PYRAMID FORCING: HEAD-AWARE PYRAMID KV CACHE POLICY FOR HIGH-QUALITY LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review

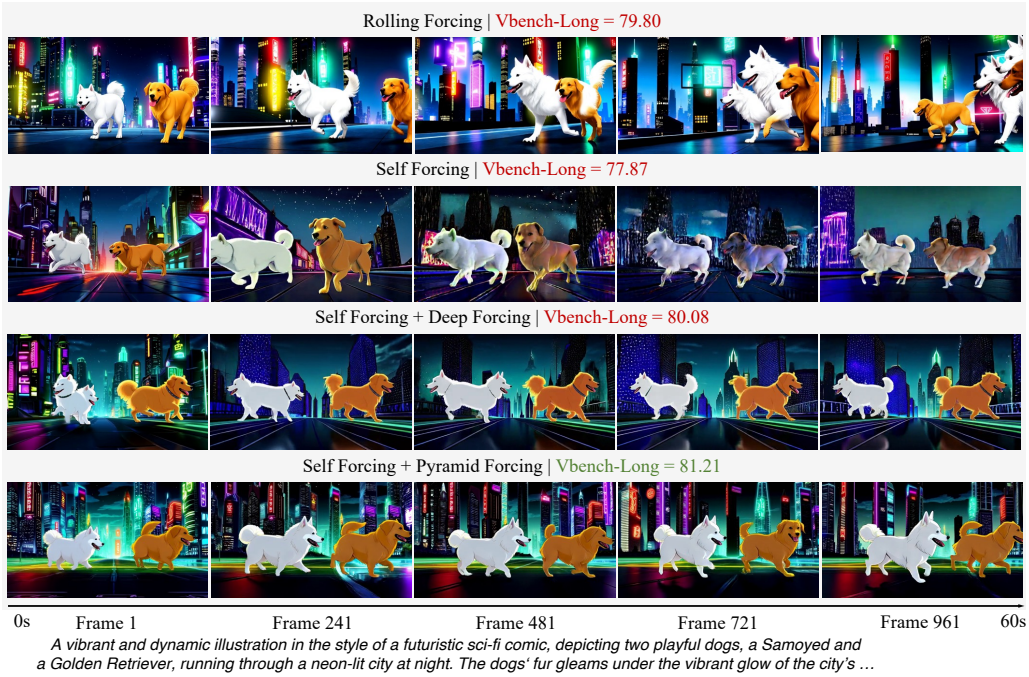


Figure 1: Pyramid Forcing mitigates appearance drift in the illustrated long-video generation, reaching 81.21 VBench-Long on 60-second Self Forcing, the highest score among evaluated baselines.

ABSTRACT

Autoregressive video generation supports streaming synthesis but remains vulnerable to long-horizon degradation. Shared KVCache policies overlook differences in historical-frame attention across heads. We identify three empirical logit patterns: Anchor Heads exhibit broadly positive historical logits, Wave Heads exhibit temporal oscillations, and Veil Heads exhibit lower intermediate-history logits. These patterns motivate **Pyramid Forcing**, a training-free framework that classifies heads offline, assigns differentiated cache strategies and capacities, and executes unequal head-level cache lengths through Ragged-Cache Attention. On Self Forcing and Causal Forcing, Pyramid Forcing improves VBench-Long Total Score, increasing the 60-second Self Forcing score from 77.87 to 81.21. Separate uniform-policy and assignment controls examine the incremental value of head-specific allocation. Causal Forcing exhibits stronger motion alongside declines in several consistency and fidelity metrics, indicating that the quality balance depends on the backbone and cache configuration.

1 INTRODUCTION

Autoregressive (AR) video generation models (Huang et al., 2025a; Teng et al., 2025; Zhu et al., 2026) provide a natural pathway for long video generation by iteratively denoising future frames conditioned

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

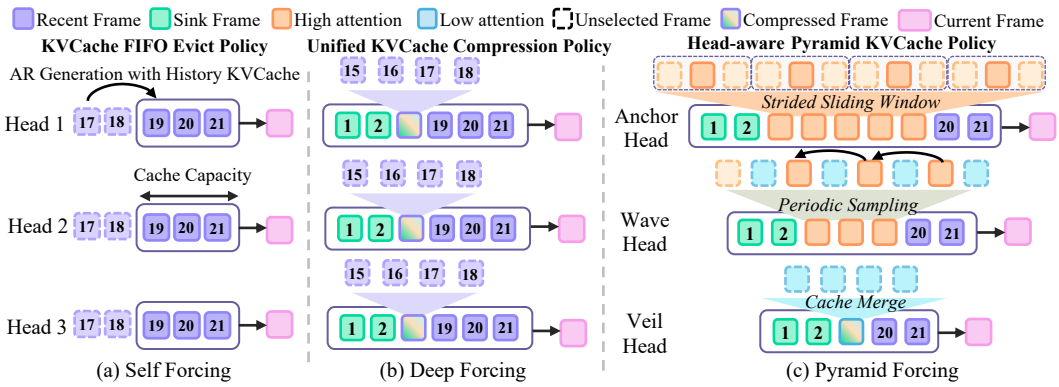


Figure 2: Comparison of KVCache policies. Self Forcing and Deep Forcing use shared retention rules across heads. Pyramid Forcing assigns strategies and capacities to Anchor, Wave, and Veil Heads according to historical-frame logit patterns.

on previously generated frames. Unlike diffusion-based video generation models (Peebles & Xie, 2023; Wan et al., 2025; Yang et al., 2024b) constrained by a fixed temporal scope, AR video generation supports streaming generation, continuation and open-ended generation, making it promising for applications such as interactive generation(He et al., 2025; Mao et al., 2025), world simulation(Yuan et al., 2026; Xie et al., 2026), and embodied intelligence(Feng et al., 2025; Ye et al., 2026). However, in practical scenarios, long video generation still suffers from long-term degradation. As illustrated in Fig. 1, as the generation process progresses, visual appearance tends to drift, motion dynamics weaken, and object identity gradually becomes blurred.

Existing works (Cui et al., 2025) have suggested that error accumulation and historical-context reuse affect quality during autoregressive generation, especially when the retained history includes artifacts from earlier predictions. To address this issue, prior methods attempt to mitigate error accumulation by retaining recent frames(Yang et al., 2025; Liu et al., 2025), preserving early target frames(Li et al., 2026; Kim et al., 2026), or selecting important historical information(Yi et al., 2025). These policies determine which history remains available, but commonly apply the same retention rule across heads. Figure 2 contrasts such shared rules with head-specific allocation. This motivates examining whether preferred historical coverage and cache representation differ across heads within a model.

We analyze pre-softmax historical-frame logits in Self Forcing and identify three empirical patterns. **Anchor Heads** exhibit broadly positive logits over historical frames, motivating wider temporal coverage. **Wave Heads** exhibit oscillatory logit sequences, motivating periodic retention. **Veil Heads** exhibit higher logits for initial and adjacent frames and lower logits over intermediate history, motivating compact summaries. These names denote groups defined by the classification procedure. Their usefulness for cache allocation is assessed through generation experiments, since raw logit signs alone do not measure attention mass or establish causal head functions.

Two questions follow: how stable are these assignments, and does matching policies to them improve generation over strong uniform strategies?

Based on this observation, we propose Pyramid Forcing, a head-aware pyramidal KVCache framework for long video generation, as shown in Fig. 2(c). Specifically, we first introduce **Offline Tri-Pattern Head Classification**, which uses sign-rate statistics and frequency-domain periodicity to identify Anchor, Wave, and Veil Heads from a small calibration set. Then, we introduce head-specific **Pyramid KVCache Policies** tailored to distinct historical-frame attention patterns, using strided retention for Anchor Heads, periodic sampling for Wave Heads, and spatiotemporal mean merging for Veil Heads to control historical coverage and representation within the attention-read budget. We further introduce **Heterogeneous KVCache Computation Optimization**, packing variable head-level cache lengths for efficient execution with existing variable-length attention kernels.

We evaluate 30s and 60s generation on Self Forcing and Causal Forcing using VBench-Long. Total Score improves on both backbones, reaching 81.21 from 77.87 on 60s Self Forcing. Separate controls test strategy and capacity assignment against uniform and shuffled policies. On Causal Forcing, Dynamic Degree improves by up to 29.36 points while some consistency and fidelity metrics decline, distinguishing aggregate gains from the backbone-dependent balance between motion and quality.

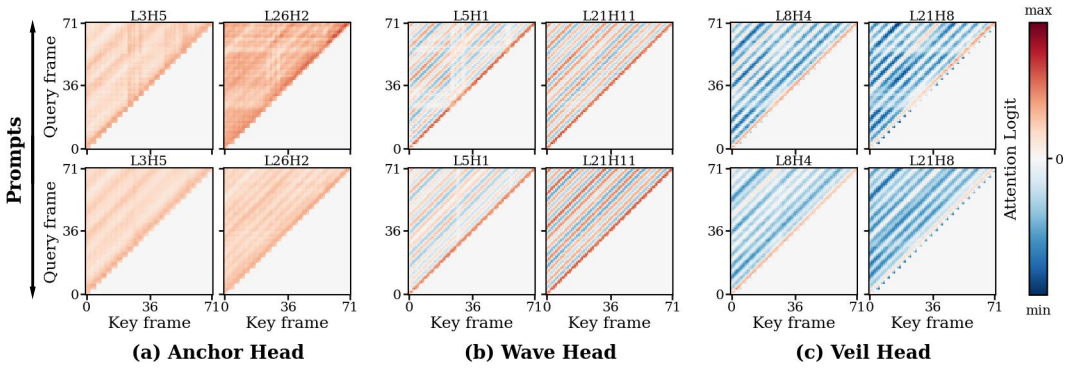


Figure 3: Historical-frame attention patterns of three head types in Self Forcing. *Anchor Heads* exhibit broadly positive historical-frame logits, *Wave Heads* exhibit temporal oscillations, and *Veil Heads* exhibit lower intermediate-history logits.

2 PRELIMINARIES

2.1 AUTOREGRESSIVE VIDEO DIFFUSION

Autoregressive video diffusion conditions each new latent frame on generated history. Cache indices count latent frames, distinct from decoded frames and denoising iterations; generation blocks may contain multiple latent frames. For N latent frames, the joint distribution factorizes as $p(x^{1:N}) = \prod_{t=1}^{N} p(x^t \mid x^{<t})$. Self-attention uses clean historical context $x^{<t} = (x^1, \ldots, x^{t-1})$ to predict x^t, with current-block attention governed by the backbone mask.

The query of the current frame, $Q \in \mathbb{R}^{B\times H\times L_q\times D}$, attends to the keys and values of historical frames, $K, V \in \mathbb{R}^{B\times H\times L_{kv}\times D}$. The attention operation is computed as

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{D}}\right) V. \tag{1}$$

Here, B is batch size, H is the number of heads, and $D = C/H$ is the head dimension. L_q and L_{kv} denote the sequence lengths of the query and key/value tokens, respectively. The resulting attention logits typically have shape $QK^\top \in \mathbb{R}^{B\times H\times L_q\times L_{kv}}$.

2.2 KVCACHE POLICY IN AUTOREGRESSIVE VIDEO DIFFUSION

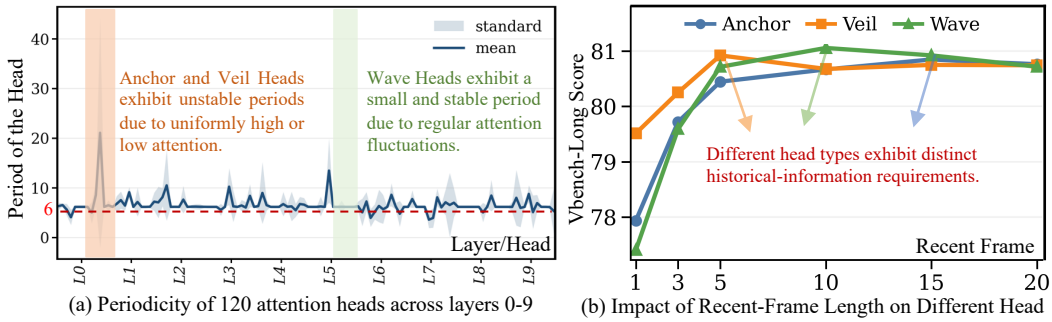
Historical latent tokens supply reusable keys and values. A cache policy specifies the resident state and the entries read by attention; a limited read set need not imply bounded storage. For uncompressed frame selection at latent index t, a subset $\mathcal{I}(t) \subset \mathcal{H}(t)$ is selected from the candidate history, typically consisting of sink frames $\mathcal{S}(t)$ for global anchoring and recent frames $\mathcal{R}(t)$ for local temporal context. Their KVCaches are concatenated as $K' = [K_{\text{sink}} \mid K_{\text{recent}}], \quad V' = [V_{\text{sink}} \mid V_{\text{recent}}]$, and attention is computed by

$$\mathrm{Attn}(Q_t, K'_t, V'_t) = \mathrm{softmax}\left(\frac{Q_t[K_{\text{sink}}^\top, K_{\text{recent}}^\top]}{\sqrt{D}}\right)[V_{\text{sink}}, V_{\text{recent}}]. \tag{2}$$

Clean keys and values enter history once after each block's clean-context pass. Intermediate denoising states are not separate historical frames. Shared retention rules can be replaced by head-specific read sets and compressed summaries; the latter represent groups of tokens rather than subsets of uncompressed frames.

3 MOTIVATION

We analyze historical-frame attention in Self Forcing to guide cache allocation, focusing on heterogeneity, periodicity, and the amount of retained history.



(a) Periodicity of 120 attention heads across layers 0-9 (b) Impact of Recent-Frame Length on Different Head

Figure 4: Analysis of head-wise logit periodicity and window sensitivity. (a) Wave spectral estimates concentrate near six latent-frame indices under the recorded FFT procedure. (b) Generation quality responds differently to retained history across head groups under the tested cache configuration.

Observation 1: Attention heads exhibit heterogeneous historical-frame logit patterns. Based on the pre-softmax attention logits and their corresponding historical-frame attention patterns, we categorize attention heads into three types: *Anchor Heads*, *Wave Heads*, and *Veil Heads*, as illustrated in Fig. 3.

Anchor Heads. As shown in Fig. 3(a), Anchor Heads exhibit broadly positive historical-frame logits in the selected examples. This pattern motivates retaining frames across a wider temporal range under a limited attention-read budget. The resulting policy is assessed by generation quality, since positive logit values alone do not establish historical-frame importance or a causal anchoring function.

Wave Heads. As shown in Fig. 3(b), Wave Heads exhibit regular positive–negative alternations along the temporal dimension, forming wave-like logit maps. This pattern motivates periodic retention of intermediate historical frames. Whether the selected frames retain useful information also depends on normalized attention and the sampling phase.

Veil Heads. As shown in Fig. 3(c), Veil Heads exhibit higher logits for the first and adjacent frames and negative logits for much of the intermediate history. Together with the window ablation below, this pattern motivates a compact cache policy. Negative logits alone do not establish that intermediate history is actively suppressed.

Moreover, the selected visualizations show recurring attention patterns across different prompts. This observation motivates offline head classification, while its stability across layers, query positions, and denoising steps requires separate evaluation. Additional visualizations and diagnostic definitions are provided in the supplementary material.

Observation 2: Wave Heads exhibit oscillatory logits over historical-frame positions. To characterize the temporal fluctuations of Wave Heads, we apply FFT to historical-frame logit sequences in Self Forcing. As shown in Fig. 4(a), their estimated periods concentrate near six latent-frame indices. This motivates retaining historical frames at intervals informed by the observed oscillations. A spectral period alone does not determine the phase of high-attention frames or prove that the corresponding cache policy is optimal. See Appendix D.5 for a discussion of RoPE and the scope of this interpretation.

Observation 3: Head groups respond differently to retained historical context. Figure 4(b) varies the recent window $\mathcal{R}(t)$ for each head group in 30-second generation. Anchor Heads improve with longer windows in this setting. Wave Heads perform best with 10 retained frames, while Veil Heads peak at 5 and decline as the window expands. These group-wise interventions change both retained context and cache size. They provide evidence for differentiated allocations within the tested configuration; the separate uniform-policy and shuffled-assignment controls examine whether matching strategies and capacities to the groups improves generation quality.

4 Method

4.1 Offline Tri-Pattern Head Classification

We classify heads offline using historical-frame logits from a small calibration set (Fig. 5(a)). The classification is an empirical rule for cache allocation: Anchor Heads exhibit broadly positive logits,

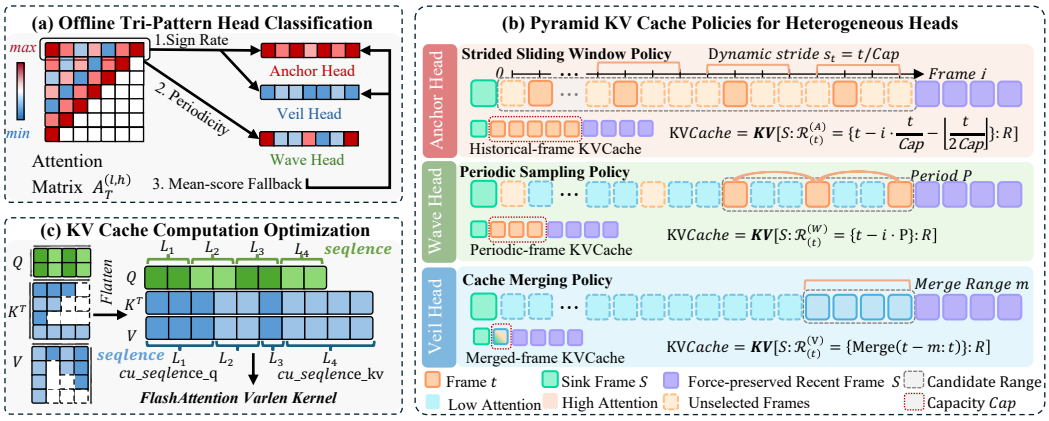


Figure 5: Overview of Pyramid Forcing. (a) Offline Tri-Pattern Head Classification identifies Anchor, Wave, and Veil Heads. (b) Pyramid KVCache Policies assign head-specific cache strategies. (c) KVCache Computation Optimization enables efficient ragged-cache attention with optimized kernels.

Wave Heads exhibit temporal oscillations, and Veil Heads exhibit predominantly low intermediate-history logits. We combine sign-rate statistics with FFT-based periodicity and keep the resulting assignments fixed during inference. Calibration is model-specific and requires no weight updates. Raw signs are not shift-invariant importance measures: adding a constant to a query row changes signs without changing softmax weights (Appendix D.3).

For frame T, head h, and layer l, we denote the frame-averaged pre-softmax logit matrix as $A_T^{(l,h)} \in \mathbb{R}^{T\times T}$ and extract the historical attention sequence as $a^{(l,h)}(t) = A_T^{(l,h)}[T,t]$, $t = 1,\ldots,T-1$. Each entry averages logits over query and key tokens in a pair of latent frames, excluding masked pairs. The sequence includes clean historical frames and excludes the current noisy block. We then compute the sign rates of sequence $a^{(l,h)}(t)$ as

$$r_{\text{pos}} = \frac{1}{T-1}\sum_{t=1}^{T-1} \mathbf{1}\{a^{(l,h)}(t) > 0\}, \quad r_{\text{neg}} = 1 - r_{\text{pos}}. \tag{3}$$

With a fixed empirical threshold α, heads with $r_{\text{pos}} \geq \alpha$ are Anchor Heads; those with $r_{\text{neg}} \geq \alpha$ are Veil Heads. Here, r_{neg} includes zero logits. Applying the sign-rate tests first separates strongly one-sided sequences before examining oscillation. Heads near either threshold can change labels across samples, so classification stability is evaluated separately from downstream generation quality.

For the remaining heads, we detect periodicity in the frequency domain. After preprocessing $a^{(l,h)}(t)$, we apply FFT and estimate the dominant period $\widehat{P} = \mathrm{Period}(\mathrm{FFT}(a^{(l,h)}(t)))$. Since the estimated periods of Wave Heads concentrate within a small range, a head is classified as Wave when a valid estimate satisfies $\widehat{P} < \beta$. This real-valued estimate and threshold differ from the integer inference interval P (Appendix D.4).

For any still-unclassified heads, we use a mean-score fallback: $\bar{a}^{(l,h)} = \frac{1}{T-1}\sum_{t=1}^{T-1} a^{(l,h)}(t)$. If $\bar{a}^{(l,h)} > 0$, the head is assigned to Anchor Heads; otherwise, it is assigned to Veil Heads. Finally, we obtain the layer-wise head sets $\mathcal{H}_A$, $\mathcal{H}_W$, and $\mathcal{H}_V$, which are mutually exclusive and jointly cover all attention heads. For notational simplicity, we use superscripts (A), (W), and (V) to denote variables associated with the three head types.

4.2 Pyramid KVCache Policies for Heterogeneous Heads

Pyramid KVCache Policies map the three empirical patterns to distinct retention rules (Fig. 5(b)): broad coverage for Anchor Heads, periodic sampling for Wave Heads, and compact summaries for Veil Heads. The mapping remains fixed during inference. Classification selects a strategy, and capacity limits its attention-read representation. All main and supplementary Pyramid Forcing experiments use adaptive-center Anchor, with parameter changes specified in the corresponding ablations (Appendix E.8).

Anchor Heads KVCache Policy. For broad historical coverage under a limited attention-read budget, we use an Adaptive Strided Sliding Window. Its stride grows with history length, distributing selected frames across the generated sequence.

Specifically, using zero-based latent-frame indices, we set the dynamic stride as $s_t = t/\mathrm{Cap}$ and select integer centers across the available history:

$$\mathcal{R}_{(t)}^{(A)} = \left\{ \left\lfloor \left(i + \frac{1}{2}\right) \frac{t}{\mathrm{Cap}} \right\rfloor \mid i = 0, 1, \ldots, \mathrm{Cap} - 1 \right\}. \tag{4}$$

The corresponding KVCache is then $\mathbf{K}_{\mathrm{recent}}^{(A)} = \mathbf{K}\big[\mathcal{R}_{(t)}^{(A)}, :\big]$, $\mathbf{V}_{\mathrm{recent}}^{(A)} = \mathbf{V}\big[\mathcal{R}_{(t)}^{(A)}, :\big]$. We filter invalid indices, remove duplicates and sink/recent overlap, and retain the remaining indices in chronological order. This selector limits attention reads but requires historical access for newly selected frames. It does not bound resident storage. Appendix E.2 details adaptive-center selection, online updates, and historical-access requirements.

Wave Heads KVCache Policy. Wave logits oscillate over relative latent-frame positions. We use Periodic Sampling with integer interval P, guided by offline attention analysis (Sec. 3) rather than object-motion frequency. When generating frame t, the policy selects up to Cap clean historical frames under a limited attention-read budget:

$$\mathcal{R}_{(t)}^{(W)} = \{t - i \cdot P \mid i = 1, \ldots, \mathrm{Cap}, \quad t - iP \geq 0\}. \tag{5}$$

The corresponding KVCache is then $\mathbf{K}_{\mathrm{recent}}^{(W)} = \mathbf{K}\big[\mathcal{R}_{(t)}^{(W)}, :\big]$, $\mathbf{V}_{\mathrm{recent}}^{(W)} = \mathbf{V}\big[\mathcal{R}_{(t)}^{(W)}, :\big]$. Sink/recent overlap is removed before readout. The current noisy block is handled separately under the backbone's attention mask. Period and phase are distinct choices; Appendix D.5 discusses their interpretation, with sampling ablations and RoPE diagnostics in Tables 9 and 18. The phase-bucket implementation reads at most C_W middle frames but can retain PC_W frames across buckets. The phase-shifted control tests a different selection at fixed spacing.

Veil Heads KVCache Policy. Veil Heads favor compact caches in the window ablation. We average keys over a block $\mathcal{B}$ of m latent frames and a spatial patch $\mathcal{P}$ of at most s^2 tokens: $\bar{\mathbf{K}}_{\mathcal{B},\mathcal{P}} = \frac{1}{m|\mathcal{P}|} \sum_{\tau \in \mathcal{B}} \sum_{u \in \mathcal{P}} \mathbf{K}[\tau, u, :]$, and apply the same operation to $\mathbf{V}$. Merged tokens retain all channels. Averaging preserves channel coordinates but changes logits and softmax normalization. It is lossy, with quality assessed against unmerged entries and temporal means at a matched read budget. Capacity C_V counts merged temporal blocks, separately from m and s. At a matched read budget, spatiotemporal means exceed unmerged entries by 0.27 points (Table 10). Appendix E.4 specifies tensor shapes and positions.

By default, all heads retain $S = 3$ sink frames and $R = 4$ forced recent frames without merging. We apply Dynamic RoPE to sink frames following prior works (Yi et al., 2025; Yesiltepe et al., 2025). Wave Heads use the same sink setting. Default Veil parameters are $C_V = 2$ and $m = 2$; ablations specify their changes separately.

4.3 Heterogeneous KVCache Computation Optimization

Ragged-Cache Attention. Head-specific policies assign different KV lengths to batch–head pairs, whereas dense attention expects $K, V \in \mathbb{R}^{B \times H \times L \times D}$ with shared length L. Padding would introduce redundant storage and computation. We instead pack the selected K/V entries into contiguous sequences in batch–head order (Fig. 5(c)). For $N = BH$ sequences of lengths $L_0, \ldots, L_{N-1}$, define $\mathrm{cu_seqlens}_0 = 0$ and $\mathrm{cu_seqlens}_i = \sum_{j=0}^{i-1} L_j$ for $1 \leq i \leq N$. Sequence i occupies $[\mathrm{cu_seqlens}_i, \mathrm{cu_seqlens}_{i+1})$ for $0 \leq i < N$. Variable-length FlashAttention (Dao et al., 2022) uses the packed queries, K/V, and their respective boundary arrays to compute attention without padding.

Efficient kernel customization. Separate per-frame FlashAttention (Dao et al., 2022) calls incur launch overhead. We fuse three per-layer variable-length attention calls into one after permuting queries to match the KV layout, and replace per-head slice writes with batched workspace packing. This reduces attention launches from three to one and GPU scheduling calls from about 60 to 6. Fusion must preserve each query's selected keys, positional mapping, and mask. Packing remains an execution cost, separate from attention computation.

Table 1: Quantitative comparison on long video generation. We evaluate Pyramid Forcing against autoregressive video diffusion generation baselines for 30- and 60-second videos using VBench-Long.

Model	Dynamic Degree ↑	Motion Smoothness ↑	Overall Consistency ↑	Imaging Quality ↑	Aesthetic Quality ↑	Quality Score ↑	Semantic Score ↑	Total Score ↑
				30 seconds				
CausVid	45.70	98.25	22.39	66.11	59.63	86.66	50.93	79.51
Rolling Forcing	30.91	98.72	24.85	71.43	61.69	86.18	55.20	79.98
LongLive	41.70	98.79	24.54	68.79	61.36	86.97	54.31	80.44
Self Forcing	44.34	98.52	24.66	70.66	63.22	87.14	54.31	80.57
+ Deep Forcing	46.48	98.43	25.10	71.70	63.77	87.23	54.00	80.59
+ Pyramid Forcing	**55.07**	**98.82**	**25.26**	**72.17**	**66.55**	**88.93**	**55.52**	**82.25**
Causal Forcing	61.38	**98.48**	23.03	**69.66**	58.22	86.45	53.12	79.78
+ Pyramid Forcing	**82.67**	97.23	**23.21**	68.64	**60.03**	**86.98**	**54.07**	**80.40**
				60 seconds				
CausVid	43.67	98.28	21.65	65.54	59.34	86.58	49.87	79.23
Rolling Forcing	32.08	98.73	24.20	70.55	61.06	86.19	54.20	79.80
LongLive	40.93	98.77	24.54	68.83	61.70	86.89	54.79	80.47
Self Forcing	43.75	97.85	22.33	64.28	55.62	84.84	50.01	77.87
+ Deep Forcing	43.10	98.35	24.48	68.43	60.48	86.37	54.91	80.08
+ Pyramid Forcing	**53.68**	**98.53**	**24.70**	**70.03**	**62.00**	**87.58**	**55.71**	**81.21**
Causal Forcing	57.03	**98.49**	**22.51**	**68.63**	56.87	85.84	52.36	79.14
+ Pyramid Forcing	**86.39**	97.03	22.41	68.38	**58.91**	**86.64**	**53.05**	**79.92**

Table 2: Ablation study on the key components of Pyramid Forcing across six variants.

Method	Dynamic Degree ↑	Image Quality ↑	VBench-Long ↑	Lat. ↓	Mem. ↓
Self Forcing	37.89	71.51	80.44	54.12s	35.80GB
+ Dynamic ROPE	41.79	70.74	80.96	54.17s	35.80GB
+ Ragged-Cache	41.01	71.08	80.43	**52.05s**	35.80GB
+ Head	39.84	71.16	80.99	54.03s	**34.87GB**
+ Head & Pyramid	45.89	**72.06**	81.58	139.70s	36.06GB
Pyramid Forcing	**50.58**	71.73	**82.16**	52.31s	36.32GB

Table 3: Efficiency and peak GPU memory usage. FPS denotes generated frames per second.

Method	VBench-Long ↑	Latency ↓	FPS ↑	Memory ↓
Self Forcing	80.57	54.01s	11.85	35.80GB
+ Deep Forcing	80.59	52.85s	12.19	37.17GB
+ Pyramid Forcing	82.25	52.18s	11.98	36.32GB
Causal Forcing	79.78	46.75s	13.31	47.93GB
+ Pyramid Forcing	80.40	45.21s	13.78	45.59GB

5 Experiments

5.1 Experimental Setup

Evaluation. Following Self Forcing (Huang et al., 2025a), we evaluate 30- and 60-second continuous generation on 128 MovieGen (Polyak et al., 2024) prompts using VBench-Long (Huang et al., 2024; 2025b). Baselines include CausVid (Yin et al., 2025), Rolling Forcing (Liu et al., 2025), LongLive (Yang et al., 2025), and Deep Forcing (Yi et al., 2025). Additional metrics appear in Appendix Table 7. Cache-policy comparisons use their respective within-backbone controls; comparisons across distilled models also reflect differences in training. We study continuous scenes. Calibration, analysis, and evaluation serve different roles, but prompt overlap is not established (Appendix A.1). Supplementary comparisons retain their own configurations and baselines.

Implementation Details. We implement Pyramid Forcing on Self Forcing (Huang et al., 2025a) and Causal Forcing (Zhu et al., 2026). All main and supplementary Pyramid Forcing experiments use adaptive-center Anchor. Defaults are 3 sink frames for every head type, 4 recent frames, $(C_A, C_W, C_V) = (4, 4, 2)$, and Veil merge range $m = 2$; C_V counts merged temporal blocks. Offline classification uses 32 VBench-Long prompts, 15-second videos, and sign-rate/period thresholds of $80\%/6.4$. Kernels use FlashInfer/FlashAttention (Ye et al., 2025; Dao et al., 2022) on NVIDIA H200 GPUs. Parameter and component changes are specified separately for each ablation (Appendix E.8).



Figure 6: Selected 30- and 60-second generations illustrating appearance preservation and motion under Pyramid Forcing and baseline methods.

5.2 Main Results

Quantitative Results. In Tab. 1, Pyramid Forcing achieves the highest Total Scores, improving Self Forcing from 80.57 to 82.25 at 30 seconds and from 77.87 to 81.21 at 60 seconds. Causal Forcing improves from 79.78 to 80.40 and from 79.14 to 79.92, respectively. Its Dynamic Degree rises from 61.38 to 82.67 and from 57.03 to 86.39, while Motion Smoothness, Imaging Quality, and Subject Consistency decline. At 60 seconds, Subject Consistency falls from 96.57 to 94.78, exposing a motion–consistency tradeoff. Independent Causal Forcing calibration changes Imaging Quality by +1.47 and Motion Smoothness by -0.28 (Appendix Table 17). Separately, four seeds on 32 MovieGen prompts with 30-second generation yield a mean paired Total Score gain of 1.57 ± 0.14 (sample standard deviation), measuring seed variation rather than prompt-level uncertainty. Additional 32-prompt evaluations show gains of 1.57 points on ChronoMagic-Bench and 1.33 on Step-Video-T2V-Eval. The separate 60-second comparison exceeds All-Merge by 0.51 points (Appendix Table 17).

Qualitative Results. Figure 6 illustrates Self Forcing's appearance drift and noise. Pyramid Forcing preserves subject appearance and scene details while maintaining motion in these selected examples.

5.3 Ablation Studies

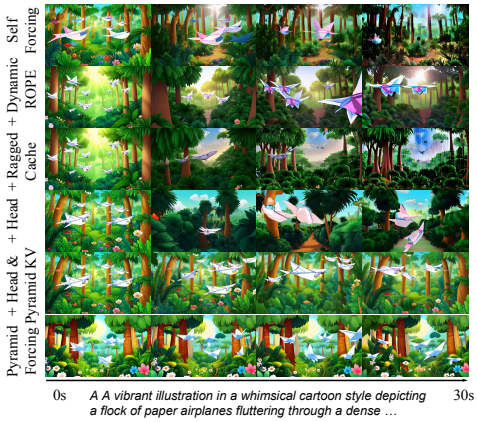


Figure 7: Qualitative ablation of components.

Ablation Study on Key Components. Table 2 and Fig. 7 compare Self Forcing, variants using only Dynamic RoPE, Ragged-Cache Attention, or Head Classification, Head Classification with Pyramid KVCache policies, and the full method. The Head Classification variant uses windows $\mathcal{R}^{(A)} = 12$, $\mathcal{R}^{(W)} = 10$, and $\mathcal{R}^{(V)} = 5$. This variant evaluates group-specific windows before introducing the differentiated cache strategies. Dynamic RoPE improves motion; Ragged-Cache Attention reduces latency with a 0.01-point Total Score change. Classification and cache policies improve quality, while the full method combines the highest Total Score with low latency. In separate controls (Appendix Table 8), AWV exceeds All-Merge by 0.42 points. Shuffling strategies with capacities fixed reduces Total Score by 0.38; shuffling capacities with strategies fixed reduces it by 0.28. These controls support both assignments; prompt-level uncertainty and resident-memory matching remain unresolved.

Efficiency and Peak GPU Memory. Table 3 reports lower latency: Self Forcing decreases from 54.01 to 52.18 seconds, and Causal Forcing from 46.75 to 45.21 seconds. Attention-read payload, resident cache, and measured peak memory are distinct quantities (Appendix E.6); timing conventions are discussed in Appendix E.7. Peak memory increases from 35.80 to 36.32 GB on Self Forcing and decreases from 47.93 to 45.59 GB on Causal Forcing, so memory savings depend on the backbone.

Table 4: Effect of prompt number and video duration on offline tri-pattern head classification.

Prompts	Duration	VBench-Long ↑	Overhead ↓	Peak Mem. ↓
1	60s	81.83	507.32 s	80.14 GB
1	30s	81.81	249.58 s	62.20 GB
1	15s	81.83	103.26 s	38.81 GB
1	10s	81.60	70.10 s	32.75 GB
1	5s	81.51	33.95 s	26.11 GB
128	15s	81.83	3.67 h	38.81 GB
64	15s	81.83	1.83 h	38.81 GB
32	15s	81.83	0.91 h	38.79 GB
16	15s	81.82	0.46 h	38.80 GB

Table 5: Effect of head-specific policy parameters in Pyramid KVCache policies.

Head Type	Setup	Dynamic Degree ↑	Image Quality ↑	VBench-Long ↑
Self Forcing	-	37.89	71.51	80.44
Pyramid Forcing	-	50.58	71.73	82.16
Anchor Head	$s_t = 3$	50.98	72.24	81.92
	$s_t = 6$	51.76	72.10	82.05
	$s_t = 12$	53.32	72.80	82.08
Wave Head	$P = 3$	48.05	72.40	81.95
	$P = 6$	53.32	72.49	82.06
	$P = 12$	50.78	72.01	81.96
Veil Head	$m = 2$	50.39	72.23	82.00
	$m = 3$	48.63	72.23	81.94

Calibration Cost and Assignment Stability. In Tab. 4, 32 prompts with 15-second videos match larger sets in Total Score, costing 0.91 hours and 38.79 GB. Matching scores do not imply identical labels. Separate assignment comparisons yield 99.17–99.72% agreement across calibration subsets, at least 98.89% across 15–60-second generation, and 89.95% on average across prompt–seed samples. Agreement across FETV, Step-Video-T2V-Eval, and ChronoMagic-Bench is 91.67–94.44%. Relative to Self Forcing, agreement is 91.94% on Causal Forcing and 87.22% on Causal-rCM. These comparisons measure label agreement (Appendix D.6). Calibration cost is amortized across generations with the same configuration.

Head-Specific Policy Parameters. All settings in Tab. 5 exceed Self Forcing. Appendix Tables 11–16 distinguish sink/recent sizes, middle-cache capacities, and classification thresholds. Total Score varies by 0.61 points in the sign-rate scan and 0.49 in the period-threshold scan, although Dynamic Degree is more sensitive. Increasing the period threshold from 6.4 to 6.5 reduces Dynamic Degree from 50.59 to 46.48, despite a smaller Total Score change. These scans establish local parameter sensitivity.

6 Related Work

Autoregressive Video Diffusion. Diffusion-based video generators achieve strong visual quality through large-scale training (Peebles & Xie, 2023; Wan et al., 2025; Yang et al., 2024b), but full-sequence denoising with bidirectional attention limits them to fixed temporal windows. Recent autoregressive video diffusion methods extend generation by conditioning future clips on synthesized history: CausVid (Yin et al., 2025) distills causal generators from bidirectional DiTs, Self Forcing (Huang et al., 2025a) reduces train-test mismatch via self-generated conditioning, and Causal Forcing (Zhu et al., 2026) improves distillation with ODE initialization. Self Forcing++ (Cui et al., 2025), Rolling Forcing (Liu et al., 2025), and LongLive (Yang et al., 2025) further improve long-horizon stability through error correction or training-inference alignment. Nevertheless, long video generation still suffers from appearance drift, motion decay, and identity inconsistency.

KVCache Policy. KVCache optimization is crucial for improving the quality and efficiency of autoregressive generation. Existing LLM studies mainly focus on cache selection (Li et al., 2024; Zhang et al., 2023; Xiao et al., 2024; Lee et al., 2024), budget allocation (Cai et al., 2024; Feng et al., 2024; Yang et al., 2024a), and cache compression (Cheng et al., 2025; Kim et al., 2023; Liu et al., 2024b;a; Dong et al., 2024). Our focus is joint strategy and capacity assignment for video attention heads, where latent-frame grouping, spatial tokens, repeated denoising, and positional handling affect cache design. Uniform-policy and shuffled-assignment controls evaluate this allocation principle, while Dynamic RoPE and variable-length attention remain supporting components adapted from prior work.

7 Conclusion

Pyramid Forcing combines empirical head classification, differentiated cache policies, and Ragged-Cache Attention for training-free autoregressive video generation. Experiments improve aggregate quality at comparable inference cost, with backbone-dependent tradeoffs between motion and visual consistency.

AI USE STATEMENT

We used ChatGPT to assist with translation, manuscript drafting and editing, literature search, clarification of methodological descriptions, feedback on ablation designs, and the interpretation and presentation of experimental results. The authors are responsible for verifying all AI-assisted material and take full responsibility for the final text, methods, experimental results, references, and conclusions.

REFERENCES

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Wen Cheng, Shichen Dong, Jiayu Qin, and Wei Wang. Qaq: Quality adaptive quantization for llm kv cache. In *ICCV*, pp. 2542–2550, 2025.

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. *ICLR*, 2025.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35:16344–16359, 2022.

Harry Dong, Xinyu Yang, Zhenyu Zhang, Zhangyang Wang, Yuejie Chi, and Beidi Chen. Get more with less: Synthesizing recurrence with kv cache compression for efficient llm inference. *ICML*, 2024.

Yao Feng, Chendong Xiang, Xinyi Mao, Hengkai Tan, Zuyue Zhang, Shuhe Huang, Kaiwen Zheng, Haitian Liu, Hang Su, and Jun Zhu. Vidarc: Embodied video diffusion model for closed-loop control. *arXiv preprint arXiv:2512.17661*, 2025.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *arXiv preprint arXiv:2407.11550*, 2024.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, pp. 21807–21818, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. Compressed context memory for online language model interaction. *ICLR*, 2023.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter A Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. *arXiv preprint arXiv:2603.12513*, 2026.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. {InfiniGen}: Efficient generative inference of large language models with dynamic {KV} cache management. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 155–172, 2024.

Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *NeurIPS*, 37: 22947–22970, 2024.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *NeurIPS*, 37:139997–140031, 2024a.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *ICLR*, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *ICML*, 2024b.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, 2023.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ICLR*, 2024.

Linxi Xie, Lisong C Sun, Ashley Neall, Tong Wu, Shengqu Cai, and Gordon Wetzstein. Generated reality: Human-centric world simulation using interactive video generation with hand and camera control. *arXiv preprint arXiv:2602.18422*, 2026.

Dongjie Yang, XiaoDong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. Pyramidinfer: Pyramid kv cache compression for high-throughput llm inference. In *ACL*, pp. 3258–3270, 2024a.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *ICLR*, 2025.

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*, 2024b.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, et al. Flashinfer: Efficient and customizable attention engine for llm inference serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. *arXiv preprint arXiv:2511.20649*, 2025.

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.

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 *CVPR*, pp. 22963–22974, 2025.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

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.

A Experimental Setup and Additional Metrics

Sections A–C cover evaluation, controlled ablations, and generalization. Section D details classification and attention analysis; Section E defines cache representations and updates. Sections F and G provide qualitative comparisons and limitations.

A.1 Evaluation Settings and Experiment Groups

The original main evaluation uses 128 MovieGen prompts for 30-second and 60-second video generation and evaluates the results with VBench-Long (Huang et al., 2024; 2025b). The default configuration uses $S = 3$ sink frames for every head type, $R = 4$ forced recent frames, and middle-cache capacities $(C_A, C_W, C_V) = (4, 4, 2)$, with Veil merge range $m = 2$. Offline classification uses 32 VBench-Long prompts with 15-second videos, a sign-rate threshold of $\alpha = 0.8$, and a period threshold of $\beta = 6.4$.

The 32-prompt calibration setting and the 256-prompt attention analysis serve different purposes. The latter supports the visualization of head labels and spectral estimates; it does not establish which label file was used by each generation experiment. The available prompt records do not establish disjoint calibration, analysis, tuning, and evaluation sets. Experiment-specific configurations and label files are discussed in Section E.8.

Table 6 summarizes the supplementary comparisons. Their Self Forcing video-quality scores use 30-second generation, except for the explicitly identified 60-second experiment. The independently calibrated Causal Forcing comparison is reported separately. We use AWV to denote the full Anchor–Wave–Veil policy, and express quality differences in score points. Spectral peak periods are measured in latent-frame indices.

All comparisons use the controls from their own experiment group. The Total Scores of 82.25 in the main experiment, 82.16 in the original component ablation, and 81.83 in the calibration-size ablation are not repeated estimates under a verified identical configuration. Likewise, the All-Merge scores of 81.62 in Table 8 and 81.67 in Table 16 belong to separate comparisons.

Table 6: Overview of supplementary comparisons. Self Forcing quality evaluations use 30-second generation except for S6. Each comparison uses its own baseline.

ID	Comparison	Result location
S1	Strong uniform cache policies	Table 8; AWV: 82.04
S2	Within-layer routing and capacity assignment	Table 8; full method: 82.09
S3	Wave period and phase	Table 9; $P = 6$: 82.02
S4	Veil merging at a matched read budget	Table 10; spatiotemporal mean: 82.06
S5	Sign-rate thresholds with All-Merge	Table 16; $\alpha = 0.8$: 82.05
S6	60-second Self Forcing generation	Table 17; +0.51 over All-Merge
S7	Independently calibrated Causal Forcing	Table 17; +0.62 over native CF
S8	FFT preprocessing and RoPE scaling	Table 18; spectral peak periods

A.2 Additional Evaluation Metrics of the Main Experiment

Table 7 reports the remaining VBench-Long metrics for the main comparison. Improvements in Total Score do not imply improvements in every dimension. For example, under Causal Forcing, Pyramid Forcing improves Dynamic Degree but reduces Subject Consistency and Motion Smoothness at both generation lengths. We discuss these tradeoffs in Section G.

Temporal Flickering is the VBench score $1 - \mathrm{MAE}/255$, displayed on a percentage scale; higher values indicate less flickering.

Table 7: Supplementary quantitative comparison on long video generation. This table includes the remaining eight VBench-Long metrics. Temporal Flickering is the normalized score, for which higher is better.

Model	App. Style ↑	Background Consist. ↑	Human Action ↑	Scene ↑	Subject Consist. ↑	Temporal Flickering ↑	Temporal Style ↑	Temporal (2D) ↑
				30 seconds				
CausVid	24.19	97.04	18.70	26.76	98.05	97.12	22.39	56.36
Rolling Forcing	23.91	96.66	20.02	**34.77**	**97.71**	97.69	24.85	56.93
LongLive	23.61	96.62	21.14	31.50	97.61	97.69	24.54	57.33
Self Forcing	23.89	95.84	21.14	30.90	96.31	97.18	24.18	57.14
+ Deep Forcing	**24.08**	95.78	**21.44**	30.62	96.19	96.86	24.35	57.66
+ Pyramid Forcing	23.82	**96.30**	20.75	32.61	**97.28**	**97.35**	**24.55**	**58.41**
Causal Forcing	23.39	**95.27**	19.78	**34.36**	**96.62**	**95.45**	23.03	58.21
+ Pyramid Forcing	**23.89**	94.88	**23.83**	33.23	95.25	94.50	**23.21**	**60.05**
				60 seconds				
CausVid	24.16	97.18	18.80	25.43	98.18	97.24	21.65	55.93
Rolling Forcing	23.89	96.64	20.49	32.51	97.74	97.72	24.20	56.65
LongLive	23.80	96.49	20.82	**33.64**	97.44	97.68	**24.54**	57.43
Self Forcing	23.56	95.53	18.22	23.95	95.75	96.85	22.33	55.00
+ Deep Forcing	**24.15**	96.15	21.09	**33.24**	96.93	**97.41**	**24.48**	57.36
+ Pyramid Forcing	23.94	**96.20**	**21.62**	29.90	**97.30**	97.30	24.16	**58.09**
Causal Forcing	23.33	**95.09**	19.63	33.20	**96.57**	**95.39**	**22.51**	57.44
+ Pyramid Forcing	**23.69**	94.58	**22.96**	**33.44**	94.78	94.00	22.41	**59.91**

B Controlled Ablation Studies

We examine the contribution of head-specific assignment, the choice of Wave sampling and Veil merging, and sensitivity to cache and classification parameters. Each study is interpreted using its own controls and tested parameter range.

B.1 Strong Uniform Policies and Head-Specific Assignment

Table 8 compares AWV with uniform Stride, Cyclic, and Merge policies. AWV reaches 82.04, exceeding the strongest uniform baseline, All-Merge, by 0.42 points. This aggregate difference concerns policy matching within S1, rather than the full framework's gain over native caching.

The second panel tests the assignments within a separate experiment group. The full method outperforms within-layer random assignment by 0.45 points. Shuffling strategies while holding capacities fixed reduces the score by 0.38 points, while shuffling capacities with strategies fixed reduces it by 0.28 points. These differences support both assignments within S2; they do not quantify prompt-level uncertainty or establish matched resident storage.

B.2 Wave Sampling Period and Phase

Table 9 varies the Wave sampling interval over $P \in \{4, 5, 6, 7, 8\}$ latent-frame indices. The best tested spacing is $P = 6$, with $P = 5$ and $P = 7$ within 0.10 points. The reported phase shift at $P = 6$ reduces the score from 82.02 to 81.72; its numerical offset is unspecified. Section D.5 examines the relation to temporal RoPE.

B.3 Effect of Spatiotemporal Mean Merging

At a matched attention-read budget, spatiotemporal mean merging reaches 82.06, improving over unmerged entries by 0.27 points and temporal means by 0.12 points (Table 10). This comparison concerns the merge operation; the capacity scan in Table 13 varies the middle-cache capacity of each head type. The highest scores occur at $C_A = 4$, $C_W = 4$, and $C_V = 2$. Veil's rows 2, 3, and 4 count retained merged units; they do not denote the temporal merge range m.

Table 8: Strong uniform-policy controls (S1) and routing/capacity ablations (S2). Results are 30-second Self Forcing VBench-Long Total Scores. The two panels are separate experiment groups.

Setting	Total Score ↑
S1: Uniform-policy controls	
All-Stride	81.43
All-Cyclic	81.57
All-Merge	81.62
AWV	**82.04**
S2: Routing and capacity assignment	
Within-layer random assignment	81.64
Shuffled strategies, fixed capacities	81.71
Shuffled capacities, fixed strategies	81.81
Full method (AWV)	**82.09**

Table 9: Wave period and phase ablation (S3). P is the sampling interval in latent-frame indices. Results are 30-second Self Forcing VBench-Long Total Scores.

Period P	Phase	Total Score ↑
4	Default	81.83
5	Default	81.94
6	Default	**82.02**
7	Default	81.92
8	Default	81.85
6	Shifted	81.72

Table 10: Veil merging at a matched attention-read budget (S4). Results are 30-second Self Forcing VBench-Long Total Scores.

Veil history representation	Total Score ↑
No merging	81.79
Temporal mean	81.94
Spatiotemporal mean	**82.06**

B.4 Impact of Sink and Forced Recent Frames

Tables 11 and 12 examine sink and recent retention on one NVIDIA H200 GPU. With $S = 3$, the best tested recent window is $R = 4$ (81.75). With $R = 4$, $S = 2$ reaches 81.93, compared with 81.75 for the default $S = 3$. These results support compact allocations without establishing that the default is globally optimal.

Table 11: Ablation Study on Different Recent Window Size R (Fixed $S = 3$).

Recent Frames	**Dynamic Degree** ↑	**Motion Smooth.** ↑	**Overall Consist.** ↑	**Imaging Quality** ↑	**Aesthetic Quality** ↑	**Quality Score** ↑	**Semantic Score** ↑	**Total Score** ↑
2	19.17	**99.09**	25.97	**73.01**	65.40	86.79	54.34	80.30
3	40.00	98.73	**26.18**	72.79	**65.62**	87.68	**55.22**	81.19
4	**55.63**	98.78	25.52	72.27	64.58	**88.61**	54.30	**81.75**
5	52.71	98.81	25.68	72.86	64.86	88.51	54.38	81.68
6	53.96	98.79	25.48	72.79	64.42	88.48	54.38	81.66

B.5 Sensitivity to Head-Specific Capacities

Table 13 varies the middle-cache capacity for each head type. Anchor reaches its highest reported Total Score at $C_A = 4$, although the change from one to two is not monotonic. Wave also reaches its highest score at $C_W = 4$. For Veil, the rows 2, 3, and 4 denote C_V, the number of retained merged

Table 12: Ablation Study on Different Sink Frames S (Fixed $R = 4$).

Sink Frames	Dynamic ↑ Degree	Motion ↑ Smooth.	Overall ↑ Consist.	Imaging ↑ Quality	Aesthetic ↑ Quality	Quality ↑ Score	Semantic ↑ Score	Total ↑ Score
1	**67.29**	98.64	25.27	71.38	62.93	**88.77**	53.35	81.69
2	58.54	98.77	**25.81**	72.18	64.28	88.68	**54.91**	**81.93**
3	55.63	98.78	25.52	**72.27**	64.58	88.61	54.30	81.75
4	56.46	**98.79**	25.52	**72.27**	**64.59**	88.66	54.15	81.76
5	57.08	**98.79**	25.52	**72.27**	**64.59**	88.72	54.26	81.83

units. The highest reported score is 82.242 at $C_V = 2$, compared with 82.102 at three and 82.046 at four.

The separate m rows in Table 5 examine merge range. Across the capacity scan, gains are not monotonic, so retaining more history does not uniformly improve quality in the tested settings.

Table 13: Ablation of head-specific middle-cache capacities on VBench-Long Lite. The three groups vary C_A, C_W, and C_V, respectively. Veil capacity counts retained merged temporal blocks; the values 2, 3, and 4 do not denote the temporal merge range m.

Capacity	Dynamic Degree ↑	Motion Smoothness ↑	Overall Consistency ↑	Imaging Quality ↑	Aesthetic Quality ↑	Quality Score ↑	Semantic Score ↑	Total Score ↑
	Anchor Head (Stride), capacity C_A							
1	51.953	98.694	25.984	71.430	**65.502**	88.378	**57.076**	82.118
2	52.148	98.729	26.042	**71.451**	65.352	88.386	57.017	82.112
4	**52.734**	**98.751**	**26.069**	71.429	65.267	**88.482**	56.894	**82.165**
	Wave Head (Periodic), capacity C_W							
3	50.000	**98.756**	**26.057**	71.512	65.457	88.287	**57.290**	82.088
4	**53.125**	98.748	26.024	71.629	**65.496**	**88.570**	56.718	**82.200**
5	52.344	98.754	25.959	**71.708**	65.223	88.409	56.775	82.082
	Veil Head (Mean Merge), capacity C_V							
2	**53.711**	98.716	**26.106**	71.466	65.512	**88.520**	**57.131**	**82.242**
3	52.734	98.751	25.962	**71.719**	65.428	88.492	56.540	82.102
4	50.195	**98.786**	26.035	71.490	**65.610**	88.409	56.592	82.046

B.6 Sensitivity to Classification Thresholds

Tables 14 and 15 scan the period and sign-rate thresholds separately. The best tested settings are $\beta = 6.4$ and $\alpha = 0.8$. Total Score ranges from 81.67 to 82.16 in the period scan and from 81.55 to 82.16 in the sign-rate scan.

Individual metrics remain sensitive to these choices: increasing β from 6.4 to 6.5 reduces Dynamic Degree from 50.59 to 46.48.

Table 14: Ablation of the period threshold with the sign-rate threshold fixed at 80%.

Period Threshold	Dynamic ↑ Degree	Motion ↑ Smooth.	Overall ↑ Consist.	Imaging ↑ Quality	Aesthetic ↑ Quality	Quality ↑ Score	Semantic ↑ Score	Total ↑ Score
6.4	**50.59**	98.72	26.00	71.74	65.23	**88.32**	**57.54**	**82.16**
6.5	46.48	98.78	26.23	72.72	65.43	88.07	56.93	81.85
7.0	46.09	**98.80**	26.14	72.74	**65.50**	88.11	56.26	81.74
8.0	46.29	98.78	**26.27**	72.64	65.39	88.06	56.12	81.67
9.0	46.29	98.79	26.19	**72.77**	65.30	88.08	56.61	81.79

Table 15: Ablation of the sign-rate threshold with the period threshold fixed at 6.4.

Sign Rate Threshold	Dynamic ↑ Degree	Motion ↑ Smooth.	Overall ↑ Consist.	Imaging ↑ Quality	Aesthetic ↑ Quality	Quality ↑ Score	Semantic ↑ Score	Total ↑ Score
70%	45.31	98.74	25.90	**72.78**	65.12	87.86	56.32	81.55
80%	**50.59**	98.72	**26.00**	71.74	**65.23**	**88.32**	**57.54**	**82.16**
90%	49.22	**98.84**	25.72	71.34	64.38	88.09	56.84	81.84

Table 16 adds a separate comparison with All-Merge. All three sign-rate thresholds, 70%, 80%, and 90%, exceed the same-group All-Merge score of 81.67 by 0.15–0.38 points. The best tested threshold is 80%, but the gain persists across the tested range.

Table 16: Sign-rate threshold sensitivity with a same-group All-Merge baseline (S5). Results are 30-second Self Forcing VBench-Long Total Scores; differences are in score points.

Setting	Total Score ↑	Δ vs. All-Merge
All-Merge	81.67	—
AWV, $\alpha = 70\%$	81.82	+0.15
AWV, $\alpha = 80\%$	**82.05**	+0.38
AWV, $\alpha = 90\%$	81.92	+0.25

C Long-Horizon and Cross-Backbone Evaluation

We evaluate whether the gains extend to longer generation and a second backbone. These comparisons are separate from the 30-second Self Forcing ablations.

C.1 Long-Horizon Validation

In the supplementary 60-second experiment, AWV improves Total Score by 3.02 points over native Self Forcing and by 0.51 points over the same-group All-Merge baseline (Table 17). These differences measure full-video quality. Table 17 provides no window-level values for locating the gain in time; Table 1 reports a separate experiment.

C.2 Independent Calibration on Causal Forcing

With independent head calibration on Causal Forcing, AWV improves Total Score by 0.62 points and Imaging Quality by 1.47 points over the native backbone, while Motion Smoothness decreases by 0.28 points (Table 17). This comparison evaluates independent calibration against native caching; it does not isolate calibration itself or compare with a Causal Forcing uniform-policy baseline.

Table 17: Supplementary long-horizon and backbone-transfer results. Entries are score-point differences within each experiment group; absolute scores are not inferred from other tables.

Comparison	Metric	Difference
S6: Self Forcing, 60-second generation		
AWV vs. native Self Forcing	Total Score	+3.02
AWV vs. All-Merge	Total Score	+0.51
S7: Causal Forcing with independent calibration		
AWV vs. native Causal Forcing	Total Score	+0.62
AWV vs. native Causal Forcing	Imaging Quality	+1.47
AWV vs. native Causal Forcing	Motion Smoothness	−0.28

D Head Classification and Attention Analysis

This section specifies the head-classification procedure and then examines the attention patterns that motivate it. We distinguish empirical sign and spectral features from claims about a head's function.

D.1 From Token Logits to Historical-Frame Sequences

We distinguish token-level logits, frame-level summaries, and normalized attention weights. For a fixed layer and head, let $z_{qk} = Q_q K_k^\top / \sqrt{D}$ denote the logit after positional encoding and before softmax. The dependence on prompt, seed, and denoising step is omitted for clarity. Let $\mathcal{Q}_t$ be the query tokens of latent frame t, $\mathcal{K}_s$ the key tokens of frame s, and $\mathcal{L}(q)$ the keys allowed by the model's attention mask. A frame-level logit summary is

$$a_t(s) = \frac{1}{|\Omega_{t,s}|} \sum_{(q,k)\in\Omega_{t,s}} z_{qk}, \qquad \Omega_{t,s} = \{(q,k) : q \in \mathcal{Q}_t,\ k \in \mathcal{K}_s \cap \mathcal{L}(q)\}. \tag{6}$$

Empty pairs and masked entries are excluded. The $T \times T$ matrix in the main text summarizes frame-level logits, rather than individual tokens. A last-block summary uses the union of query tokens in that block and differs from a last-frame summary.

The sequence contains clean-history frames $s < t$, including sink frames, and excludes the current noisy block. Restrictions to intermediate history are recorded separately. We retain one sequence per prompt, seed, query set, and denoising step before classification; averaging these sequences first defines a different procedure. Exact regeneration of the archived visualizations requires their extraction configuration.

Figures 8 and 9 illustrate frame-level summaries from the same layer for two prompts. The heatmaps retain query and history positions, while the curves summarize the last block.

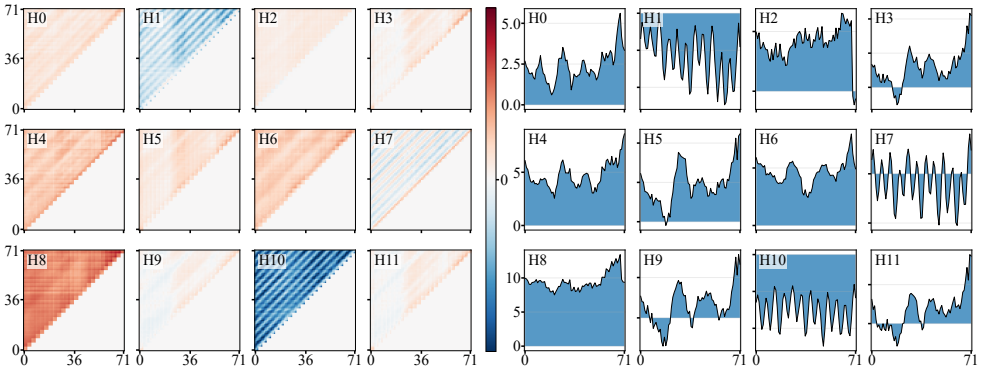


Figure 8: Pre-softmax attention visualization for a 72-latent-frame Self Forcing rollout at Layer 23: "A majestic eagle soaring through a cloudy sky, cinematic lighting."

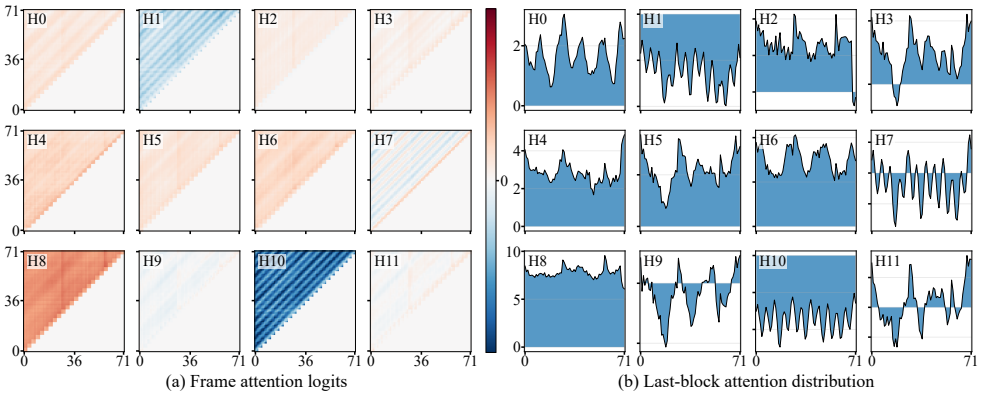


Figure 9: Pre-softmax attention visualization for a 72-latent-frame Self Forcing rollout at Layer 23: "A FPV drone shot through a castle on a cliff."

D.2 Sign Rates and the Classification Rule

For a historical sequence $a = (a_1, \ldots, a_L)$, we use

$$r_{\mathrm{pos}} = \frac{1}{L} \sum_{n=1}^{L} \mathbf{1}\{a_n > 0\}, \qquad r_{\mathrm{nonpos}} = 1 - r_{\mathrm{pos}}. \tag{7}$$

The second statistic corresponds to r_{neg} in the main text and includes exact zeros. With $\alpha = 0.8$, we first assign Anchor when $r_{\mathrm{pos}} \geq \alpha$ and Veil when $r_{\mathrm{nonpos}} \geq \alpha$. For the remaining sequences, we assign Wave when a valid spectral estimate satisfies $\widehat{P} < \beta$, with $\beta = 6.4$. Otherwise, we use the mean-logit fallback: Anchor if $\bar{a} > 0$, and Veil otherwise. This ordering makes the three assignments mutually exclusive.

These rules define empirical groups under a fixed model parameterization. Their functional interpretation is discussed in Section D.3.

D.3 Interpreting Pre-Softmax Logits

We use pre-softmax logits as empirical features for head classification. Their absolute signs do not, by themselves, determine how a head uses historical information. For a query row z and any scalar c,

$$\mathrm{softmax}(z + c\mathbf{1}) = \mathrm{softmax}(z). \tag{8}$$

For example, $(1, 2)$ and $(-2, -1)$ have identical weights, approximately $(0.269, 0.731)$, despite opposite sign rates. A negative logit can receive the largest weight in its row, so its sign alone does not imply inhibition.

Under a fixed checkpoint and extraction procedure, sign rates may still correlate with useful cache policies. To evaluate this relationship, we distinguish raw sign rates from row-centered scores,

$$z^c_{qk} = z_{qk} - \frac{1}{|\mathcal{L}(q)|} \sum_{j \in \mathcal{L}(q)} z_{qj}, \tag{9}$$

and from normalized frame-level attention mass,

$$w_{t,s} = \frac{1}{|\mathcal{Q}_t|} \sum_{q \in \mathcal{Q}_t} \sum_{k \in \mathcal{K}_s \cap \mathcal{L}(q)} \frac{\exp(z_{qk})}{\sum_{j \in \mathcal{L}(q)} \exp(z_{qj})}. \tag{10}$$

Centering removes sensitivity to a constant row shift. In Equation 10, softmax is computed over every legal key, including current-block keys, before spatial aggregation. Applying softmax to averaged logits generally produces a different result.

For a historical segment $\mathcal{G}$, its attention mass is $M_t(\mathcal{G}) = \sum_{s \in \mathcal{G}} w_{t,s}$. Disjoint sink, middle, recent, and current-block segments make their contributions comparable. Frame-level entropy is $-\sum_s w_{t,s} \log w_{t,s}$ over all legal frame groups, with $0 \log 0 = 0$.

The relationship between normalized attention and logit sign can vary with the prompt, query position, and denoising step. Policy comparisons in Section B.1 test the labels' practical value; logit colors do not establish causal suppression.

Figures 10 and 11 provide additional prompt-level examples of broad positive logits, oscillations, and low intermediate-history logits. These selected patterns motivate the classification features without establishing invariance across inputs.

D.4 Spectral Estimation and Its Scope

The reported spectral analysis uses 256 prompts, 30 layers, and 12 heads per layer, with historical indices 0–68. FFT assumes uniformly spaced indices; an irregularly selected cache cannot be treated as consecutive frames. The recorded preprocessing consists of a first-order difference, mean removal, a Hann window, and a real-valued FFT. For a length-L sequence, let $N = L - 1$ and

$$d_n = a_{n+1} - a_n, \qquad u_n = (d_n - \bar{d}) \, \frac{1 - \cos(2\pi n/(N-1))}{2}, \quad n = 0, \ldots, N-1, \tag{11}$$

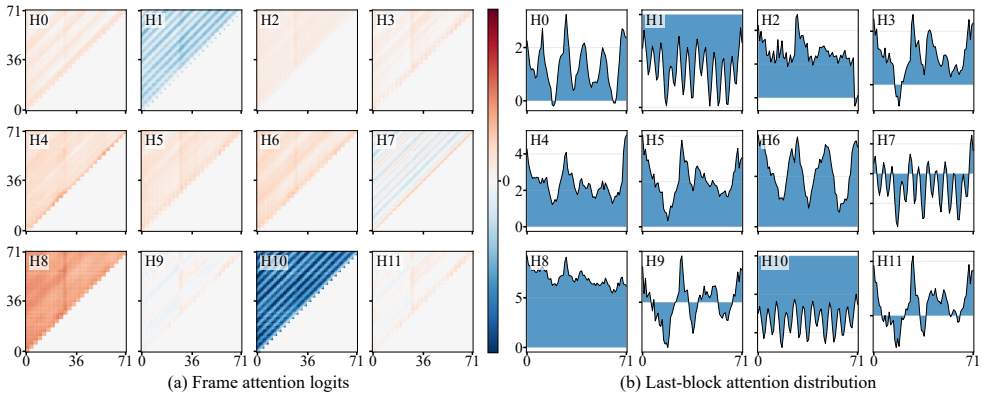


Figure 10: Pre-softmax attention visualization for a 72-latent-frame Self Forcing rollout at Layer 23: "Super fast zoom out from the peak of a frozen mountain where a lonely hiker is arriving to the summit."

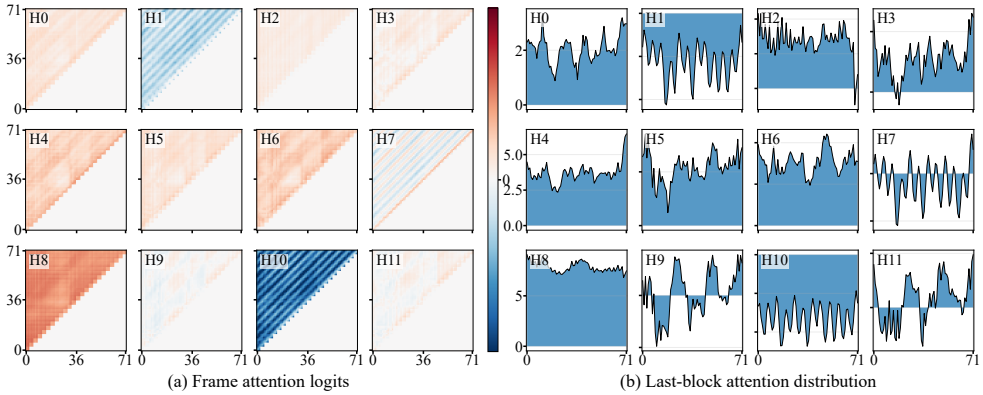


Figure 11: Pre-softmax attention visualization for a 72-latent-frame Self Forcing rollout at Layer 23: "A white and orange tabby cat is seen happily darting through a dense garden, cinematic warm tones, shallow depth of field."

where the sequence is indexed from zero in this expression. For an FFT length $N_{\mathrm{FFT}} \geq N$, the positive-frequency bins are $f_k = k/N_{\mathrm{FFT}}$, measured in cycles per latent-frame index. The zero-frequency component is excluded.

Harmonic folding combines spectral energy at a candidate fundamental and its harmonics. A reproducible specification takes the harmonic order J, nonnegative weights η_j, FFT length, and candidate-bin set $\mathcal{F}$ as explicit inputs:

$$S(k) = \sum_{\substack{1 \leq j \leq J \\ jk \leq \lfloor N_{\mathrm{FFT}}/2 \rfloor}} \eta_j |X_{jk}|^2, \qquad k^\star = \arg\max_{k \in \mathcal{F}} S(k), \qquad \widehat{P} = \frac{N_{\mathrm{FFT}}}{k^\star}, \tag{12}$$

where $X = \mathrm{rFFT}(u)$ and $\mathcal{F}$ contains only positive-frequency bins. Equation 12 uses the FFT grid without peak interpolation and resolves equal maxima by the smallest bin index. Finite, nonempty sequences with fewer than four samples or zero spectral energy have no valid estimate and use the mean-logit fallback. Empty or nonfinite sequences are flagged as invalid extractions and excluded from voting. These conventions specify a diagnostic rerun; the folding order, weights, search range, window convention, and interpolation settings behind the archived figure still require confirmation. The figure is therefore not a complete specification of the estimator.

A highest spectral peak can also occur in a nonperiodic sequence. Alongside $\widehat{P}$, a useful diagnostic is the unfolded peak-energy fraction

$$\rho = \frac{|X_{k^\star}|^2}{\sum_{k \in \mathcal{F}} |X_k|^2}, \tag{13}$$

together with autocorrelation across query positions and denoising steps. These diagnostics do not alter the reported $\widehat{P} < \beta$ rule unless a new classification configuration is explicitly evaluated.

First-order differencing has magnitude response $2|\sin(\omega/2)|$ and emphasizes high frequencies. We therefore examine controls without differencing and without harmonic folding. The dominant peak remains near $\widehat{P} \approx 6$–6.5 (Table 18), indicating that the observed concentration is not solely produced by these preprocessing choices. Peak location alone does not establish oscillation strength or stability across query positions and denoising steps.

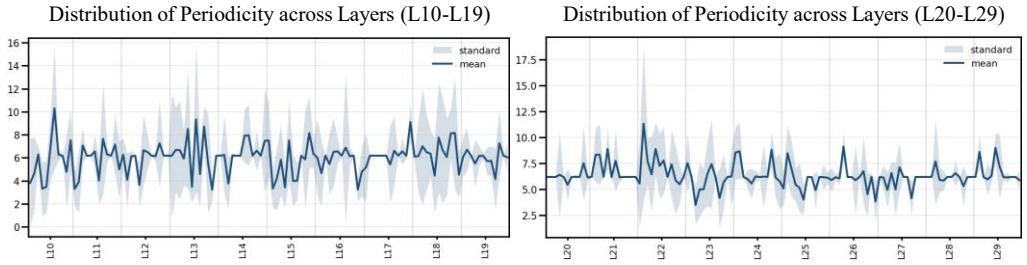


Figure 12: Distribution of estimated periods across layers L10–L29 under the recorded preprocessing. Peaks near six latent frames motivate periodic sampling; they do not establish an exact period or the phase of high-attention history.

D.5 RoPE and Wave-Head Periodicity

RoPE is a candidate explanation for the observed temporal oscillations. For temporal channel pair j, the rotated query–key product can be expanded as

$$z_{t,s}^{\text{time}} = \sum_j \left[A_j(q_t, k_s)\cos(\theta_j \Delta) + B_j(q_t, k_s)\sin(\theta_j \Delta)\right], \qquad \Delta = \widetilde{t} - \widetilde{s}, \tag{14}$$

where $\widetilde{t}, \widetilde{s}$ are the temporal positions actually supplied to RoPE, and the content-dependent coefficients include the attention scaling. Spatial RoPE channels contribute additional terms. For the frequency schedule $\theta_j = 10000^{-2j/d_{\text{rot}}}$, small j corresponds to higher angular frequency. In particular, $\theta_0 = 1$ is the highest frequency in this schedule, not the lowest.

The contribution of each frequency depends on query/key content, temporal-channel allocation, and position remapping. The frequency schedule alone therefore does not explain which heads exhibit clear oscillations.

A component $\cos(\Delta)$ has continuous period $2\pi \approx 6.283$, but $\cos(n)$ at integer n has no exact positive integer period. Rounding to six is a sampling heuristic. A finite-window spectral peak also depends on sequence length, windowing, and the FFT grid.

A period does not determine the phase of high-attention positions. For $A\cos(\omega\Delta + \phi)$, changing ϕ can move the same sampled offset from a peak to a trough. The period and phase comparison in Section B.2 evaluates the effect of these two choices on generation quality.

Holding unrotated Q/K fixed, temporal position scales of 0.75, 1, and 1.25 yield dominant periods of approximately 8, 6, and 4.8 latent-frame indices (Table 18). This inverse movement supports a contribution from temporal RoPE. Together with the preprocessing controls in Section D.4, it argues against a peak caused solely by differencing or harmonic folding, without establishing dominance of a single channel.

Here, periods are measured in latent-frame indices. They do not represent an object's physical motion period or the video's playback rate.

Figures 13 and 14 show an extended Self Forcing record and a Causal Forcing example for the eagle prompt. The caption identifies the unequal plotting ranges in the archived extended record.

Table 18: FFT preprocessing and RoPE diagnostics (S8). Peak periods are approximate and measured in latent-frame indices. The position-scaling diagnostic holds unrotated Q/K fixed.

Diagnostic setting	Dominant peak period
Controls without differencing or harmonic folding	$\approx$ 6–6.5
Temporal position scale 0.75	$\approx$ 8
Temporal position scale 1	$\approx$ 6
Temporal position scale 1.25	$\approx$ 4.8

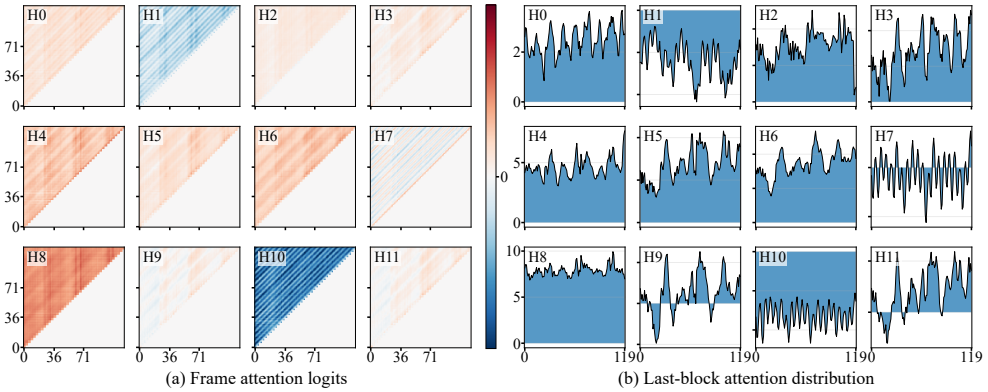


Figure 13: Self Forcing attention visualization at Layer 23 for the eagle prompt. The archived heatmaps show indices 0–71, whereas the last-block curves extend to index 119. These unmatched ranges are retained for qualitative inspection and are not used as a matched-window duration comparison.

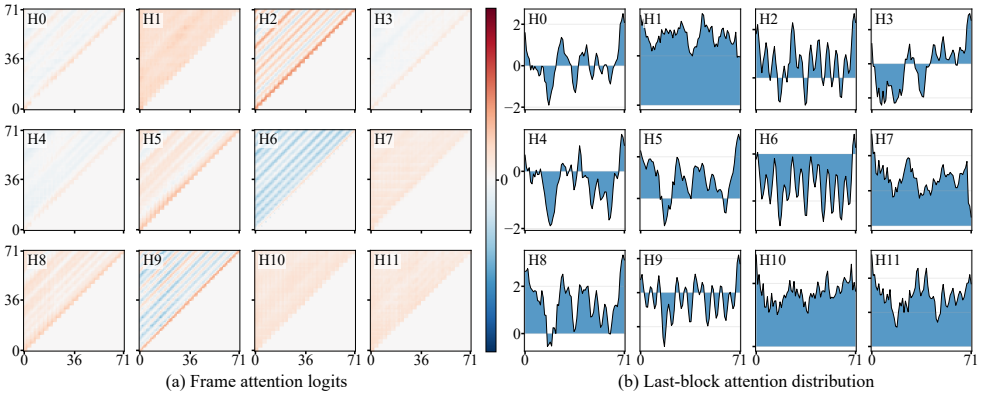


Figure 14: Pre-softmax attention visualization for a 72-latent-frame Causal Forcing rollout at Layer 23: "A majestic eagle soaring through a cloudy sky, cinematic lighting."

D.6 Aggregation and Label Stability

Figures 15 and 16 aggregate prompt-level labels over 256 prompts. Classification precedes voting; classifying a mean logit sequence is a different procedure. This analysis set is distinct in purpose from the 32-prompt calibration setting.

Relative to the MovieGen reference labels, reported agreement is 99.17–99.72% across calibration subsets, 91.67–94.44% across FETV, Step-Video-T2V-Eval, and ChronoMagic-Bench, and at least 98.89% across 15–60-second generation. Three FETV prompts with four seeds yield 89.95% mean agreement. Relative to Self Forcing, Causal Forcing and Causal-rCM yield 91.94% and 87.22%, respectively. These measurements concern assignment agreement, not downstream quality or universal head identity.

For deterministic diagnostic reruns, fix the checkpoint, prompts, seeds, query frames, denoising steps, attention mask, and spectral parameters. Extract each sequence using Equation 6, then apply the sign-rate tests, valid-period test, and mean-logit fallback in order. Vote within each prompt before voting across prompts with equal weight. Diagnostic ties follow Anchor, Wave, Veil order

and are flagged. Record each layer–head pair's label, vote fractions, winning margin, tie indicator, and extraction settings. These records distinguish label changes from changes in the sampling and aggregation procedure.

Original within-prompt aggregation and tie handling remain incompletely documented; this rerun convention does not reconstruct archived labels. Agreement can hide layer-, query-, or step-specific changes. Similar scores across calibration sizes likewise do not establish identical labels, and label drift requires a separate downstream-quality assessment.

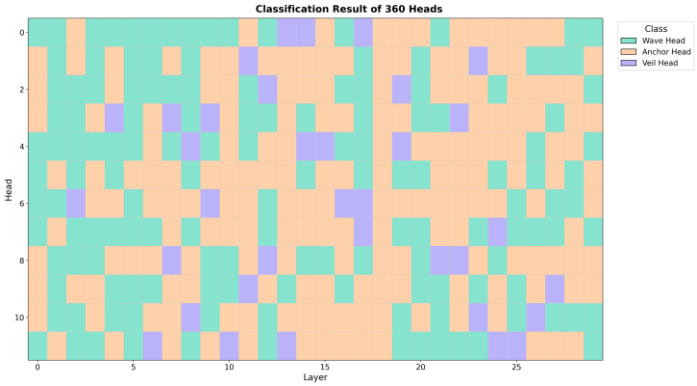


Figure 15: Prompt-level head labels aggregated by voting over 256 prompts with 15-second videos in Self Forcing. The period and sign-rate thresholds are 6.4 and 80%, respectively. This is the analysis set described in Section A.1.

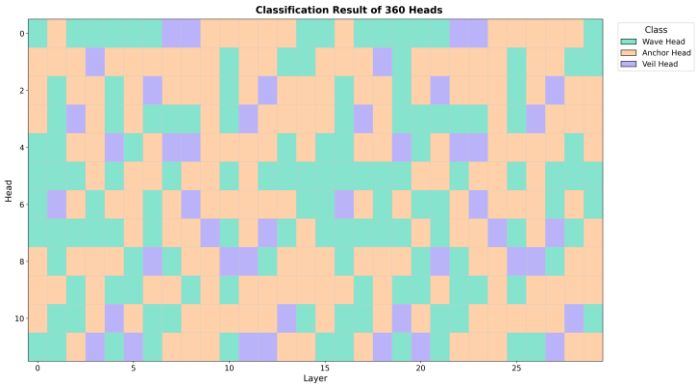


Figure 16: Prompt-level head labels aggregated by voting over 256 prompts with 15-second videos in Causal Forcing. The period and sign-rate thresholds are 6.4 and 80%, respectively. This is the analysis set described in Section A.1.

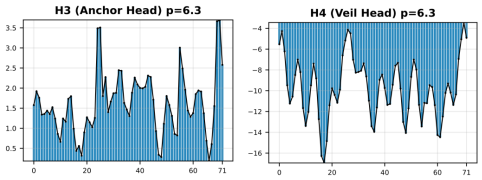
D.7 Classification Boundaries

A short dominant-period estimate is not sufficient to identify a stable oscillation. Figure 17a shows sequences with spectral peaks but strongly one-sided logits, which the sign-rate test assigns before periodicity is considered. Figure 17b shows examples close to the sign-rate or period thresholds. Small changes in the sampled sequence can change these labels.

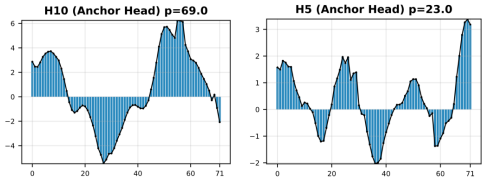
The mean-logit fallback assigns the remaining heads deterministically, without establishing the best policy for a boundary head. These examples concern classification stability, not video-generation failures.

E Cache Definitions and Implementation

We define the cache units, selection rules, and online updates before discussing their storage requirements and experimental configuration. The merge-operation ablation is reported in Section B.3.



(a) Logit sequences with one-sided bias and spectral peaks.



(b) Sequences near classification thresholds.

Figure 17: Classification diagnostics. Spectral peaks and threshold proximity can affect the assigned label; these plots do not show video-generation failure cases.

E.1 Cache Units, History, and the Current Block

We use zero-based latent-frame indices in this section. When the next block begins at t, the clean history is $\mathcal{H}_t = \{0, \dots, t-1\}$. Before compression, a layer's cache has shape $K, V \in \mathbb{R}^{B\times H\times T\times N\times D}$, where $N = H_g W_g$ is the number of spatial tokens per latent frame and D is the head dimension. Flattening the frame and spatial axes gives the usual token sequence. The middle-cache capacities C_A, C_W, and C_V are distinct from the sink count S and forced recent count R.

For one head, define

$$\mathcal{S}_t = \mathcal{H}_t \cap \{0, \dots, S-1\}, \qquad \mathcal{R}_t = \{s \in \mathcal{H}_t : s \geq t - R\} \setminus \mathcal{S}_t. \tag{15}$$

Selected middle frames are restricted to $\mathcal{H}_t \setminus (\mathcal{S}_t \cup \mathcal{R}_t)$ and sorted by their original indices. Sink frames take precedence over overlapping recent frames, and both take precedence over middle candidates. Invalid indices and duplicates are removed without refilling the candidate set. During warmup, fewer than the nominal number of middle units may therefore be read.

The current block's noisy K/V are separate from the clean-history cache. They participate only where allowed by the backbone's attention mask. Chunk-causal attention can permit attention within the current block, whereas frame-causal attention can impose a stricter boundary. Cache compression does not replace that mask. Clean K/V are committed once after the block's clean-context pass; intermediate denoising states do not become separate historical frames.

E.2 Anchor Selection and Storage Requirements

The adaptive-center Anchor rule used throughout the main and supplementary Pyramid Forcing experiments is

$$\widetilde{\mathcal{A}}_t = \left\{ \left\lfloor \left(j + \frac{1}{2}\right) \frac{t}{C_A} \right\rfloor : j = 0, \dots, C_A - 1 \right\}, \qquad \mathcal{A}_t = \widetilde{\mathcal{A}}_t \cap \big(\mathcal{H}_t \setminus (\mathcal{S}_t \cup \mathcal{R}_t)\big). \tag{16}$$

For $t = 0$, the selection is empty. Flooring is applied to the complete index expression, followed by filtering, deduplication, and chronological ordering. Repeated centers at short history lengths yield fewer than C_A frames.

This rule specifies a set of attention inputs, not a bounded-memory update algorithm. As t changes, a newly selected index may refer to a frame discarded at an earlier step. Exact evaluation therefore requires access to the full historical K/V, an external archive, or recomputation. Directly retaining all historical entries costs $O(tND)$ per head even though each attention call reads at most $C_A ND$ middle-cache elements.

After each clean-context pass, the history boundary advances and the adaptive centers are recomputed using the current history length. The next read set is selected from this history, independently of membership in the previous read set. Filtering then removes sink/recent overlap, invalid indices, and duplicates. Capacity ablations change C_A within this rule; the default $C_A = 4$ and other parameter conventions are summarized in Section E.8.

E.3 Wave Selection and Phase Buckets

For the default phase-zero sampling rule, the clean-history candidates are

$$\widetilde{\mathcal{W}}_t = \{t - iP : i = 1, \dots, C_W\}, \qquad \mathcal{W}_t = \widetilde{\mathcal{W}}_t \cap \big(\mathcal{H}_t \setminus (\mathcal{S}_t \cup \mathcal{R}_t)\big). \tag{17}$$

Starting at $i = 1$ excludes the current frame. Negative indices and overlap with sink or recent frames are removed. Here P is an integer spacing in latent-frame indices; it is distinct from the real-valued spectral estimate and from the classification threshold β.

A phase-bucket representation stores a newly completed frame s in bucket $s \bmod P$, retaining at most C_W frames in each bucket. A query at time t reads bucket $t \bmod P$, after sink/recent filtering. Thus, the middle-cache read count is at most C_W, while the resident count can reach PC_W. A phase-offset diagnostic reads bucket $(t - \phi) \bmod P$ for $\phi \in \{0, \ldots, P - 1\}$ under the same read-count limit. Period alone does not determine which offset retains useful history.

E.4 Veil Spatiotemporal Mean Merging

Veil Heads use spatiotemporal mean merging. We distinguish three parameters: C_V is the number of retained merged temporal blocks, m is the number of latent frames in each block, and s is the spatial patch edge length in tokens. Let $\mathcal{B}_b$ contain m consecutive frames and let $\mathcal{P}_g$ be a spatial patch containing at most s^2 tokens. For each batch and head, the merged key and value are

$$\bar{K}_{b,g} = \frac{1}{m|\mathcal{P}_g|} \sum_{\tau \in \mathcal{B}_b} \sum_{u \in \mathcal{P}_g} K_{\tau,u,:}, \qquad \bar{V}_{b,g} = \frac{1}{m|\mathcal{P}_g|} \sum_{\tau \in \mathcal{B}_b} \sum_{u \in \mathcal{P}_g} V_{\tau,u,:}. \tag{18}$$

Each output remains a complete D-dimensional token. This operation averages across time and space; it does not slice or concatenate feature channels. For the $H_V = |\mathcal{H}_V|$ Veil Heads in a layer, the merged cache has shape $B \times H_V \times C_V \times N_s \times D$, where $N_s = \lceil H_g/s \rceil \lceil W_g/s \rceil$. Boundary patches are normalized by their actual token counts.

The default configuration retains $C_V = 2$ merged temporal blocks, each formed from $m = 2$ latent frames. Spatial patch size s controls a separate compression axis; the temporal range is not set by s^2. Completed blocks enter the retained merged history, while an unfinished block remains available for subsequent accumulation. Blocks overlapping the protected sink or recent history are excluded at readout.

The merge operator preserves corresponding channel coordinates, but positional handling additionally depends on the order of merging and RoPE and the representative temporal and spatial positions. Averaging unrotated keys before applying RoPE differs from averaging already rotated keys. The spatial patch size, representative positions, and rotation order remain separate configuration fields; they cannot be recovered from $C_V = 2$ and $m = 2$ alone. Table 13 varies retained capacity C_V, while the m rows in Table 5 vary temporal merge range. These two ablations therefore test different aspects of Veil compression.

Mean merging preserves feature coordinates but changes logits and softmax normalization, so it is not generally equivalent to attention over the original tokens. Section B.3 evaluates its quality at a matched read budget.

E.5 Online Update Procedure

The online procedure follows the adaptive-center selection in Equation 16, together with Wave sampling and Veil mean merging. Historical access for Anchor is distinct from the packed attention-read buffer.

1. Initialize sink and recent stores, historical K/V access for Anchor Heads, Wave phase buckets, and Veil block accumulators for each layer–head pair, according to its assigned strategy.
2. For the current history boundary t, recompute Anchor centers and gather their historical K/V. Collect Wave candidates and completed Veil blocks, remove invalid indices and sink/recent overlap, and order retained entries chronologically. Append the current noisy block under the backbone's attention mask and positional mapping.
3. After the clean-context pass, commit each newly completed frame once. Update sink and recent stores and make clean Anchor K/V available for subsequent adaptive-center queries. Update Wave phase buckets and finalize each Veil block after all m frames arrive, retaining up to C_V completed blocks.

4. Advance the history boundary and recompute Anchor centers for the next query. Preserve unfinished Veil accumulators and historical access needed by future queries. Reset all state between independent videos.

Anchor reads at most $C_A ND$ middle-cache elements per head; materializing its complete history requires $O(tND)$ elements. Wave phase buckets can retain $O(PC_W ND)$, and Veil retains $O((C_V + 1)N_s D)$ including an unfinished block. Sink/recent stores, position metadata, packing buffers, and workspaces add further costs. The adaptive-center read bound alone does not specify total resident storage.

E.6 Attention Reads, Resident State, and GPU Memory

We distinguish the K/V tokens read by one attention call from those held in memory. Let $L_{l,h}^{\text{read}}$ and $L_{l,h}^{\text{resident}}$ be these token counts at layer l, head h, and let b be the bytes per element. For a common batch size B and head dimension D, the corresponding payload sizes are

$$M_{\text{read}} = 2bBD \sum_{l,h} L_{l,h}^{\text{read}}, \qquad M_{\text{resident}} = 2bBD \sum_{l,h} L_{l,h}^{\text{resident}}. \tag{19}$$

The factor two accounts for keys and values. These quantities exclude position metadata, packing buffers, activations, weights, allocator overhead, and other workspaces. Logical payload sizes also do not directly measure device-memory traffic or peak allocation.

For Wave's phase buckets, the middle read limit is $C_W N$ tokens per head while the resident limit is $PC_W N$. Equal read budgets therefore need not imply equal resident budgets. A resource comparison must state which budget is matched, provide the head counts and cache lengths for each layer, and report workspace allocation and measured GPU memory separately. A single nominal frame capacity does not establish equivalence across these quantities.

E.7 Efficiency Measurement Conventions

Latency comparisons require a common measurement boundary. Batch size, head dimension, query count, each head's KV length, dtype, denoising steps, and hardware/software versions determine the workload. Packing, device transfers, and VAE decoding must each be identified as included or excluded. Warmup, device synchronization, and repeated measurements determine how median and tail latency are summarized.

For video i with F_i decoded frames and latency ℓ_i, averaging F_i/ℓ_i differs from dividing the average frame count by the average latency. Consequently, a reported mean FPS need not equal the reciprocal of mean latency times a fixed frame count. Interpreting the recorded results in Table 3 requires the per-video frame counts, aggregation rule, and timing boundary.

E.8 Experimental Configuration and Parameter Conventions

Table 19 summarizes the common algorithm and default configuration used in the main and supplementary Pyramid Forcing experiments. Anchor uses adaptive-center selection throughout. All head types, including Wave, use $S = 3$ sink frames and $R = 4$ forced recent frames by default. Veil retains $C_V = 2$ merged temporal blocks with temporal range $m = 2$.

A shared algorithm does not imply identical experimental settings. Component ablations remove the stated components; policy and capacity ablations change the stated parameters. Each result remains associated with its own prompts, seeds, generation length, calibration, and comparison group.

Reproduction records additionally identify historical K/V access, spatial merge size, positional conventions, and the head-label file. These fields are distinct from the confirmed default capacities and sink/recent settings and must accompany the corresponding checkpoint, evaluator configuration, and per-prompt outputs.

F Additional Qualitative Results

Figures 18 and 19 compare 30-second generation in an object-centric scene and a scene with substantial camera and subject motion. In these selected examples, Pyramid Forcing better preserves appearance and scene layout across time.



Figure 18: Additional qualitative comparison on an object-centric scene for 30-second generation.



Figure 19: Additional qualitative comparison with substantial subject and camera motion for 30-second generation.

Table 19: Default configuration and parameter conventions for Pyramid Forcing. Main and supplementary experiments use adaptive-center Anchor; ablations specify changes to the corresponding parameters or components.

Item	Default configuration	Parameter convention
Anchor	Adaptive-center selection; $C_A = 4$. Centers follow the current history length.	Used throughout the main and supplementary evaluations; exact readout requires access to selected historical K/V.
Wave	Periodic historical sampling with integer spacing P and capacity $C_W = 4$.	Spacing and phase define the read set; the phase-bucket representation can retain up to PC_W frames.
Veil	Spatiotemporal mean merging; $C_V = 2$ retained blocks and $m = 2$ latent frames per block.	C_V, m, and spatial patch size s are separate parameters. Rotation order and representative positions require explicit configuration.
Sink/recent	$S = 3$ sink frames and $R = 4$ forced recent frames for all head types, including Wave.	Sink and recent entries remain unmerged; size ablations specify the setting used in each comparison.
Head labels	Offline head classification with the stated calibration setting.	Per-experiment records must identify the model, calibration and label file; the 256-prompt analysis serves a separate purpose.

Figures 20 and 21 extend the qualitative comparison to 60-second generation in an underwater environment and a portrait scene. They illustrate temporal preservation of scene structure and local details; these selected cases do not estimate the frequency of gains or failures.

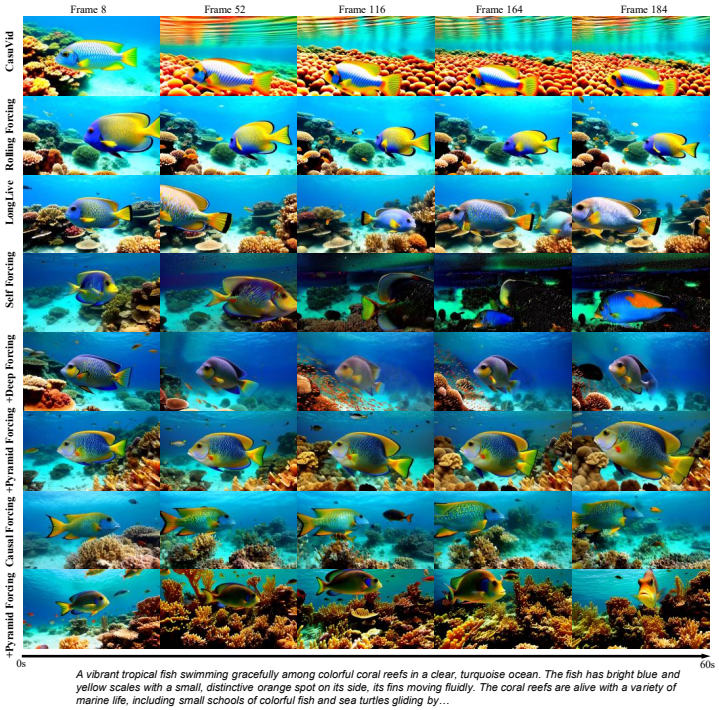


Figure 20: Additional qualitative comparison in an underwater scene for 60-second generation.



Figure 21: Additional qualitative comparison of a human portrait for 60-second generation.

G Limitations and Scope

G.1 Scope of Aggregate Comparisons

For a per-video metric, let $s_{A,p,k}$ and $s_{B,p,k}$ denote the scores of two methods on prompt p and seed k. With the same K seeds for each prompt, we first compute

$$d_p = \frac{1}{K}\sum_{k=1}^{K}(s_{A,p,k} - s_{B,p,k}), \qquad \bar{d} = \frac{1}{N_p}\sum_{p=1}^{N_p} d_p. \tag{20}$$

A paired analysis resamples prompts with both methods and all seeds together. It can report $\bar{d}$, a bootstrap interval, the fraction with $d_p > 0$, and the lower tail.

For VBench-Long Total Score, each bootstrap replicate must recompute the official multidimensional aggregation. Table means alone cannot recover confidence intervals or establish statistical significance.

The four-seed, 32-prompt MovieGen comparison yields a mean paired gain of 1.57 ± 0.14 (sample standard deviation). This measures generation-seed variation, separately from assignment variation or prompt-bootstrap uncertainty. Multiple seeds for one prompt are not independent semantic examples.

G.2 Long-Horizon Distribution Shift

Autoregressive video diffusion factorizes the sequence distribution as

$$p(x^{1:N}) = \prod_{t=1}^{N} p(x^t \mid x^{<t}). \tag{21}$$

Future frames depend on generated history, and changes to that history can affect later predictions beyond the training horizon. This dependence does not imply that every local error grows monotonically.

Self Forcing (Huang et al., 2025a) trains with self-generated rollouts to reduce a training–inference mismatch. Long-horizon extrapolation and cache selection introduce additional effects. The reported metrics describe their net outcome and do not isolate error-propagation mechanisms.

G.3 Observed Tradeoffs and Unmeasured Failure Rates

In the original 60-second Causal Forcing comparison, Total Score increases from 79.14 to 79.92 and Dynamic Degree from 57.03 to 86.39, while Subject Consistency, Motion Smoothness, and Imaging Quality decrease. The independently calibrated comparison in Section C.2 improves Imaging Quality but still reduces Motion Smoothness by 0.28 points. The quality tradeoff thus depends on the configuration.

Selected qualitative examples do not measure failure frequency. Identity changes, reappearance after occlusion, camera motion, and fine-texture preservation require inspection of prompt-level differences and the corresponding clips. Classification boundary cases in Section D.7 address a separate issue.

G.4 Applicability

Pyramid Forcing uses checkpoint-specific labels and selected capacities, sampling intervals, and merge settings. Changing the backbone, positional encoding, or generation regime may require recalibration and renewed comparison with uniform policies. Training-free deployment still incurs calibration and configuration-selection costs. Online head identification is a future direction; its value depends on whether adapting to label changes improves quality or efficiency beyond the fixed routing used here.