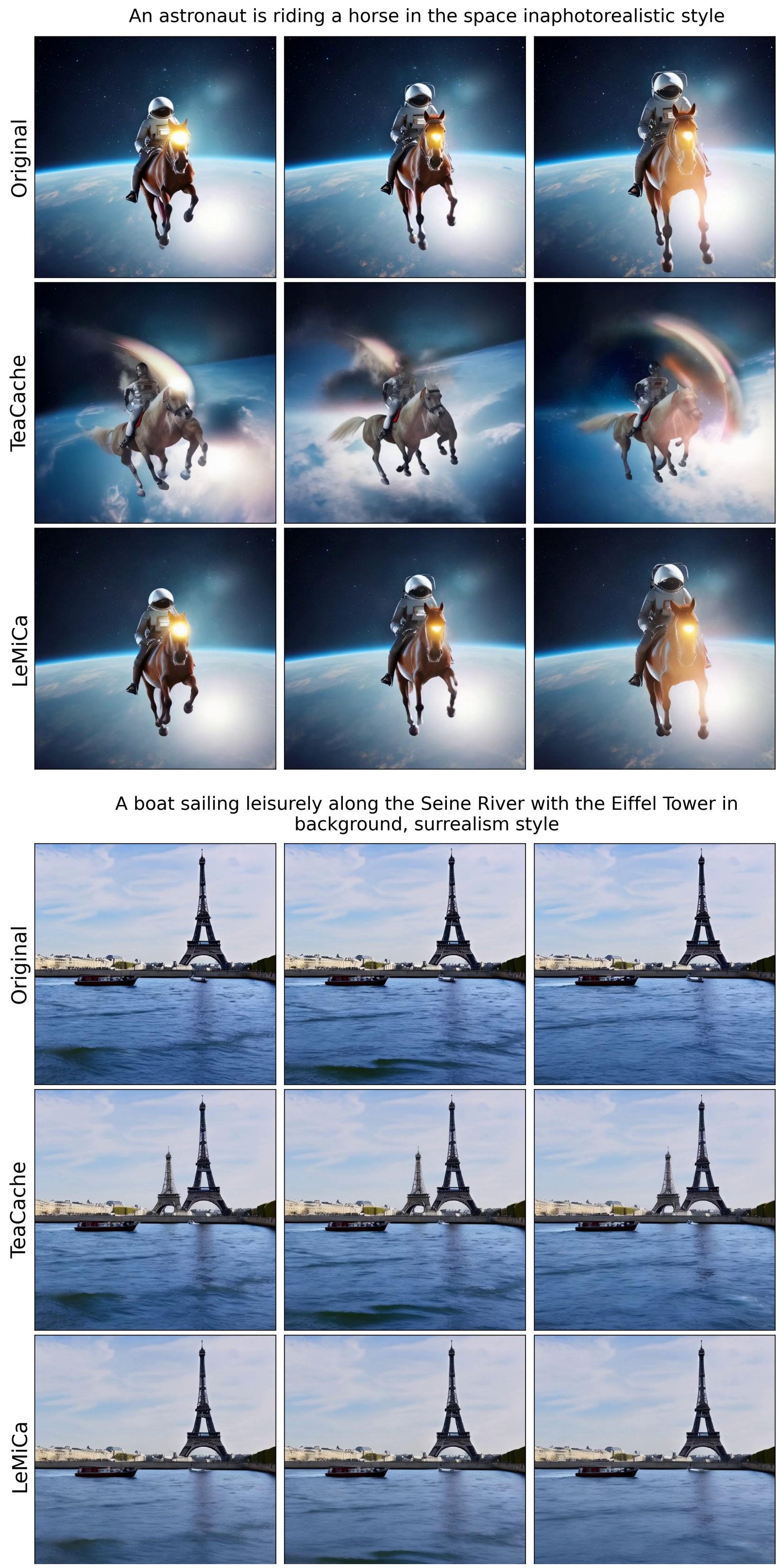


Figure 12: More visual results on Latte (Part II).

F 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.

G 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.