Focused Forcing: Content-Aware Per-Frame KV Selection for Efficient Autoregressive Video Diffusion

Peiliang Cai[1] **Evelyn Zhang**[1] **Jiacheng Liu**[1,2] **Hao Lin**[3] **Ruiqi Zhang**[4] **Weile Mo**[1]
Yue Ma[5] **Shikang Zheng**[1,6] **Jiehang Huang**[1] **Dongrui Liu**[7] **Linfeng Zhang**[1†]

[1]SJTU [2]SDU [3]HUST [4]UTokyo [5]HKUST [6]SCUT [7]Shanghai AI Lab

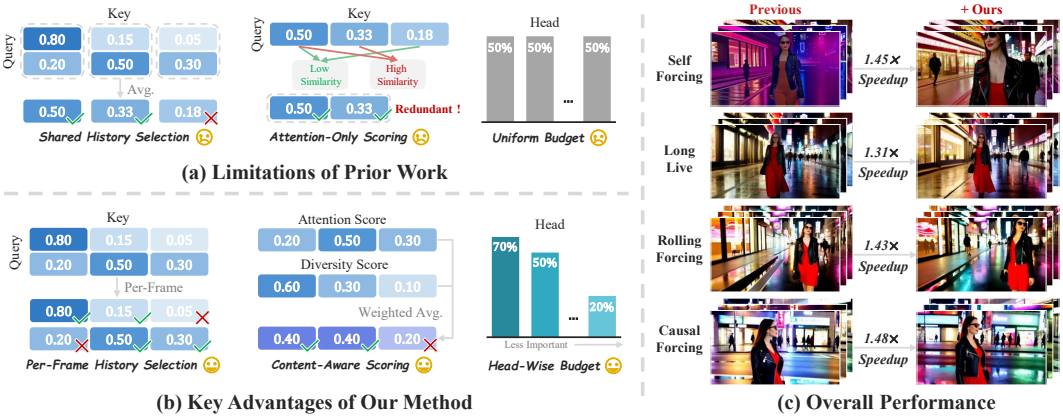


Figure 1: **Overview of Focused Forcing.** (a) Existing methods use shared history selection, attention-only scoring, and coarse head budgeting, illustrated here with uniform budgets. (b) Focused Forcing uses per-frame history selection, content-aware scoring, and head-adaptive budgets. (c) Focused Forcing achieves up to 1.48× speedup across multiple paradigms while preserving quality.

Abstract

Recent advances in autoregressive video diffusion have enabled sequential and streaming video generation. However, long-horizon generation requires increasingly large KV caches, making efficient compression without sacrificing quality challenging. Existing methods mostly select historical frames based on attention scores, but their context decisions remain coarse. When multiple frames are generated in the same chunk, these methods often apply a shared history selection to the whole chunk, score historical frames solely by attention, and assign head-wise budgets either uniformly or by attention-pattern heuristics rather than explicit head-importance estimation. We show that frames within the same generated chunk can depend on distinct historical frames, that the same historical frame can receive different attention scores as its relative temporal distance to the current frames changes, and that masking different heads induces unequal generation degradation. Motivated by these findings, we propose **Focused Forcing**, a training-free KV selection method that focuses cached history along both generated-frame and head dimensions. For each generated frame, Focused Forcing preserves the most relevant and distinctive historical frames by combining attention scores with diversity scores of historical frames, while assigning larger budgets to heads with higher estimated importance. Across multiple autoregressive generation paradigms, Focused Forcing achieves up to **1.48**× end-to-end acceleration without training, while **improving visual quality and text alignment**. *Our code is available at* https://github.com/caipeiliang2004/Focused-Forcing

†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Recent advances in video diffusion models [4, 17, 30, 41, 47] have greatly improved visual fidelity and motion coherence, making long-form, interactive, and streaming video generation increasingly practical. Unlike bidirectional paradigms that synthesize an entire clip jointly, autoregressive video diffusion [6, 51] generates video sequentially, often in chunks, and naturally supports online rollout with KV caching [21, 26, 59]. However, long-horizon rollout requires maintaining an ever-growing cache of previously generated frames, making later attention operations increasingly expensive in memory and computation.

To improve efficiency, existing methods typically restrict or compress the accessible history with sliding-window or rolling-KV designs [8, 9, 21, 59]. Some preserve temporal continuity through structured cache rules such as attention sinks or cache re-caching [46, 48], while others select or prune cached frames by attention scores [48]. Recent work also explores head-specific cache strategies [16]. Despite these advances, as illustrated in Fig. 1(a), existing methods still make coarse-grained context decisions in three aspects: **1) Shared Chunk-Level Selection**, where all frames in a generated chunk share the same history selection; **2) Attention-Only Ranking**, ranking historical frames mainly by attention scores; and **3) Coarse Head-Wise Budgeting**, allocating head-wise budgets uniformly or heuristically without explicit head-importance estimation. These designs reduce cache overhead, but do not fully match the fine-grained context requirements of autoregressive video generation.

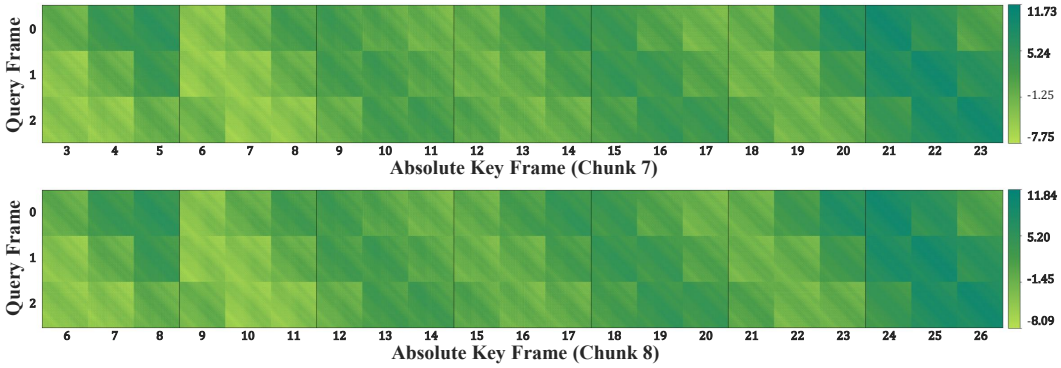


Figure 2: **Attention is generated-frame-dependent and relative-temporal-distance-sensitive.** Rows denote query frames in the current generated chunk, and columns denote historical key frames. Attention varies across query frames and changes with relative temporal distance.

Attention patterns in autoregressive video diffusion suggest that history selection should be performed at a finer granularity. As shown in Fig. 2, frames within the same generated chunk can attend to different historical frames, indicating that context demand is generated-frame-dependent rather than shared by the whole chunk. Moreover, the same historical frame can receive different attention scores as its relative temporal distance to the current frames changes, suggesting that attention reflects not only content relevance but also temporal position. Therefore, selecting history with a single aggregated attention score may discard frames that are important for specific generated frames, while attention-only ranking can be unstable as relative temporal distance affects attention scores.

Beyond selecting historical frames for each generated frame, fine-grained cache compression also requires allocating appropriate history budgets across attention heads. Existing head-specific strategies mainly rely on aggregated attention scores or heuristic attention patterns, but they do not explicitly estimate head importance for generation quality. As a result, they lack a direct criterion for head-wise KV budget allocation. This motivates assigning larger historical budgets to more important heads.

Taken together, these observations motivate fine-grained KV compression across two dimensions: per-frame history selection and per-head budget allocation. Instead of pruning historical frames with a single aggregated attention score, KV compression should determine ***which*** historical frames each generated frame should access and ***how much*** history each attention head should retain. Meanwhile, the selection criterion should go beyond attention alone, since attention scores can be affected by relative temporal distance.

Motivated by this formulation, we propose **Focused Forcing**, a training-free KV compression method for autoregressive video generation. As shown in Fig. 1(b), Focused Forcing addresses fine-grained

context allocation with three key designs: **1) Generated-Frame-Wise History Selection.** Focused Forcing selects historical frames separately for each generated frame, rather than using one shared selection for the whole generated chunk. **2) Content-Aware Scoring.** Focused Forcing combines attention relevance with diversity scores of historical frames, preserving distinctive historical frames when attention scores are affected by relative temporal position. **3) Head-Wise Budget Allocation.** Focused Forcing estimates head importance by masking each attention head during denoising rollout and measuring the resulting degradation with a distribution-matching loss, then assigns larger KV budgets to more influential heads.

Experiments across multiple autoregressive generation paradigms show that Focused Forcing consistently reduces cache overhead without additional training. As shown in Fig. 1(c), it achieves up to $1.48\times$ end-to-end acceleration while also improving visual quality and text alignment, demonstrating that fine-grained KV compression enhances both efficiency and quality in long-horizon video generation.

Our contributions are summarized as follows:

- We observe that attention-based history selection in autoregressive video generation is generated-frame-dependent and relative-position-sensitive, and that different attention heads have unequal impact on generation quality.
- We propose **Focused Forcing**, a training-free KV compression method that performs fine-grained context allocation through generated-frame-wise history selection, content-aware scoring with attention and diversity scores of historical frames, and head-wise differentiated KV budgets.
- Experiments across multiple autoregressive generation paradigms show that Focused Forcing achieves up to **1.48**$\times$ end-to-end acceleration without training, while **improving visual quality and text alignment**.

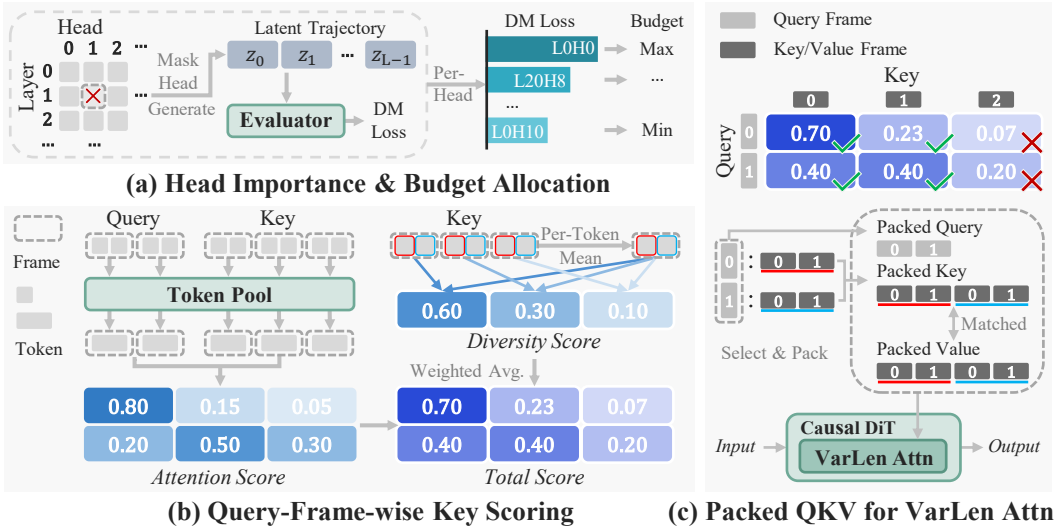
2 Related Work

Video Diffusion Models. Diffusion models [3, 19, 31–33] have set a strong baseline for image and video generation. Benefiting from scaled data and compute, recent video generators achieve high visual fidelity and temporal consistency, e.g., Wan [41], HunyuanVideo 1.5 [42], LTX Video [18], and Seedance2 [36], most of which rely on bidirectional spatiotemporal attention that ensures global consistency but is costly for long-form or interactive use. To support long-horizon and online generation, causal autoregressive video diffusion synthesizes future frames sequentially with historical context maintained through KV caching, via causal conversion from bidirectional models [51], rollout-aware training [21], streaming rolling-window generation [26, 46, 55], and AR diffusion distillation [59], and has been scaled in industrial world-modeling and game-content systems [39, 40, 52].

Efficient Video Generation. Efficient video diffusion typically reduces denoising steps, per-step cost, or redundant computation along the trajectory. Better samplers accelerate inference via deterministic non-Markovian sampling [38], high-order ODE solvers [27], and unified predictor-corrector schemes [57]. Distillation and consistency training compress trajectories into few- or one-step generation [14, 15, 28, 35]. Per-step cost is cut by memory-efficient or low-bit attention, structured attention for real-time video [1], and multi-GPU parallelism [10, 11, 13, 23, 53, 54]. Training-free diffusion caching exploits feature redundancy across adjacent denoising steps, evolving from reusing slow features in U-Net/DiT [29, 37] to finer-grained or video-aware reuse [24, 60], and to predictive caching that extrapolates future activations [5, 25].

KV Cache Compression for Autoregressive Generation. In autoregressive language models, KV cache compression prunes or allocates history via attention salience, recency, sink tokens, or head-wise specialization [20, 43, 44, 56]. Video counterparts face growing KV caches and spatiotemporal redundancy. Dummy Forcing [16] studies head-level dependence on long-range context and applies head-wise KV compression. Relax Forcing [58] organizes temporal memory into Sink, History, and Tail, and dynamically selects informative historical frames. These methods improve cache efficiency and rollout stability, but mainly make head-level or coarse temporal context decisions. Our contribution lies in the joint formulation of query-frame-wise history selection and quality-impact-based head-wise budget allocation, rather than introducing either mechanism independently.

3 Method



(a) Head Importance & Budget Allocation

(b) Query-Frame-wise Key Scoring

(c) Packed QKV for VarLen Attn

Figure 3: **Overview of Focused Forcing.** (a) We estimate head importance by masking each head and measuring the DM loss, then allocate larger KV budgets to more important heads. (b) For each query frame and head, we score historical frames by combining attention scores with their diversity scores. (c) The selected QKV rows are packed and computed by variable-length FlashAttention.

3.1 Preliminary

Autoregressive Video Diffusion. Autoregressive video diffusion generates videos causally, where each frame or chunk is conditioned on previously generated context. Given a video decomposed into autoregressive units $x^{1:N} = (x^1, x^2, \ldots, x^N)$, the joint distribution is factorized as $p(x^{1:N}) = \prod_{i=1}^{N} p(x^i \mid x^{<i})$, where each x^i can denote either a frame or a chunk.

In chunk-wise generation, tokens are grouped into F_q query frames from the current chunk and F_h historical frames from the KV cache:

$$Q = [Q^{(1)}, \ldots, Q^{(F_q)}], \qquad K = [K^{(1)}, \ldots, K^{(F_h)}], \tag{1}$$

where $Q^{(f)}$ and $K^{(t)}$ denote tokens from the f-th query frame and the t-th historical frame.

Distribution Matching Loss. Following distribution matching distillation [49, 50], the distribution matching gradient is given by the difference between fake and real score functions. We use its surrogate loss form:

$$\mathcal{L}_{\mathrm{DM}}(x) = \frac{1}{2} \left\| x - \mathrm{sg}(x - g_{\mathrm{DM}}(x_t, t)) \right\|_2^2, \quad g_{\mathrm{DM}}(x_t, t) = w_t \alpha_t \left(s_{\mathrm{fake}}(x_t, t) - s_{\mathrm{real}}(x_t, t) \right), \tag{2}$$

where x_t is obtained by adding noise to x, s_{fake} and s_{real} denote the score functions of the generated and target distributions, w_t is a timestep-dependent weight, α_t denotes the signal scaling coefficient at timestep t, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. A larger $\mathcal{L}_{\mathrm{DM}}$ indicates stronger distributional deviation and is used to measure generation degradation.

Variable-Length FlashAttention. For samples with different sequence lengths $\{L_b\}_{b=1}^{B}$, where $Q_b, K_b, V_b \in \mathbb{R}^{1 \times L_b \times H \times D}$, variable-length FlashAttention [10, 11] concatenates them along the token dimension:

$$Q_{\mathrm{pack}} = [Q_1; \ldots; Q_B], \quad K_{\mathrm{pack}} = [K_1; \ldots; K_B], \quad V_{\mathrm{pack}} = [V_1; \ldots; V_B]. \tag{3}$$

Let $c_0^q = c_0^k = 0$, $c_b^q = \sum_{i=1}^{b} L_i^q$, and $c_b^k = \sum_{i=1}^{b} L_i^k$ denote the cumulative boundaries for query and key-value sequences. The attention output is written back to the packed query layout:

$$O_{\mathrm{pack}}[c_{b-1}^q : c_b^q] = \mathrm{softmax}\left(\frac{Q_{\mathrm{pack}}[c_{b-1}^q : c_b^q] \, K_{\mathrm{pack}}[c_{b-1}^k : c_b^k]^\top}{\sqrt{D}} \right) V_{\mathrm{pack}}[c_{b-1}^k : c_b^k], \tag{4}$$

where O_{pack} denotes the packed attention output, and D is the attention head dimension.

3.2 Head-Wise Budget Allocation

Different attention heads can have unequal impact on generation quality, so assigning the same KV budget to all heads may over-compress important heads while wasting cache on less important ones. As shown in Fig. 3(a), we estimate head importance by masking each layer-head pair during denoising rollout and measuring the induced DM loss. The head-importance estimation is performed once offline, and the resulting KV budget table is reused during inference.

For a layer-head pair $m = (\ell, h)$, we mask the output of this head while keeping other computations unchanged, obtaining a perturbed latent trajectory $\tilde{\mathbf{z}}^{(m)}$. The importance of head m is measured by the average DM loss over prompts and sampled temporal windows:

$$I_m = \frac{1}{|\mathcal{P}|\,|\mathcal{W}|} \sum_{p\in\mathcal{P}} \sum_{w\in\mathcal{W}} \mathcal{L}_{\mathrm{DM}}\left(\tilde{\mathbf{z}}^{(m)}_{p,w}\right), \tag{5}$$

where $\mathcal{P}$ denotes the number of prompts, and $\mathcal{W}$ denotes sampled temporal windows from the generated latent trajectory. A larger I_m indicates that masking head m causes stronger generation degradation. The detailed algorithmic procedure is provided in App. B.1.

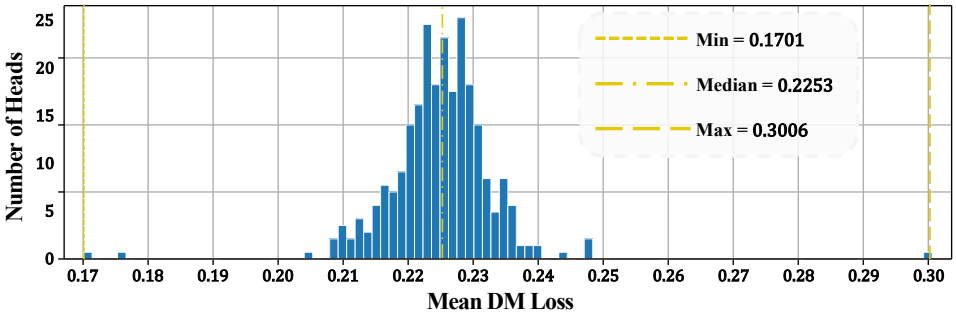


Figure 4: **Head-wise importance is non-uniform.** Each bar counts the number of heads falling into its DM-loss interval, and the vertical lines mark the minimum, median, and maximum values.

The DM loss distribution in Fig. 4 is clearly non-uniform. For each head, we average the DM loss over sampled temporal windows and prompts, and report the resulting mean DM loss as its importance score. The histogram shows that most heads cluster in a narrow range around the median, while a few heads have much higher or lower mean loss, forming a long-tailed distribution. The mean loss reflects the expected degradation caused by masking a head and is used as the head-importance score for budget allocation.

After obtaining $\{I_m\}$ for all layer-head pairs, we normalize the importance scores:

$$\hat{I}_m = \frac{I_m - I_{\min}}{I_{\max} - I_{\min} + \epsilon}. \tag{6}$$

We then map each normalized score to a discrete KV budget:

$$b_m = \mathrm{round}\left(b_{\min} + \hat{I}_m^{\gamma}\,(b_{\max} - b_{\min})\right), \tag{7}$$

where $b_{\min}$ and $b_{\max}$ are the minimum and maximum budgets, and γ controls the curvature of the mapping. More important heads therefore retain more historical frames.

3.3 Query-Frame-Wise History Scoring and Selection

Frames within the same generated chunk may require different historical context. Focused Forcing therefore selects historical frames separately for each query frame and head by combining attention scores with diversity scores under the assigned head-wise budget, as shown in Fig. 3(b).

Attention Scores. For efficiency, we group tokens within each frame into P groups. The grouped query and key features are represented as $\bar{Q} \in \mathbb{R}^{B\times QF\times P\times H\times D}$ and $\bar{K} \in \mathbb{R}^{B\times KF\times P\times H\times D}$. Let $\bar{Q}_{b,q_f,u,h}$ and $\bar{K}_{b,k_f,v,h}$ denote the pooled features for query frame q_f, historical frame k_f, group indices u, v, and head h. The frame-level attention score is defined as:

$$A_{b,q_f,h,k_f} = \frac{1}{P^2} \sum_{u=1}^{P} \sum_{v=1}^{P} \frac{\langle \bar{Q}_{b,q_f,u,h}, \bar{K}_{b,k_f,v,h} \rangle}{\sqrt{D}}, \tag{8}$$

where D is the head dimension. We standardize A_{b,q_f,h,k_f} along the historical-frame dimension for each batch, query frame, and head, so that scores over candidate historical frames have zero mean and unit variance, yielding $\tilde{A}_{b,q_f,h,k_f}$.

Diversity Scores of Historical Frames. Attention scores may not fully capture frame distinctiveness, especially when they vary with relative temporal distance, as proved in App. B.2. We therefore compute diversity scores from historical key frames. Let $K^{(k_f)} \in \mathbb{R}^{B\times L\times H\times D}$ denote the key of the k_f-th historical frame, where L is the number of tokens per frame. For each batch index b, token index l, and head index h, we compute the average key representation over historical frames and measure the redundancy of historical frame k_f by its average cosine similarity to this representation:

$$\bar{K}^{\text{mean}}_{b,l,h} = \frac{1}{F_h}\sum_{k_f=1}^{F_h} K^{(k_f)}_{b,l,h}, \quad R_{b,h,k_f} = \frac{1}{L}\sum_{l=1}^{L}\left\langle \frac{K^{(k_f)}_{b,l,h}}{\left\|K^{(k_f)}_{b,l,h}\right\|_2+\epsilon}, \frac{\bar{K}^{\text{mean}}_{b,l,h}}{\left\|\bar{K}^{\text{mean}}_{b,l,h}\right\|_2+\epsilon}\right\rangle. \tag{9}$$

Here, ϵ is a small constant for numerical stability. A historical frame is less distinctive if it is closer to the average historical representation. We therefore define the diversity score of historical frame k_f as negative redundancy:

$$D_{b,h,k_f} = -R_{b,h,k_f}. \tag{10}$$

After standardization along the historical-frame dimension, the diversity score of each historical frame is broadcast to each query frame, yielding $\tilde{D}_{b,q_f,h,k_f}$.

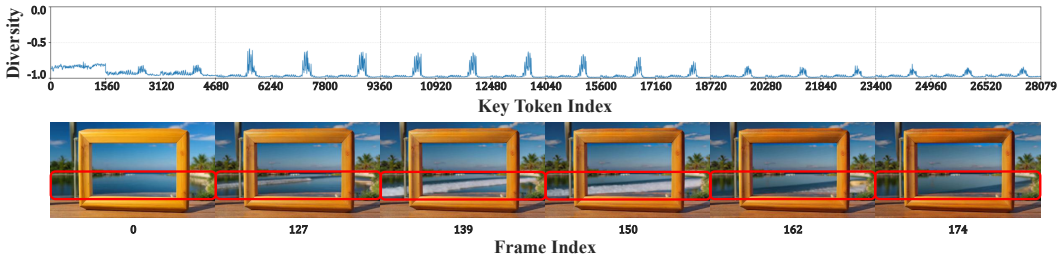


Figure 5: **Diversity scores highlight changing regions.** Historical frames with high diversity scores often contain regions that differ from the average historical representation.

As shown in Fig. 5, high diversity scores highlight regions that differ across historical frames, while static regions stay close to the average representation. Thus, diversity scores complement attention scores by preserving distinctive historical frames that may be under-ranked by attention alone.

Final Score and Selection. We compute the final score by weighting the attention score and the diversity score of each historical frame, and select historical frames under the assigned budget:

$$S_{b,q_f,h,k_f} = \lambda\tilde{A}_{b,q_f,h,k_f} + (1-\lambda)\tilde{D}_{b,q_f,h,k_f}, \quad \mathcal{S}_{b,q_f,h} = \text{TopK}_{k_f}\left(S_{b,q_f,h,k_f}, b_{\ell,h}\right). \tag{11}$$

where λ controls the trade-off between the two scores, and $b_{\ell,h}$ is the budget of the current layer-head pair. In practice, we reserve anchor and generated frames, and use the remaining budget for the highest-scoring historical frames to preserve reference and consistency.

3.4 Packed QKV for Variable-Length FlashAttention

Because the selected historical frames vary across query frames and heads, the resulting KV lengths are irregular. As shown in Fig. 3(c), we pack the corresponding QKV rows into variable-length sequences. For each query frame and head, the query length is fixed, while the key-value length is determined by the assigned budget. We compute attention only over the retained historical frames:

$$O_{\text{pack}} = \text{VarLenFlashAttn}\left(Q_{\text{pack}}, K_{\text{pack}}, V_{\text{pack}}, \text{cu_q}, \text{cu_k}\right), \tag{12}$$

Specifically, each batch, query-frame, and head tuple forms an independent variable-length attention sequence. The selected K/V frame blocks are gathered using identical indices and concatenated into packed tensors, with `cu_q` and `cu_k` recording the sequence boundaries. Variable-length FlashAttention processes these sequences independently, and the outputs are scattered back to their original query positions. This execution preserves the selection across query frames and heads while avoiding dense attention.

4 Experiments

4.1 Settings

Baselines and comparisons. We evaluate Focused Forcing under two settings. To verify compatibility across autoregressive paradigms, we plug it into Self Forcing [21], LongLive [46], Rolling Forcing [26], and Causal Forcing [59], comparing against their original baselines as well as representative autoregressive diffusion models NOVA [12], SkyReels-V2 [7], and MAGI-1 [2]. For acceleration, we take chunk-wise Self Forcing as the base model and compare with its Attention Sink variant, MonarchRT [1], TaylorSeer [25], and Dummy Forcing [16], which respectively represent structured sparse attention, cached-feature-based prediction, and head-wise KV cache pruning. Implementation details are provided in App. C.1.

Evaluation. Following prior work [9], we evaluate long video generation on VBench-Long [22] using the first 128 prompts from MovieGen [34] refined by Qwen2.5-7B-Instruct [45]. We follow the VBench-Long protocol to assess generation quality, and further evaluate inference efficiency using the average generation latency, average self-attention latency, and their corresponding speedups. Detailed metric definitions are provided in App. C.2.

4.2 Comparison with Autoregressive Video Diffusion Models

Quantitative results. We compare Focused Forcing with state-of-the-art autoregressive video models on VBench-Long. As shown in Tab. 1, Focused Forcing consistently improves the efficiency-quality trade-off across different autoregressive paradigms. On Self Forcing, it achieves 1.45× end-to-end speedup, improves visual quality from 76.58 to 80.00, and substantially increases Dynamic Degree from 41.55 to 61.39. Similar improvements on LongLive, Rolling Forcing, and Causal Forcing show that Focused Forcing generalizes across different autoregressive backbones.

Table 1: Quantitative comparison for long video generation on VBench-Long.

Methods	Gen. Latency/s	Gen. Speedup	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Visual Quality	Text Alignment
Autoregressive Diffusion Models										
NOVA	2558.21	1.00×	90.15	93.35	98.81	34.08	44.14	38.81	66.56	24.63
SkyReels-V2	1244.40	1.00×	96.93	96.15	98.68	38.22	58.86	63.42	75.38	27.54
MAGI-1	6740.65	1.00×	97.29	96.53	99.10	39.10	57.93	59.24	74.87	29.41
Distilled Autoregressive Diffusion Models										
CausVid	61.69	1.00×	97.88	96.82	98.08	48.61	59.94	65.79	77.85	27.19
Self Forcing	78.06	1.00×	96.61	96.17	98.27	41.55	58.83	68.04	76.58	28.03
+ Ours	53.90	1.45×	95.84	95.49	98.04	61.39	60.59	68.62	80.00	28.75
LongLive	90.06	1.00×	97.60	96.36	98.90	35.42	60.74	68.06	76.18	29.06
+ Ours	68.60	1.31×	96.26	95.79	98.30	46.03	61.51	69.58	77.91	29.40
Rolling Forcing	83.35	1.00×	97.88	96.47	98.77	29.67	60.69	70.76	75.71	28.85
+ Ours	58.20	1.43×	96.21	95.40	97.91	50.03	59.77	68.47	77.97	28.99
Causal Forcing	77.99	1.00×	95.66	94.34	97.56	94.05	56.46	68.35	84.40	26.80
+ Ours	52.68	1.48×	94.53	94.11	96.63	97.77	57.54	68.29	84.81	28.27

Qualitative results. As shown in Fig. 6, Self Forcing suffers from noticeable long-horizon degradation, with the dog appearance and background details becoming increasingly unstable over time. Although Long-Live better preserves the overall scene layout, it exhibits evident color degradation in later frames. Focused Forcing leads to more stable visual results in these cases, with the corgi's appearance and background better preserved. For Rolling Forcing and Causal Forcing, whose baselines already produce relatively stable results, the visual improvement is less pronounced. Nevertheless, Focused Forcing preserves comparable visual quality while substantially improving inference efficiency, indicating its effectiveness as a general acceleration module for long-video generation.

4.3 Comparison with Acceleration Methods

Quantitative results. We further compare Focused Forcing with representative acceleration methods on chunk-wise Self Forcing. Shown in Tab. 2, Focused Forcing achieves 2.77× attention speedup and 1.45× end-to-end speedup, while delivering the best visual quality among all acceleration methods. It improves visual quality over Attention Sink from 79.2 to 80.0 with substantial speedup,



Figure 6: Qualitative comparison on 30s video generation across different autoregressive paradigms.

surpasses MonarchRT and TaylorSeer in both quality and end-to-end speedup, and remains competitive with Dummy Forcing in efficiency while achieving higher visual quality. Fig. 8 further highlights its strong Dynamic Degree and competitive performance across other quality dimensions, confirming a stronger speed-quality trade-off through selective KV-context compression.

Table 2: Efficiency comparison for long video generation on VBench-Long.

Methods	Attn. Latency/s	Attn. Speedup	Gen. Latency/s	Gen. Speedup	Temporal Quality	Frame-wise Quality	Visual Quality	Text Alignment
Self Forcing	37.29	1.00×	78.06	1.00×	83.15	63.44	76.58	28.03
+ Attention Sink	37.15	1.00×	78.07	1.00×	86.46	64.68	79.20	28.42
MonarchRT	33.13	1.13×	72.61	1.08×	85.95	64.04	78.65	29.24
TaylorSeer	29.77	1.25×	68.88	1.13×	85.67	64.38	78.57	28.85
Dummy Forcing	13.35	2.79×	53.64	1.46×	84.79	65.55	78.38	28.57
Ours	13.48	2.77×	53.90	1.45×	87.69	64.61	80.00	28.75

Qualitative results. As shown in Fig. 7, Attention Sink preserves the overall scene layout, whereas Self Forcing and TaylorSeer both exhibit glass-shape drift and physically implausible milk-flow positions, and MonarchRT and Dummy Forcing introduce visible artifacts and quality degradation, respectively. In contrast, Focused Forcing yields a more coherent trajectory, with the object stream evolving smoothly throughout the rollout, indicating that our method removes redundant historical information while preserving motion-relevant temporal cues.

4.4 Ablation Studies

We ablate the KV budget and attention weight, with detailed results provided in App. C.3.

Ablation on KV budget. As shown in Fig. 9, Self Forcing is the uncompressed baseline. For Focused Forcing, each colored curve represents a fixed minimum KV budget $b_{\min}$, as indicated by the legend, e.g., "min 4" denotes $b_{\min} = 4$. The markers on the same curve correspond to different maximum KV budgets $b_{\max}$, which are decreased from 15 with a step size of 3. Moving along a



Figure 7: Qualitative comparison of inference acceleration methods on 30s video generation.

curve toward larger attention speedup therefore indicates stronger KV compression. The results show that moderate budget settings achieve the best trade-off, substantially improving attention speedup over Self Forcing while maintaining high visual quality, whereas overly aggressive compression leads to noticeable quality degradation.

Ablation on attention weight. As shown in Fig. 10, all settings outperform Self Forcing in both visual quality and text alignment. Visual quality improves as the attention weight increases and peaks at a moderate value, while larger weights slightly reduce both metrics. This indicates that combining attention with diversity is important for selecting useful yet non-redundant historical context.

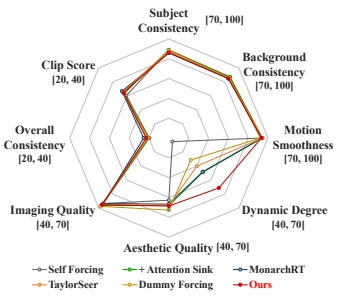


Figure 8: Metric comparison with acceleration methods.

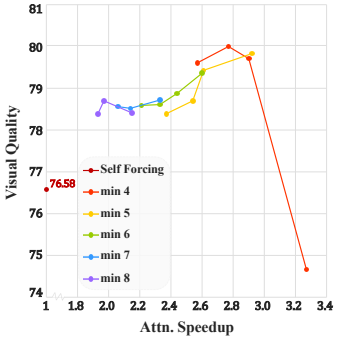


Figure 9: Ablation study on the KV budget.

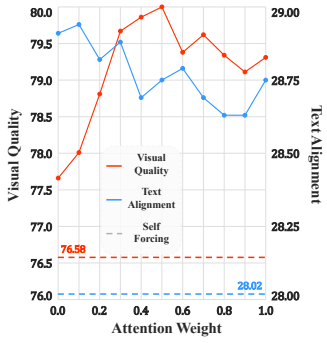


Figure 10: Ablation study on the attention weight.

5 Conclusion

In this paper, we present **Focused Forcing**, a training-free KV compression method for autoregressive video generation. By reformulating KV compression as fine-grained context allocation, Focused Forcing selects historical frames separately for each query frame, combines attention scores with diversity scores of historical frames, and assigns differentiated KV budgets according to head importance. This design preserves relevant and distinctive historical context under compact KV budgets. Experiments on VBench-Long across multiple autoregressive paradigms show that Focused Forcing consistently improves the efficiency-quality trade-off, achieving up to **1.48**× end-to-end acceleration without training while **improving visual quality and text alignment**. These results show that fine-grained and content-aware KV selection is an effective strategy for scalable long-horizon autoregressive video generation.

References

[1] Krish Agarwal, Zhuoming Chen, Cheng Luo, Yongqi Chen, Haizhong Zheng, Xun Huang, Atri Rudra, and Beidi Chen. Monarchrt: Efficient attention for real-time video generation, 2026. URL https://arxiv.org/abs/2602.12271.

[2] Sand. ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. Magi-1: Autoregressive video generation at scale, 2025. URL https://arxiv.org/abs/2505.13211.

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[5] Peiliang Cai, Jiacheng Liu, Haowen Xu, Xinyu Wang, Chang Zou, and Linfeng Zhang. Lesa: Learnable stage-aware predictors for diffusion model acceleration, 2026. URL https://arxiv.org/abs/2602.20497.

[6] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[7] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. URL https://arxiv.org/abs/2504.13074.

[8] Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

[9] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DzvPiqh23f.

[10] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 35549–35562, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/98ed250b203d1ac6b24bbcf263e3d4a7-Paper-Conference.pdf.

[11] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

[12] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JE9tCwe3lp.

[13] Jiarui Fang, Jinzhe Pan, Aoyu Li, Xibo Sun, and WANG Jiannan. Pipefusion: Patch-level pipeline parallelism for diffusion transformers inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[14] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2025. URL https://arxiv.org/abs/2410.12557.

[15] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025. URL https://arxiv.org/abs/2505.13447.

[16] Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *arXiv preprint arXiv:2601.20499*, 2026.

[17] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[18] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model, 2026. URL https://arxiv.org/abs/2601.03233.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[20] Connor Holmes, Masahiro Tanaka, Michael Wyatt, Ammar Ahmad Awan, Jeff Rasley, Samyam Rajbhandari, Reza Yazdani Aminabadi, Heyang Qin, Arash Bakhtiari, Lev Kurilenko, and Yuxiong He. Deepspeed-fastgen: High-throughput text generation for llms via mii and deepspeed-inference, 2024. URL https://arxiv.org/abs/2401.08671.

[21] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=mSiN7i0BYH.

[22] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3268–3285, 2026. doi: 10.1109/TPAMI.2025.3633890.

[23] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, 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 (CVPR)*, 2024.

[24] 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. *arXiv preprint arXiv:2411.19108*, 2024.

[25] Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15853–15863, October 2025.

[26] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IAyzXjbfwo.

[27] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL http://dx.doi.org/10.1007/s11633-025-1562-4.

[28] 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.

[29] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[30] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4117–4125, 2024.

[31] Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Bingyuan Wang, Qinghe Wang, Xuanhua He, Hongfa Wang, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

[32] Yue Ma, Yulong Liu, Qiyuan Zhu, Ayden Yang, Kunyu Feng, Xinhua Zhang, Zexuan Yan, Zhifeng Li, Sirui Han, Chenyang Qi, et al. Follow-your-motion: Video motion transfer via efficient spatial-temporal decoupled finetuning. *arXiv preprint arXiv:2506.05207*, 2025.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

[34] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025. URL https://arxiv.org/abs/2410.13720.

[35] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

[36] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, Mojie Chi, Xuyan Chi, Jian Cong, Qinpeng Cui, Fei Ding, Qide Dong, Yujiao Du, Haojie Duanmu, Junliang Fan, Jiarui Fang, Jing Fang, Zetao Fang, Chengjian Feng, Yu Gao, Diandian Gu, Dong Guo, Hanzhong Guo, Qiushan Guo, Boyang Hao, Hongxiang Hao, Haoxun He, Jiaao He, Qian He, Tuyen Hoang, Heng Hu, Ruoqing Hu, Yuxiang Hu, Jiancheng Huang, Weilin Huang, Zhaoyang Huang, Zhongyi Huang, Jishuo Jin, Ming Jing, Ashley Kim, Shanshan Lao, Yichong Leng, Bingchuan Li, Gen Li, Haifeng Li, Huixia Li, Jiashi Li, Ming Li, Xiaojie Li, Xingxing Li, Yameng Li, Yiying Li, Yu Li, Yueyan Li, Chao Liang, Han Liang, Jianzhong Liang, Ying Liang, Wang Liao, J. H. Lien, Shanchuan Lin, Xi Lin, Feng Ling, Yue Ling, Fangfang Liu, Jiawei Liu, Jihao Liu, Jingtuo Liu, Shu Liu, Sichao Liu, Wei Liu, Xue Liu, Zuxi Liu, Ruijie Lu, Lecheng Lyu, Jingting Ma, Tianxiang Ma, Xiaonan Nie, Jingzhe Ning, Junjie Pan, Xitong Pan, Ronggui Peng, Xueqiong Qu, Yuxi Ren, Yuchen Shen, Guang Shi, Lei Shi, Yinglong Song, Fan Sun, Li Sun, Renfei Sun, Wenjing Tang, Boyang Tao, Zirui Tao, Dongliang Wang, Feng Wang, Hulin Wang, Ke Wang, Qingyi Wang, Rui Wang, Shuai Wang, Shulei Wang, Weichen Wang, Xuanda Wang, Yanhui Wang, Yue

Wang, Yuping Wang, Yuxuan Wang, Zijie Wang, Ziyu Wang, Guoqiang Wei, Meng Wei, Di Wu, Guohong Wu, Hanjie Wu, Huachao Wu, Jian Wu, Jie Wu, Ruolan Wu, Shaojin Wu, Xiaohu Wu, Xinglong Wu, Yonghui Wu, Ruiqi Xia, Xin Xia, Xuefeng Xiao, Shuang Xu, Bangbang Yang, Jiaqi Yang, Runkai Yang, Tao Yang, Yihang Yang, Zhixian Yang, Ziyan Yang, Fulong Ye, Bingqian Yi, Xing Yin, Yongbin You, Linxiao Yuan, Weihong Zeng, Xuejiao Zeng, Yan Zeng, Siyu Zhai, Zhonghua Zhai, Bowen Zhang, Chenlin Zhang, Heng Zhang, Jun Zhang, Manlin Zhang, Peiyuan Zhang, Shuo Zhang, Xiaohe Zhang, Xiaoying Zhang, Xinyan Zhang, Xinyi Zhang, Yichi Zhang, Zixiang Zhang, Haiyu Zhao, Huating Zhao, Liming Zhao, Yian Zhao, Guangcong Zheng, Jianbin Zheng, Xiaozheng Zheng, Zerong Zheng, Kuan Zhu, and Feilong Zuo. Seedance 2.0: Advancing video generation for world complexity, 2026. URL https://arxiv.org/abs/2604.14148.

[37] Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration, 2024. URL https://arxiv.org/abs/2407.01425.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[39] Junshu Tang, Jiacheng Liu, Jiaqi Li, Longhuang Wu, Haoyu Yang, Penghao Zhao, Siruis Gong, Xiang Yuan, Shuai Shao, Linfeng Zhang, and Qinglin Lu. Hunyuan-gamecraft-2: Instruction-following interactive game world model, 2026. URL https://arxiv.org/abs/2511.23429.

[40] Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

[41] Team Wan, 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, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong 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.

[42] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, Xiao He, Xin Li, Xinchi Deng, Xuefei Zhe, Yang Li, Yanxin Long, Yuanbo Peng, Yue Wu, Yuhong Liu, Zhenyu Wang, Zuozhuo Dai, Bo Peng, Coopers Li, Gu Gong, Guojian Xiao, Jiahe Tian, Jiaxin Lin, Jie Liu, Jihong Zhang, Jiesong Lian, Kaihang Pan, Lei Wang, Lin Niu, Mingtao Chen, Mingyang Chen, Mingzhe Zheng, Miles Yang, Qiangqiang Hu, Qi Yang, Qiuyong Xiao, Runzhou Wu, Ryan Xu, Rui Yuan, Shanshan Sang, Shisheng Huang, Siruis Gong, Shuo Huang, Weiting Guo, Xiang Yuan, Xiaojia Chen, Xiawei Hu, Wenzhi Sun, Xiele Wu, Xianshun Ren, Xiaoyan Yuan, Xiaoyue Mi, Yepeng Zhang, Yifu Sun, Yiting Lu, Yitong Li, You Huang, Yu Tang, Yixuan Li, Yuhang Deng, Yuan Zhou, Zhichao Hu, Zhiguang Liu, Zhihe Yang, Zilin Yang, Zhenzhi Lu, Zixiang Zhou, and Zhao Zhong. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

[43] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 21875–21895, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5e5fd18f863cbe6d8ae392a93fd271c9-Paper-Conference.pdf.

[44] Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, junxian guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context LLM inference with retrieval and streaming

heads. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

[45] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[46] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCAODkpsPJ.

[47] 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.

[48] Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

[49] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 47455–47487. Curran Associates, Inc., 2024. doi: 10.52202/079017-1505. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54dcf25318f9de5a7a01f0a4125c541e-Paper-Conference.pdf.

[50] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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 (CVPR)*, pages 6613–6623, June 2024.

[51] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22963–22974, June 2025.

[52] Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

[53] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization, 2025. URL https://arxiv.org/abs/2411.10958.

[54] Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration, 2025. URL https://arxiv.org/abs/2410.02367.

[55] Songchun Zhang, Zeyue Xue, Siming Fu, Jie Huang, Xianghao Kong, Y Ma, Haoyang Huang, Nan Duan, and Anyi Rao. Astrolabe: Steering forward-process reinforcement learning for distilled autoregressive video models. *arXiv preprint arXiv:2603.17051*, 2026.

[56] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in*

Neural Information Processing Systems, volume 36, pages 34661–34710. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

[57] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[58] Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation, 2026. URL https://arxiv.org/abs/2603.21366.

[59] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

[60] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yYZbZGo4ei.

A Limitations

Focused Forcing is designed as a training-free KV selection method for efficient autoregressive video diffusion. Experiments evaluate it under the VBench-Long protocol and several representative autoregressive video diffusion paradigms, where it demonstrates consistent speed-quality improvements. Broader evaluation on higher resolutions, other model architectures, and more diverse deployment settings would further validate its generality. Moreover, the current implementation adopts fixed KV-budget bounds and a fixed attention-diversity weight, which provide a favorable trade-off in our experiments but may be further adapted to different models, resolutions, or latency targets. The realized acceleration may also vary across hardware platforms and attention kernel implementations.

B Method Details

B.1 Estimation of Head Importance for Budget Allocation

Algorithm 1 Head Importance Estimation and KV-Budget Allocation

Require: Prompt set $\mathcal{P} = \{p_m\}_{m=1}^{M}$; number of layers L; number of heads per layer H; rollout window length K; number of sampled windows N_w; minimum and maximum KV budgets $b_{\min}, b_{\max}$; budget curvature γ; fake-score model S_{fake}; real-score model S_{real}; CFG scale s_{cfg}

Ensure: Fixed per-head KV budget table Π_{KV}

Initialize head score table $\mathcal{S} \in \mathbb{R}^{L \times H}$ to zeros
for $m = 1, \ldots, M$ **do**
  Sample a random seed s_m for prompt p_m
  Construct $B = L \times H$ rollouts, with rollout b masking head $(\lfloor b/H \rfloor,\ b \bmod H)$
  Run denoising rollout and obtain masked latent trajectories $\{\mathbf{Z}^{(b)}\}_{b=0}^{B-1}$
  for $b = 0$ to $B - 1$ **do**
    $\text{DMsum} \leftarrow 0$ ▷ Accumulates DM losses over sampled windows.
    Split $\mathbf{Z}^{(b)}$ into N_w temporal windows $\{W_r^{(b)}\}_{r=0}^{N_w-1}$, each with length K
    for $r = 0$ to $N_w - 1$ **do**
      Sample timestep t and Gaussian noise ϵ; add noise $W_{r,t}^{(b)} = \text{AddNoise}(W_r^{(b)}, \epsilon, t)$
      Predict fake clean window $\hat{W}_r^{\text{fake}} = S_{\text{fake}}(W_{r,t}^{(b)}, p_m, t)$
      Predict real clean windows $\hat{W}_{r,\text{cond}}^{\text{real}} = S_{\text{real}}(W_{r,t}^{(b)}, p_m, t)$ and $\hat{W}_{r,\text{uncond}}^{\text{real}} = S_{\text{real}}(W_{r,t}^{(b)}, p_\emptyset, t)$
      Apply CFG: $\hat{W}_r^{\text{real}} = \hat{W}_{r,\text{cond}}^{\text{real}} + s_{\text{cfg}}(\hat{W}_{r,\text{cond}}^{\text{real}} - \hat{W}_{r,\text{uncond}}^{\text{real}})$
      Compute DM gradient $g_r^{(b)} = \hat{W}_r^{\text{fake}} - \hat{W}_r^{\text{real}}$
      Normalize DM gradient $g_r^{(b)} \leftarrow g_r^{(b)} / (|W_r^{(b)} - \hat{W}_r^{\text{real}}|_{\text{mean}} + \epsilon)$
      Compute DM loss $d_r^{(b)} = \frac{1}{2} \left\| W_r^{(b)} - \text{sg}(W_r^{(b)} - g_r^{(b)}) \right\|_2^2$
      $\text{DMsum} \leftarrow \text{DMsum} + d_r^{(b)}$
    end for
    $\mathcal{S}[\ell_b, h_b] \leftarrow \mathcal{S}[\ell_b, h_b] + \text{DMsum}/N_w$
  end for
end for
Average head scores over prompts: $\mathcal{S} \leftarrow \mathcal{S}/M$ ▷ Higher score means stronger degradation.
$S_{\min} \leftarrow \min_{\ell,h} \mathcal{S}[\ell, h]$, $S_{\max} \leftarrow \max_{\ell,h} \mathcal{S}[\ell, h]$
for $\ell = 0$ to $L - 1$ **do**
  for $h = 0$ to $H - 1$ **do**
    Normalize head importance:

$$\hat{S}_{\ell,h} = \frac{\mathcal{S}[\ell, h] - S_{\min}}{S_{\max} - S_{\min} + \epsilon}$$

    Map importance to KV budget:

$$b_{\ell,h} = \text{round}\left(b_{\min} + \hat{S}_{\ell,h}^{\gamma} \, (b_{\max} - b_{\min}) \right)$$

  end for
end for
Construct fixed per-head KV budget table $\Pi_{\text{KV}} = \{b_{\ell,h}\}_{\ell=0,h=0}^{L-1,H-1}$ ▷ The budget table is computed offline.
Freeze Π_{KV} for subsequent inference

Offline Profiling Cost. We use 32 calibration prompts and evaluate 360 layer-head pairs through 10-second autoregressive rollouts. This is a one-time offline cost and is excluded from online inference latency. The resulting budget table can be reused as long as the model weights and attention architecture remain unchanged.

B.2 Proof of the Temporal Effect of Causal RoPE

We show that causal RoPE makes the attention logit explicitly depend on the relative temporal distance between a query frame and a key frame. This explains why the same historical frame can receive different attention scores as its relative temporal distance to the current frame changes.

Let $d = 2m$ be the dimension of one attention head. RoPE groups a feature vector into m two-dimensional blocks. For the j-th block, define the rotation matrix

$$R(\theta) := \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

In causal RoPE, a token at frame index t is assigned a temporal rotary phase. Let $\mathcal{T}$ denote the index set of two-dimensional rotary blocks associated with temporal encoding, and let ω_j be the temporal frequency for $j \in \mathcal{T}$. For a token at frame index t, the j-th temporal block is rotated by angle $\omega_j t$.

For a query token at frame t_q and a key token at frame t_k, denote their pre-RoPE representations in the j-th temporal block as

$$q_j := \begin{bmatrix} q_{2j-1} \\ q_{2j} \end{bmatrix}, \qquad k_j := \begin{bmatrix} k_{2j-1} \\ k_{2j} \end{bmatrix}.$$

After applying temporal RoPE, we have

$$\widetilde{q}_j = R(\omega_j t_q) q_j, \qquad \widetilde{k}_j = R(\omega_j t_k) k_j.$$

Proposition. The temporal RoPE contribution to the attention logit is an explicit function of the relative temporal distance $\Delta t = t_k - t_q$. In particular,

$$\ell_T(\Delta t) = \frac{1}{\sqrt{d}} \sum_{j \in \mathcal{T}} \left[A_j \cos(\omega_j \Delta t) + B_j \sin(\omega_j \Delta t) \right],$$

where

$$A_j = q_{2j-1} k_{2j-1} + q_{2j} k_{2j}, \qquad B_j = q_{2j} k_{2j-1} - q_{2j-1} k_{2j}.$$

Proof. The contribution of the j-th temporal block to the attention logit is

$$\widetilde{q}_j^\top \widetilde{k}_j = \left(R(\omega_j t_q) q_j \right)^\top \left(R(\omega_j t_k) k_j \right).$$

Using the identity $R(a)^\top R(b) = R(b-a)$, we obtain

$$\widetilde{q}_j^\top \widetilde{k}_j = q_j^\top R(\omega_j t_q)^\top R(\omega_j t_k) k_j = q_j^\top R(\omega_j (t_k - t_q)) k_j.$$

Let $\Delta t = t_k - t_q$. Then

$$\widetilde{q}_j^\top \widetilde{k}_j = q_j^\top R(\omega_j \Delta t) k_j.$$

Expanding the rotation matrix gives

$$\begin{aligned} q_j^\top R(\omega_j \Delta t) k_j &= \begin{bmatrix} q_{2j-1} & q_{2j} \end{bmatrix} \begin{bmatrix} \cos(\omega_j \Delta t) & -\sin(\omega_j \Delta t) \\ \sin(\omega_j \Delta t) & \cos(\omega_j \Delta t) \end{bmatrix} \begin{bmatrix} k_{2j-1} \\ k_{2j} \end{bmatrix} \\ &= (q_{2j-1} k_{2j-1} + q_{2j} k_{2j}) \cos(\omega_j \Delta t) + (q_{2j} k_{2j-1} - q_{2j-1} k_{2j}) \sin(\omega_j \Delta t). \end{aligned}$$

Substituting the definitions of A_j and B_j, we have

$$q_j^\top R(\omega_j \Delta t) k_j = A_j \cos(\omega_j \Delta t) + B_j \sin(\omega_j \Delta t).$$

Summing over all temporal rotary blocks and applying the standard attention scaling factor $1/\sqrt{d}$, we obtain

$$\ell_T(\Delta t) = \frac{1}{\sqrt{d}} \sum_{j \in \mathcal{T}} \left[A_j \cos(\omega_j \Delta t) + B_j \sin(\omega_j \Delta t) \right].$$

Therefore, the temporal part of the attention logit explicitly depends on the relative temporal distance Δt. This proves the proposition.

General case. In the non-degenerate case, there exists at least one temporal block $j \in \mathcal{T}$ such that $q_j \neq 0$ and $k_j \neq 0$. For this block,

$$A_j^2 + B_j^2 = \left(q_{2j-1}^2 + q_{2j}^2\right)\left(k_{2j-1}^2 + k_{2j}^2\right) = \|q_j\|^2 \|k_j\|^2 > 0.$$

Thus, at least one of A_j and B_j is nonzero. Therefore,

$$A_j \cos(\omega_j \Delta t) + B_j \sin(\omega_j \Delta t)$$

is not a constant function of Δt. Hence, in general, changing the relative temporal distance changes the attention logit. Since attention weights are computed by a softmax, such logit changes may be further amplified in the final attention distribution.

Special cases. The temporal effect may disappear or become weak in the following degenerate cases.

Case 1: Same temporal position. If $t_q = t_k$, then $\Delta t = 0$ and $R(\omega_j \Delta t) = I$. Hence,

$$q_j^\top R(\omega_j \Delta t) k_j = q_j^\top k_j.$$

In this case, temporal RoPE does not alter the inner product.

Case 2: Zero temporal components. If $q_j = 0$ or $k_j = 0$ for all temporal blocks $j \in \mathcal{T}$, then all temporal coefficients vanish, namely $A_j = B_j = 0$. Hence,

$$\ell_T(\Delta t) = 0.$$

In this case, temporal RoPE has no effect on the attention logit.

Case 3: Periodic phase alignment. If two temporal distances Δt_1 and Δt_2 satisfy

$$\omega_j(\Delta t_2 - \Delta t_1) \in \{2\pi z : z = 0, \pm 1, \pm 2, \ldots\}$$

for all active temporal frequencies j, then the temporal phases are identical. Hence,

$$\ell_T(\Delta t_1) = \ell_T(\Delta t_2).$$

Case 4: Frequency cancellation. Different frequency components may partially cancel each other at particular values of Δt, reducing the overall temporal effect.

B.3 Computational Complexity and Extra Memory Overhead

We analyze the computation and extra memory introduced by packed variable-length FlashAttention. In each attention call, the current chunk contains $QF = 3$ query frames. The model has $L = 30$ layers and $H = 12$ heads per layer. For each layer-head pair (ℓ, h), Focused Forcing assigns a KV budget $b_{\ell,h}$, which denotes the number of historical key-value frames retained per query frame. With $b_{\min} = 4$, $b_{\max} = 12$, and $\gamma = 2$, the allocated budgets satisfy

$$\sum_{\ell=0}^{29} \sum_{h=0}^{11} b_{\ell,h} = 1958.$$

Thus, the packed frame-level attention cost is

$$C_{\text{pack}} = QF \sum_{\ell,h} b_{\ell,h} = 3 \times 1958 = 5874.$$

For dense attention with a 21-frame KV window,

$$C_{\text{dense}} = L \times H \times QF \times F_{\text{dense}} = 30 \times 12 \times 3 \times 21 = 22680.$$

Therefore, packed attention uses

$$\frac{C_{\text{pack}}}{C_{\text{dense}}} = \frac{5874}{22680} = 0.259.$$

of the dense frame-level attention computation, corresponding to a 74.1% reduction and a theoretical frame-level speedup of $3.86\times$.

We next estimate the extra QKV packing memory. Each frame contains $N = 1560$ tokens. The per-head dimension is $D = 128$, and BF16 uses $s = 2$ bytes per element. Thus, one frame in one head occupies

$$N \times D \times s = 1560 \times 128 \times 2 = 399360 \text{ bytes} = 0.381 \text{ MiB}.$$

The packed query tensor contains all $QF = 3$ query frames and all $H = 12$ heads:

$$M_Q = QF \times H \times N \times D \times s = 13.71 \text{ MiB}.$$

For layer ℓ, let $S_\ell = \sum_{h=0}^{11} b_{\ell,h}$, the packed key and value tensors together require

$$M_{K+V}^{(\ell)} = 2 \times QF \times S_\ell \times N \times D \times s.$$

Hence the total extra packing memory in layer ℓ is

$$M_{\text{pack}}^{(\ell)} = M_Q + M_{K+V}^{(\ell)}.$$

From the allocated budgets, the layer-wise sums satisfy

$$S_{\min} = 61, \qquad S_{\text{avg}} = \frac{1958}{30} = 65.27, \qquad S_{\max} = 72.$$

Substituting these values gives

$$M_{\text{pack}}^{\min} = 153.10 \text{ MiB}, \qquad M_{\text{pack}}^{\text{avg}} = 162.86 \text{ MiB}, \qquad M_{\text{pack}}^{\max} = 178.24 \text{ MiB}.$$

Therefore, when only Q_{pack}, K_{pack}, and V_{pack} are counted, the extra packing memory per attention call is

$$153.10 \text{ MiB} \le M_{\text{pack}}^{(\ell)} \le 178.24 \text{ MiB},$$

with an average of 162.86 MiB. This overhead is layer-local because the packed tensors are temporary buffers used only during the current attention computation.

C Experimental Results and Details

C.1 Implementation Details

For each prompt, all methods generate an approximately 30s video with 486 frames at 16 FPS and a resolution of 832×480, except that NOVA is evaluated at 768×480 due to its fixed training shape. All experiments are conducted on 8 NVIDIA A100 GPUs. For Focused Forcing, unless otherwise specified, we set the KV budget to $(b_{\min}, b_{\max}) = (4, 12)$ and the attention weight to 0.5.

C.2 Evaluation Metrics

For generation quality, we report six visual metrics, including Subject Consistency, Background Consistency, Motion Smoothness, Dynamic Degree, Aesthetic Quality, and Imaging Quality. We further report aggregated scores, where Temporal Quality averages the first four metrics, Frame-wise Quality averages the last two, and Visual Quality averages all six. Text Alignment is computed by averaging Overall Consistency and CLIP Score.

C.3 Detailed Ablation Results

Ablation on KV budget. The KV budget defines the head-wise capacity for retaining historical KV entries and therefore controls the trade-off between temporal-context preservation and inference efficiency, as reported in Tab. 3. Smaller budgets yield higher attention speedup by aggressively pruning historical KV entries, but they may discard useful historical context and degrade visual quality. For example, $(4, 6)$ achieves the highest attention speedup of $3.27\times$, but its visual quality drops to 74.67. In contrast, larger budgets preserve more temporal context and generally improve quality, while reducing the acceleration gain. Among the tested settings, $(b_{\min}, b_{\max}) = (4, 12)$ achieves the best visual quality of 80.00 with $2.77\times$ attention speedup and $1.45\times$ generation speedup, offering the most favorable balance between quality and efficiency.

Table 3: Ablation experiments of kv budget on VBench-Long.

KV Budget $(b_{\min}, b_{\max})$	Attn. Latency/s	Attn. Speedup	Gen. Latency/s	Gen. Speedup	Temporal Quality	Frame-wise Quality	Visual Quality	Text Alignment
Self Forcing	37.29	1.00×	78.06	1.00×	83.15	63.44	76.58	28.02
(4, 6)	11.42	3.27×	52.11	1.50×	80.78	62.46	74.67	29.44
(4, 9)	12.84	2.90×	53.26	1.47×	87.19	64.76	79.71	29.19
(4, 12)	13.48	2.77×	53.90	1.45×	87.69	64.61	80.00	28.75
(4, 15)	14.51	2.57×	54.96	1.42×	87.15	64.54	79.61	28.55
(5, 6)	12.79	2.92×	53.18	1.47×	87.41	64.66	79.83	28.91
(5, 9)	14.31	2.61×	54.73	1.43×	86.97	64.34	79.43	28.61
(5, 12)	14.71	2.54×	55.12	1.42×	85.59	64.91	78.70	28.64
(5, 15)	15.74	2.37×	56.15	1.39×	85.15	64.88	78.39	28.56
(6, 6)	14.35	2.60×	54.75	1.43×	87.03	64.02	79.36	28.40
(6, 9)	15.27	2.44×	55.69	1.40×	86.07	64.50	78.88	28.50
(6, 12)	16.01	2.33×	56.44	1.38×	85.64	64.53	78.61	28.50
(6, 15)	16.84	2.21×	57.25	1.36×	85.50	64.78	78.59	28.57
(7, 9)	15.99	2.33×	56.43	1.38×	85.94	64.29	78.72	28.52
(7, 12)	17.39	2.14×	57.84	1.35×	85.49	64.59	78.52	28.56
(7, 15)	18.08	2.06×	58.52	1.33×	85.54	64.62	78.56	28.48
(8, 9)	17.38	2.15×	57.86	1.35×	85.39	64.44	78.41	28.61
(8, 12)	18.88	1.97×	59.35	1.32×	85.78	64.56	78.70	28.52
(8, 15)	19.30	1.93×	59.75	1.31×	85.25	64.63	78.38	28.58

The improvement in aggregate Visual Quality under the $(4, 12)$ budget is largely driven by increased Dynamic Degree, rather than uniform improvements across all quality metrics. In contrast, the more conservative $(8, 12)$ and $(8, 15)$ budgets generally offer a better balance between inference efficiency and temporal consistency.

Ablation on attention weight. The attention weight controls the relative importance of attention relevance and key diversity when selecting historical frames, with detailed results reported in Tab. 4. Compared with Self Forcing, all tested settings improve visual quality, indicating that our historical-frame selection consistently benefits generation quality. However, relying on either factor alone is suboptimal: using only diversity improves visual quality from 76.58 to 77.66 but remains clearly worse than balanced settings, while using only attention reaches 79.31 but can overemphasize redundant high-attention frames. Moderate attention weights achieve a better trade-off. In particular, a balanced setting of 0.5 obtains the best visual quality of 80.00, while also maintaining strong text alignment. These results show that attention relevance and key diversity are complementary, and combining them is important for robust historical-frame selection.

Table 4: Ablation experiments of attention weight on VBench-Long.

Attention Weight	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Overall Consistency	Clip Score	Visual Quality	Text Alignment
Self Forcing	96.61	96.17	98.27	41.55	58.83	68.04	23.93	32.12	76.58	28.02
0.0	94.12	94.92	97.86	51.95	59.76	67.36	24.77	33.05	77.66	28.91
0.1	94.44	94.99	97.92	53.20	59.86	67.63	24.84	33.03	78.01	28.94
0.2	95.00	95.08	97.96	56.82	59.99	67.99	24.67	32.97	78.81	28.82
0.3	95.38	95.23	97.99	60.90	60.21	68.33	24.75	33.00	79.67	28.88
0.4	95.69	95.34	97.94	61.75	60.10	68.34	24.45	32.93	79.86	28.69
0.5	95.84	95.49	98.04	61.39	60.59	68.62	24.53	32.97	80.00	28.75
0.6	96.12	95.66	98.16	57.23	60.52	68.58	24.53	33.05	79.38	28.79
0.7	96.22	95.62	98.18	58.27	60.63	68.79	24.37	33.00	79.62	28.69
0.8	96.19	95.68	98.19	56.68	60.64	68.65	24.29	32.97	79.34	28.63
0.9	96.24	95.65	98.22	55.65	60.41	68.51	24.26	33.00	79.11	28.63
1.0	96.32	95.64	98.23	56.75	60.55	68.40	24.40	33.09	79.31	28.75

Ablation on Importance-Aware Budget Allocation. The importance-aware budget allocation controls how historical-frame budgets are distributed across attention heads, as summarized in Tab. 5. Uniform budgets improve over Self Forcing when the budget is sufficiently large, e.g., $(8, 8)$ and $(12, 12)$ reach visual quality scores of 78.85 and 78.86, respectively, but they do not fully exploit the unequal importance of different heads. In contrast, the proposed normal allocation with $(4, 12)$ achieves the best visual quality of 80.00, showing that assigning larger budgets to more important heads better preserves useful historical context. Reverse and random allocation under the same $(4, 12)$

budget range perform worse, confirming that the gain comes from importance-aware allocation rather than merely using non-uniform budgets. These results validate the effectiveness of head-importance-guided KV budget allocation under compact cache budgets.

Table 5: Ablation experiments of head-wise KV budget allocation strategies on VBench-Long.

KV Budget $(b_{\min}, b_{\max})$	Mode	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Overall Consistency	Clip Score	Visual Quality	Text Alignment
Self Forcing	–	96.61	96.17	98.27	41.55	58.83	68.04	23.93	32.12	76.58	28.02
(4, 4)	Normal	91.93	93.88	97.55	42.78	60.15	68.24	25.86	33.78	75.76	29.82
(8, 8)	Normal	96.93	95.94	98.23	53.28	59.86	68.88	24.25	32.80	78.85	28.53
(12, 12)	Normal	97.13	96.10	98.33	52.03	60.37	69.24	24.22	32.89	78.86	28.55
(4, 12)	Normal	95.84	95.49	98.04	61.39	60.59	68.62	24.53	32.97	80.00	28.75
(4, 12)	Reverse	97.00	95.90	98.28	52.60	60.11	68.85	24.33	32.87	78.79	28.60
(4, 12)	Random	96.82	95.86	98.26	52.93	60.40	69.05	23.92	32.72	78.89	28.32

C.4 User Study

We conduct a user study to evaluate the perceptual quality of different acceleration methods under the chunk-wise Self Forcing paradigm. As shown in Fig. 11, each group presents four anonymized videos generated from the same prompt by MonarchRT, TaylorSeer, Dummy Forcing, and our method. Sixteen participants evaluate 10 groups, resulting in 160 preference votes. For each group, participants select the video that best balances visual quality and text alignment.

As reported in Tab. 6, our method achieves the highest preference ratio of 57.50%, outperforming MonarchRT, TaylorSeer, and Dummy Forcing. This indicates that our method is also favored in subjective evaluation.

Table 6: User study of different acceleration methods.

Methods	**User Preference (%)**
MonarchRT	18.12
TaylorSeer	20.00
Dummy Forcing	4.38
Ours	**57.50**

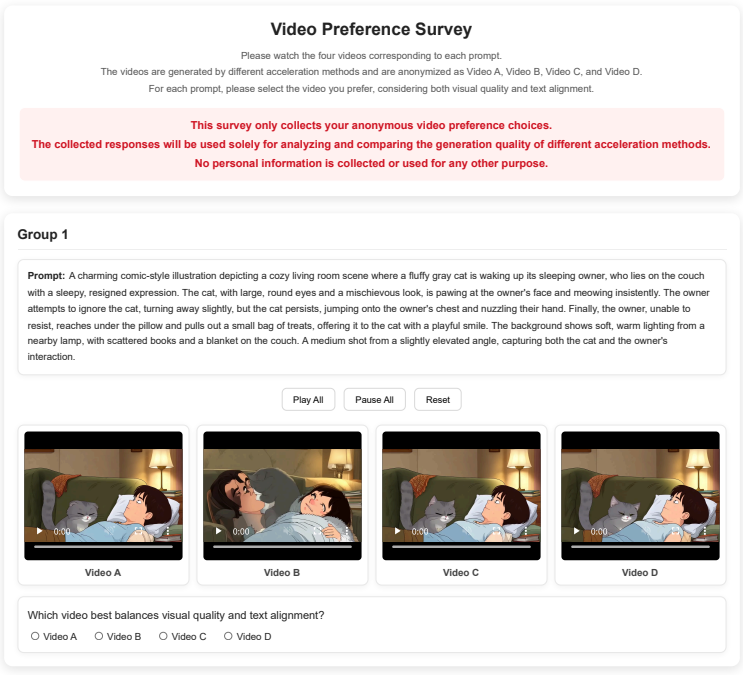


Figure 11: User study interface for comparing videos generated by different acceleration methods.

C.5 Additional Qualitative Results

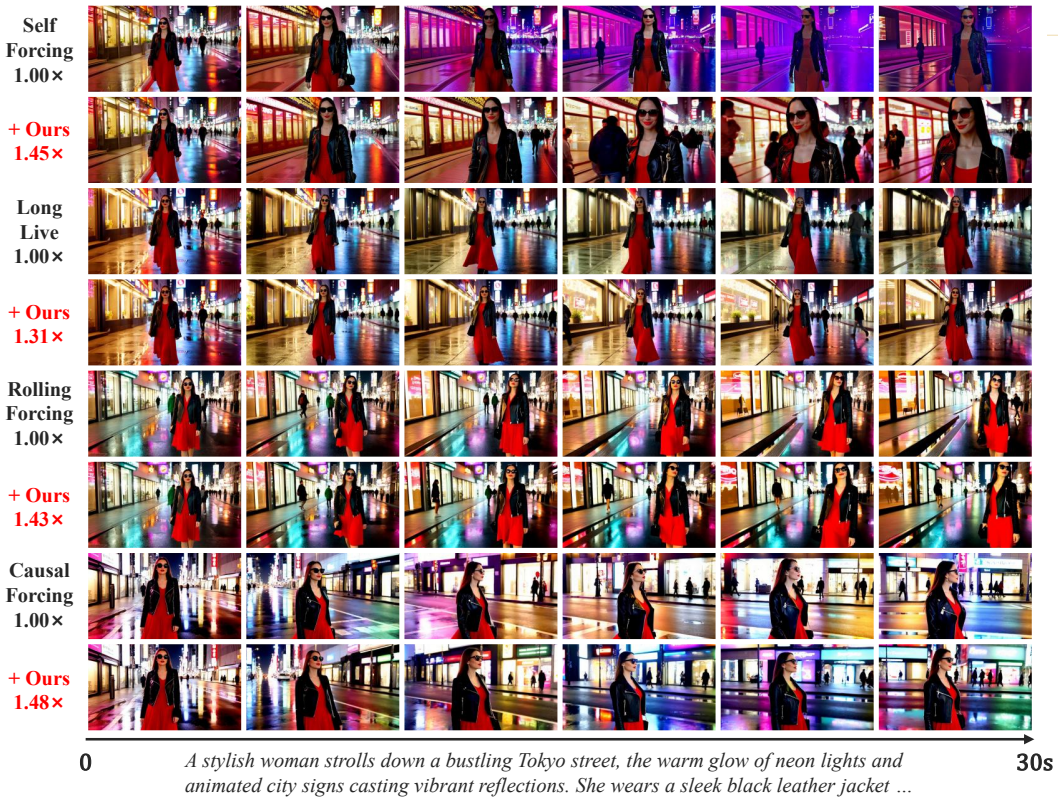


Figure 12: Qualitative examples comparing long-horizon consistency across different paradigms.

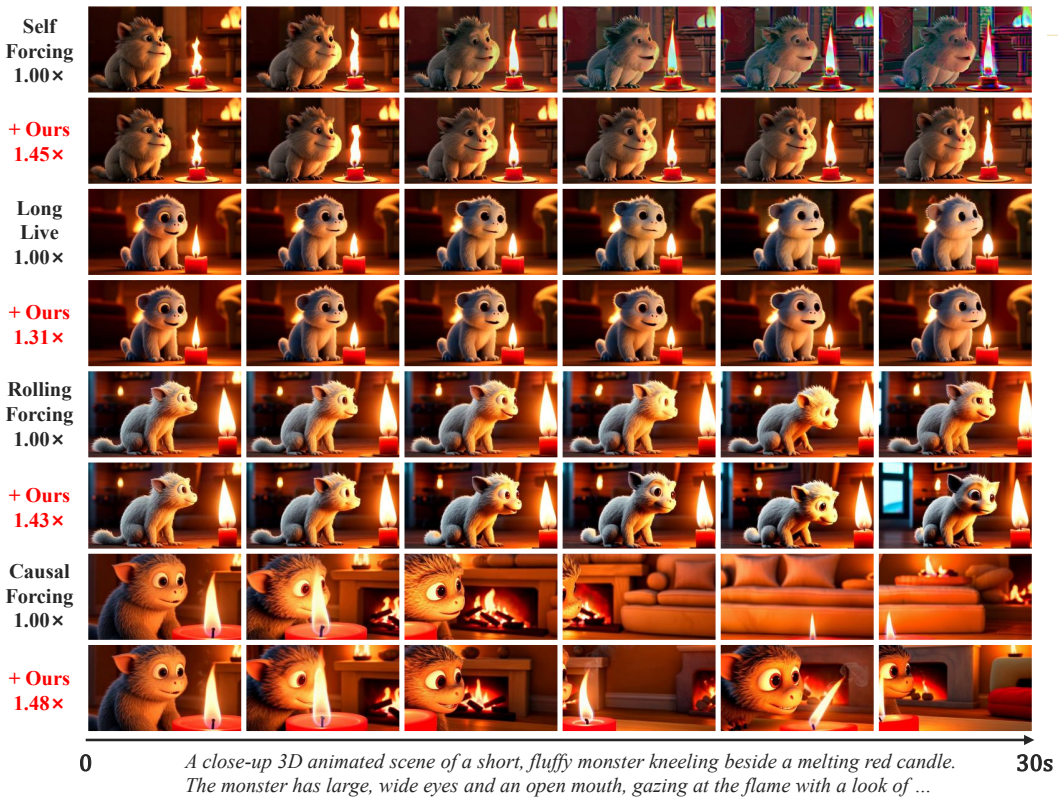


Figure 13: Qualitative examples comparing long-horizon consistency across different paradigms.

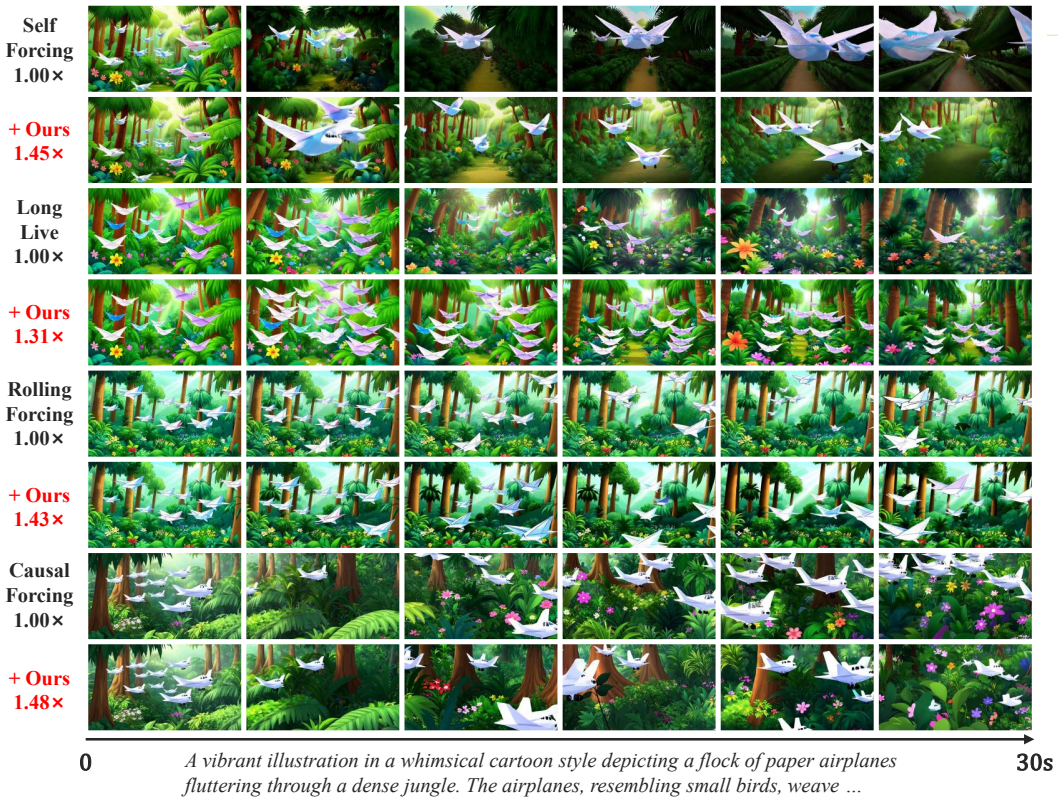


Figure 14: Qualitative examples comparing long-horizon consistency across different paradigms.



Figure 15: Qualitative examples comparing long-horizon consistency across different paradigms.



Figure 16: Qualitative examples comparing quality preservation across acceleration methods.

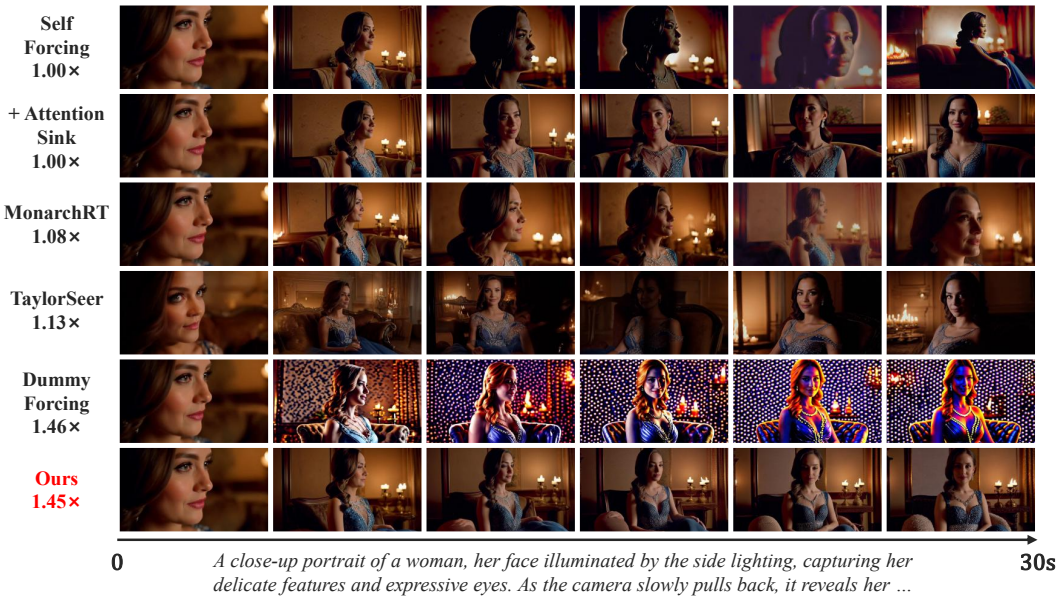


Figure 17: Qualitative examples comparing quality preservation across acceleration methods.



Figure 18: Qualitative examples comparing quality preservation across acceleration methods.



Figure 19: Qualitative examples comparing quality preservation across acceleration methods.

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: As stated in the abstract and Sec. 1, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 paper discusses limitations in App. A.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: The paper provides complete proof for the theoretical result on the effect of causal RoPE in App. B.2. The proof states the RoPE setup, derives the dependence of the attention logit on relative temporal distance, and discusses relevant special cases.

Guidelines:

- The answer [N/A] 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: The paper discloses the proposed training-free algorithm, evaluation setup, baselines, metrics, implementation details, and key hyperparameters needed to reproduce the main results. The supplemental material further provides anonymized code and instructions for reproducing the main experimental results.

Guidelines:

- The answer [N/A] 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: The supplemental material provides anonymized code, environment details, and commands to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: As described in Sec. 4.1, App. C.1 and App. C.2, the paper specifies the experimental setup, baselines, evaluation protocol, metrics, implementation details, and key hyperparameters. Training-specific details such as optimizer settings are not applicable since the proposed method is training-free.

   Guidelines:

   - The answer [N/A] 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: The paper reports quantitative experimental results and ablations, but does not currently include error bars, confidence intervals, or statistical significance tests for the main results.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: The paper reports the GPU type and some inference latency measurements, but does not provide full compute details for every experiment, such as CPU/system configuration, per-run GPU hours or wall-clock time.

   Guidelines:

   - The answer [N/A] 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: The paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include a broader impact discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: The paper does not yet provide a consolidated license table for all external models and code assets used in the experiments.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not introduce a new released dataset or released model asset.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The paper provides the participant instructions and relevant interface screenshots. Participants were informed of the scope of collected information and its intended use before participating.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: Participants were informed of the scope of collected information and its intended use, but the paper does not report IRB approval or an equivalent approval.

    Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core methodology in this research.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.