RESIDUAL FORCING: STRUCTURE-AWARE SPARSE ATTENTION FOR AUTOREGRESSIVE VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video diffusion enables streaming long-video generation, but self-attention over cached visual history remains a major inference bottleneck. Training-free block-sparse methods can flexibly select key blocks for each query block, yet routing based on inexpensive mean-pooled scores can miss concentrated interactions amplified by softmax. These misses degrade generation quality, while compensating with more key blocks erodes the efficiency gains; independently changing the selected context across chunks can also introduce periodic visual discontinuities. In this paper, we show that many of the missed high-response interactions follow identifiable attention structures that can be protected at low cost. We therefore introduce **Residual Forcing**, a training-free framework that separates structure preservation from residual attention selection. It protects spatially aligned key blocks, their axial halo tokens, and shared salient keys, then routes only the residual key space using reusable head-level priors and block-adaptive budgets. To improve temporal continuity, it shares candidate-frame choices across chunk boundaries without increasing the frame budget. Fused routing and dynamic sparse-attention kernels efficiently execute the resulting heterogeneous routes. On LongLive, Residual Forcing achieves a $4.12\times$ self-attention speedup in fixed-cache profiling and a $1.25\times$ end-to-end speedup for 30-second generation, while maintaining competitive video quality and mitigating chunk-boundary discontinuities.

1 INTRODUCTION

Autoregressive video diffusion enables streaming generation by denoising short chunks conditioned on previously generated context. Advances in sequence diffusion, causal distillation, and training on self-generated history have made this approach practical for long and interactive videos (Chen et al., 2024; Yin et al., 2024; Huang et al., 2025; Yang et al., 2025a). Nevertheless, self-attention over visual tokens remains a major inference bottleneck (Samuel et al., 2026; Shen et al., 2026). A bounded key–value (KV) cache limits history, but repeatedly attending to that context still incurs substantial cost. Efficient streaming generation therefore requires reducing attention computation while retaining the context most relevant to subsequent generation.

Block-sparse attention offers a flexible way to reduce this cost. Approaches for video diffusion transformers (DiTs) exploit local tiles and semantic token rearrangement (Zhang et al., 2025b; Yang et al., 2025b), inexpensive importance proxies (Zhang et al., 2025a; Xu et al., 2025; Shen et al., 2025), and adaptive support selection (Xia et al., 2025; Hu et al., 2026). Recent autoregressive methods combine sparse attention with hierarchical frame selection, persistent context, or cache compression (Lv et al., 2026; Xu et al., 2026; Samuel et al., 2026; Lu et al., 2026). Despite this progress, block-sparse attention methods still face three interconnected challenges.

Accurate block selection: Inexpensive scoring proxies, such as mean-pooled scores, can dilute concentrated query–key responses that softmax amplifies (Xu et al., 2025; Han et al., 2026). Consequently, blocks containing sparse yet dominant interactions can be systematically underestimated. **Fine-grained budget allocation:** a uniform block count cannot accommodate heterogeneous heads and input-dependent query demand (Wu et al., 2025; Zheng et al., 2026; Zhao et al., 2026). **Temporal continuity:** dynamic selection adapts to new content, but abrupt changes in historical evidence can disrupt chunk transitions (Ji et al., 2026; Kim et al., 2026).

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.

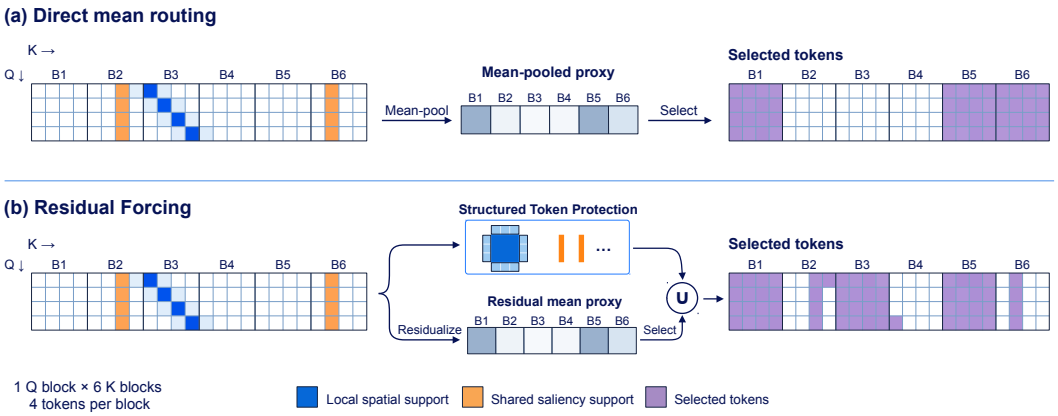


Figure 1: **Protect first, route the residual.** (a) Direct mean routing selects whole blocks from pooled scores, which can dilute concentrated responses. (b) Residual Forcing protects local spatial support (blue) and shared saliency support (orange), scores the remaining tokens, and unions protected and selected residual supports. Purple denotes the final selected tokens.

Our central observation is that many high-response interactions missed by inexpensive routing have identifiable structure. Spatially corresponding interactions form *local spatial support*, while a small set of keys attracts responses across many queries, forming *shared saliency support*. We observe that both structures are prevalent and can be identified in advance with lightweight computation. Figure 1 contrasts our design with current mean-routing approaches. Instead of routing directly over all tokens, our method first protects these structured supports, removes their contributions from block statistics, and routes over the residual support.

Therefore, we introduce **Residual Forcing**, a training-free framework combining structured token protection, head-calibrated residual routing, and cross-chunk context sharing. Within candidate frames, we protect spatially aligned tiles, neighboring strips, and shared salient keys, then score each residual block using its remaining mean key and token count. Offline calibration supplies one threshold per head; online, each query block applies it to its current residual distribution allowing its block budget to vary with the current content. To support continuity, boundary queries inherit retrieved historical frames that remain in a persistent cache region, while later queries adapt retrieval to current content. Fused routing and within-chunk index reuse efficiently assemble the mixed support, while attention is recomputed over the selected KV cache.

We evaluate short- and long-video generation using LongLive (Yang et al., 2025a) with the Wan2.1-T2V-1.3B backbone (Wan et al., 2025), assessing generation quality with VBench (Huang et al., 2023) and our boundary-continuity diagnostic. Compared with dense attention, Residual Forcing achieves $4.12\times$ self-attention speedup, together with $1.41\times$ DiT and $1.25\times$ end-to-end speedups for 30-second videos. Meanwhile, it achieves higher VBench scores than the evaluated sparse baselines and lower boundary-specific perceptual excess. Our contributions are threefold:

- We find that strong interactions missed by mean-pooled routing often follow local spatial patterns or concentrate on shared salient keys. We protect these interactions, remove their contributions from block statistics, and use head-calibrated thresholds to adapt residual routing budgets to each query block.

- We propose cross-chunk context sharing to preserve historical evidence at chunk boundaries without increasing cache capacity or the candidate-frame count, and evaluate continuity with a boundary-focused perceptual diagnostic.

- We develop efficient execution through deduplicated block–token indexing, fused support construction, and route reuse. Experiments show higher inference speedup and better long-video quality than the evaluated sparse baselines.

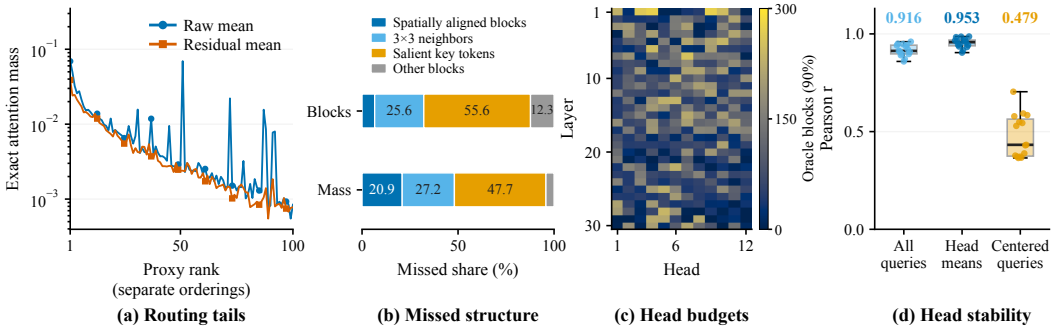


Figure 2: **Structured routing errors and variability in attention budgets.** (a) Exact attention mass versus proxy rank before and after removing protected support; each curve uses its own ranking and the full-context softmax denominator. (b) Structural composition of oracle-selected blocks missed by low-efficiency mean routing, measured by block count and attention mass. (c) Mean oracle budgets required to retain 90% of full-context mass, by layer and head. (d) Budget correlations: individual query-block budgets (All queries), per-head averages (Head means), and query-block budgets after subtracting the corresponding head average (Centered queries).

2 Related Work

Autoregressive video generation. Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have become a core architecture for video generation models such as CogVideoX (Yang et al., 2025c) and Wan (Wan et al., 2025). However, jointly denoising a full clip with bidirectional attention limits streaming. This motivates autoregressive (AR) diffusion, which generates successive chunks conditioned on past context. Diffusion Forcing (Chen et al., 2024) and CausVid (Yin et al., 2024) develop causal diffusion, while Self Forcing (Huang et al., 2025) and LongLive (Yang et al., 2025a) advance rollout training and interactive streaming. Self-attention is a major latency bottleneck in autoregressive video generation, and our method specifically targets this component for acceleration.

Sparse attention. Sparse attention has been widely studied for accelerating bidirectional video DiTs. Existing methods exploit spatial structure and rearrangement (Zhang et al., 2025b; Yang et al., 2025b; Hu et al., 2026), learned scoring (Han et al., 2026), adaptive budgets (Wu et al., 2025; Zhang et al., 2026; Zheng et al., 2026; Zhao et al., 2026; Yehezkel et al., 2026), and routing reuse or approximation (Liu et al., 2026b; Lin et al., 2026; Nordenram & Dittmann, 2026; Taghavi et al., 2026; Li et al., 2026). General sparse-attention methods also study pooled scoring and cumulative selection (Zhang et al., 2025a; Liu et al., 2026a; Lai et al., 2025). More recently, sparse attention has been adapted to AR generation, where cached history evolves across chunks. Light Forcing (Lv et al., 2026) uses hierarchical frame–block selection based on the inexpensive mean-pooling proxy; Sparse Forcing (Xu et al., 2026) learns persistent sparse memory; FAST-AR (Samuel et al., 2026) combines cache compression with sparse retrieval; and Light Interaction (Lu et al., 2026) combines context management, denoising reuse, and sparse attention. In contrast, we identify two recurring structures and protect them before routing, mitigating the risk that mean-pooled proxies overlook concentrated but important interactions.

Context and continuity. Visual continuity is also important for the viewing experience, particularly across autoregressive chunk boundaries. Forcing-KV (Ji et al., 2026), MemRoPE (Kim et al., 2026), and HeadCast (Shen et al., 2026) study how context or cache updates affect temporal continuity. We also address this issue through cross-chunk context sharing and propose PM-BLEx, a boundary-focused perceptual metric based on LPIPS (Zhang et al., 2018). By measuring boundary change above nearby phase-matched temporal variation, PM-BLEx highlights transition artifacts that aggregate quality scores can obscure.

3 OBSERVATIONS

3.1 EXPONENTIAL RESPONSE GAP IN MEAN-POOLED ROUTING

For query block A and key block B, write $z_{ij} = q_i^\top k_j/\sqrt{d}$ and $\bar{z}_{AB} = \mathbb{E}_{AB}[z_{ij}] = \bar{q}_A^\top \bar{k}_B/\sqrt{d}$, where the expectation averages all pairs in $A \times B$. Averaging is inexpensive but can dilute strong matches amplified by exponentiation. If a fraction α of pairs satisfy $z_{ij} - \bar{z}_{AB} \geq t > 0$, then

$$\frac{\mathbb{E}_{AB}[e^{z_{ij}}]}{e^{\bar{z}_{AB}}} = 1 + \mathbb{E}_{AB}\left[e^{z_{ij}-\bar{z}_{AB}} - 1 - (z_{ij} - \bar{z}_{AB})\right] \geq 1 + \alpha(e^t - 1 - t). \tag{1}$$

For $\alpha = 1/64$ and $t = 8$, the lower bound is approximately 47.44. Thus, even a small fraction of strong interactions can be substantially underrepresented by a mean-pooled routing proxy.

We examine the effect of this discrepancy on block selection in Figure 2a, plotting exact key-block attention mass against mean-proxy rank for an illustrative query block. Whole-block mean routing leaves isolated high-mass peaks far down the ranking; the residual curve is smoother after excluding protected structures and recomputing scores and masses. Beyond isolated spikes, Figure 2b shows that oracle-selected blocks missed by mean routing exhibit clear structure: spatially aligned blocks, neighboring (3×3) regions, and distant key-column ridges account for 20.91%, 27.20%, and 47.67% of the missed attention mass, respectively, totaling 95.78%. These results reveal a consistent pattern: **high-mass omissions concentrate in local spatial regions and shared salient key tokens.** This motivates protecting such structures before residual routing.

3.2 STABLE HEAD STRUCTURE, DYNAMIC QUERY DEMAND

To examine routing-budget patterns, we measure the minimum block count retaining 90% of full-context mass under true-mass ranking. Head means range from 1.15 to 274.07 blocks (Figure 2c), making uniform chunk- or layer-wise budgets prone to under-allocation or waste. Across captures, the 360 head means have an average pairwise correlation of 0.953. However, for individual query-block budgets, the correlation falls from 0.916 to 0.479 after subtracting each capture's head means (Figure 2d), revealing that **a stable head-level profile coexists with input-dependent within-head demand.** We therefore use head-level calibration while adapting block counts to each query.

3.3 CONTINUITY BEYOND INDIVIDUAL CHUNKS

Unlike full-clip DiT inference, autoregressive generation uses evolving KV caches across chunks, which sparse attention can further destabilize at chunk boundaries. Figure 3 compares decoded frames from the final query frame Q2 of one chunk and the first query frame Q0 of the next. At three selected boundaries for the same prompt and seed, Light Forcing (Lv et al., 2026) exhibits larger changes in flower size and petal arrangement. This suggests that **sparse autoregressive generation can exhibit abrupt visual changes at chunk boundaries.** Accordingly, we share candidate context across boundaries while preserving adaptive retrieval inside each chunk. Section 6.3 evaluates the policy across prompts and reports its quality trade-offs.

4 RESIDUAL FORCING

4.1 STRUCTURED TOKEN PROTECTION

For an 8×8 query block $A = (r, u)$ at frame r and spatial position u, let $\mathcal{F}_A$ be its candidate frames and $\mathcal{C}_A$ their key tokens. All queries in A share local and salient-token protection (Figure 4); partial boundary blocks contain only valid tokens.

Local spatial support. We retain the spatially corresponding key block in every candidate frame, together with a narrow exterior halo (Figure 4b). Let $B_{f,u}$ be the aligned key block in frame f, and let $\partial_4 B_{f,u}$ contain its four exterior, one-token-wide axial strips. Define $\mathcal{P}_A^{\text{align}} = \bigcup_{f\in\mathcal{F}_A} B_{f,u}$ and $\mathcal{P}_A^{\text{halo}} = \bigcup_{f\in\mathcal{F}_A} \partial_4 B_{f,u}$. Their union is $\mathcal{P}_A^{\text{loc}} = \mathcal{P}_A^{\text{align}} \cup \mathcal{P}_A^{\text{halo}}$.



Figure 3: **Temporal continuity across autoregressive chunk boundaries.** Qualitative comparison of Light Forcing and Residual Forcing at three selected transitions. Each pair shows decoded frames associated with the previous chunk's Q2 and the next chunk's Q0, separated by 0.25 seconds; orange boxes mark corresponding image crops.

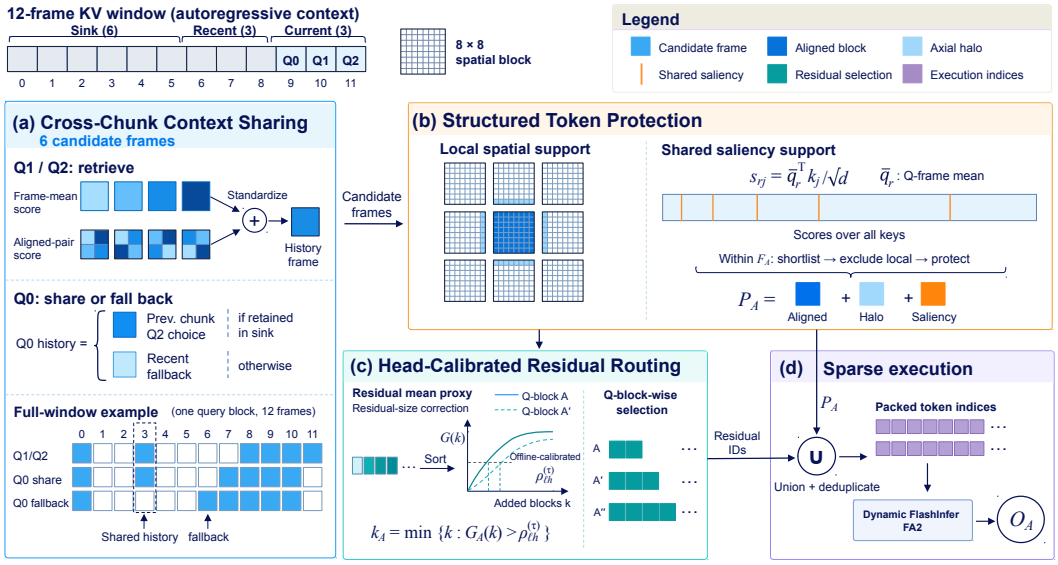


Figure 4: **Residual Forcing overview.** (a) Six candidate frames are selected from a 12-frame cache; Q0 shares retained history from the previous Q2 or uses a recent-frame fallback. (b) Aligned blocks, axial halos, and shared salient keys form the protected support. (c) A head-calibrated cumulative threshold determines the residual prefix for each query block. (d) Deduplicated indices expose original K/V to a dynamic sparse executor with joint softmax normalization.

Shared saliency support. To retrieve narrow columns of high response, we retain individual-key resolution while sharing query pooling across a frame. Let $\mathcal{Q}_r$ be its query tokens and $\bar{q}_r$ their mean. We score

$$s_{rj} = \frac{\bar{q}_r^\top k_j}{\sqrt{d}} = \frac{1}{|\mathcal{Q}_r|} \sum_{i \in \mathcal{Q}_r} \frac{q_i^\top k_j}{\sqrt{d}}. \tag{2}$$

Related pooled-query retrieval appears in VecAttention (Liu et al., 2026a); here it supplies protected tokens before residual block routing. Given shortlist budget k_{cand} and protection budget $k_{\text{prot}} \leq k_{\text{cand}}$, we form $\mathcal{T}_A = \text{Top}_{k_{\text{cand}}}\{s_{rj} : j \in \mathcal{C}_A\}$ and $\mathcal{P}_A^{\text{sal}} = \text{Top}_{k_{\text{prot}}}\{s_{rj} : j \in \mathcal{T}_A \setminus \mathcal{P}_A^{\text{loc}}\}$. Scores and per-frame shortlists are shared across spatial blocks. The protected set is $\mathcal{P}_A = \mathcal{P}_A^{\text{loc}} \cup \mathcal{P}_A^{\text{sal}}$. Aligned blocks remain whole, while halo and saliency indices are packed into compact token lists. Retaining their source blocks in full would waste computation on unprotected positions.

4.2 Head-Calibrated Residual Routing

Routing over unprotected support. Given the protected support $\mathcal{P}_A$, we route only over the remaining tokens. Partition candidate frames into spatial key blocks B_b. The residual $R_{A,b} = B_b \setminus \mathcal{P}_A$ contains their unprotected tokens. For nonempty residual blocks, the corrected mean proxy is

$$u_{A,b} = \frac{\bar{q}_A^\top \bar{k}_{R_{A,b}}}{\sqrt{d}} + \log |R_{A,b}|, \qquad \pi_{A,b} = \frac{\exp(u_{A,b})}{\sum_{v:R_{A,v}\neq\varnothing} \exp(u_{A,v})}. \tag{3}$$

Here $\pi_{A,b}$ is the normalized proxy weight of logit $u_{A,b}$. We sort nonempty residual blocks by decreasing $\pi_{A,b}$. The first k blocks define support $\mathcal{S}_A(k) = \mathcal{P}_A \cup \bigcup_{v=1}^{k} R_{A,(v)}$ and cumulative proxy mass $G_A(k) = \sum_{v=1}^{k} \pi_{A,(v)}$, where (v) denotes rank.

Head-calibrated adaptive budgets. Offline, we find the shortest residual prefix whose union with protected tokens retains a target fraction τ of candidate-frame dense attention mass. Averaging its critical proxy-CDF thresholds yields a fixed per-layer, per-head threshold $\rho_{\ell h}^{(\tau)}$ (Appendix A.3). At inference, this threshold adapts the budget to each query:

$$k_A = \min\{k \geq 1 : G_A(k) > \rho_{\ell h}^{(\tau)}\}, \qquad \mathcal{S}_A = \mathcal{P}_A^{\mathrm{loc}} \cup \mathcal{P}_A^{\mathrm{sal}} \cup \bigcup_{v=1}^{k_A} R_{A,(v)}. \tag{4}$$

For query block A, k_A is the number of selected residual blocks, and $\mathcal{S}_A$ is the final key support including protected tokens.

4.3 Cross-Chunk Context Sharing

Sharing history at boundaries. We maintain a 12-frame KV cache, while each query attends to only 6 selected frames. Each 3-frame chunk (Q0–Q2) uses 5 base frames plus 1 historical frame per query block. To stabilize transitions, Q0 inherits the retrieval used by the preceding Q2 for the same spatial block, layer, and head if the retrieved frame remains in the 6-frame persistent prefix of the 12-frame cache; otherwise, it uses the most recent cached frame outside the base set. Q1 and Q2 retrieve from current features (frame identities and startup: Appendix A.4).

Retrieval with local interaction evidence. For this retrieval, we augment the mean-pooled frame score with spatially aligned query–key interactions. Over historical candidates $\mathcal{H}_A$ outside the base,

$$\widehat{f}_A = \arg\max_{f\in\mathcal{H}_A} \left[\mathsf{z}(\bar{q}_A^\top \bar{k}_f) + \mathsf{z}\left(\frac{1}{|A|} \sum_{i\in A} q_i^\top k_{f,\mathrm{xy}(i)} \right) \right], \tag{5}$$

where $\bar{k}_f$ is the frame-mean key, $\mathrm{xy}(i)$ is the query's spatial position, and z standardizes each score family across historical frames.

Measuring boundary discontinuity. To distinguish boundary artifacts from ordinary temporal variation (Kim et al., 2026; Ji et al., 2026; Shen et al., 2026), we define **Phase-Matched Boundary LPIPS Excess (PM-BLEx)**. Let e_b be the peak adjacent-frame LPIPS (Zhang et al., 2018) in a short window centered at boundary b, and m_b the median peak over identical windows at fixed non-boundary offsets. We measure

$$\mathrm{PM\text{-}BLEx} = \frac{1}{|\mathcal{B}|} \sum_{b\in\mathcal{B}} \frac{[e_b - m_b]_+}{\max(\bar{d}, 10^{-12})}, \tag{6}$$

where $\mathcal{B}$ contains valid post-startup boundaries and $\bar{d}$ is mean adjacent-frame LPIPS. PM-BLEx discounts local non-boundary variation and normalizes by overall temporal change to emphasize boundary artifacts. Appendix A.5 specifies the full protocol.

4.4 Efficient Implementation

We pack each query-block–head pair as an attention row and expose deduplicated protected and residual tokens through compact page tables, accessing original K/V without per-query copies.

Table 1: **Generation quality at three durations.**

Setting	Method	Efficiency Metrics		Continuity	General Metrics ↑							
		K_{eq} ↓	Speedup↑	PM-BLEx↓	Image quality	Subject cons.	Backgr. cons.	Motion smooth.	Aesthetic quality	Dynamic degree	Semantic	Total
LongLive 5-second	Dense FA2	229.82	1.00×	0.445	69.23	97.32	96.06	98.70	66.20	31.94	80.67	82.27
	Light Forcing	32.06	1.22×	1.026	68.80	96.60	95.83	98.65	65.73	48.61	80.44	82.92
	DFSAttn	37.48	1.10×	1.379	69.01	96.93	95.88	98.78	65.55	36.11	80.51	82.34
	SpargeAttention	37.70	1.09×	1.347	68.71	96.45	95.66	98.61	65.60	62.50	80.71	**83.71**
	Residual Forcing	34.98	**1.36**×	**0.568**	68.83	96.76	95.81	98.70	65.81	44.44	80.49	82.73
LongLive 30-second	Dense FA2	281.80	1.00×	0.359	69.47	97.46	96.34	98.75	59.56	39.37	70.74	79.47
	Light Forcing	29.26	1.28×	0.544	65.70	95.71	95.12	98.57	58.05	44.69	71.25	78.68
	DFSAttn	32.53	1.18×	1.199	67.86	96.79	95.78	98.58	59.23	35.68	72.26	79.01
	SpargeAttention	32.81	1.17×	0.765	66.07	95.88	95.25	98.52	58.55	50.52	71.91	79.22
	Residual Forcing	30.08	**1.41**×	**0.284**	68.23	97.20	96.01	98.75	59.05	41.51	70.59	**79.31**
LongLive 60-second	Dense FA2	287.08	1.00×	0.451	68.97	97.05	95.97	98.53	59.60	42.71	70.88	79.33
	Light Forcing	28.96	1.32×	0.520	65.08	95.11	94.67	98.49	57.49	43.02	71.42	78.23
	DFSAttn	32.02	1.17×	0.816	67.31	96.13	95.33	98.33	59.17	38.33	72.56	78.75
	SpargeAttention	32.31	1.16×	0.619	65.75	95.32	94.89	98.38	58.32	51.67	72.03	78.99
	Residual Forcing	29.59	**1.48**×	**0.438**	67.65	96.86	95.73	98.61	59.32	40.70	71.20	**79.13**

FlashInfer's FlashAttention-2 backend (Ye et al., 2025; Dao, 2023) evaluates this union with a single softmax. To reduce routing overhead, we share saliency scores and shortlists across spatial blocks and fuse residual-score correction and masking in Triton, followed by index compaction. We further reuse CUDA Graphs, execution plans, and workspaces. Within each chunk, the first denoising call constructs the route; three subsequent denoising calls and one context write reuse its indices while recomputing attention from current Q/K/V. Section 5.3 reports routing, execution, and full-video latency.

5 Experiments

5.1 Experimental Settings

Setup and evaluation. We evaluate on LongLive (Yang et al., 2025a) with the Wan2.1-T2V-1.3B backbone (Wan et al., 2025), generating single-prompt videos at 832×480 resolution and 16 fps without training. We use 946 VBench prompts for 5-second videos (Huang et al., 2023) and 128 refined MovieGen prompts for 30- and 60-second videos (Polyak et al., 2024). All methods share the same pretrained weights, prompts, and fixed noise initialization (seed 42). We evaluate generation quality with VBench, boundary continuity with PM-BLEx, and inference efficiency with DiT-stage speedup and K_{eq}, the average number of 64-token key blocks attended per query token after deduplication.

Baselines and implementation. We compare dense FlashAttention-2 (Dao, 2023), Light Forcing (Lv et al., 2026), DFSAttn (Hu et al., 2026), and SpargeAttention (Zhang et al., 2025a) on the same autoregressive generator. We use 8×8 blocks with a 12-frame KV cache, 6 candidate frames, a target mass of $\tau = 0.8$ and one routing update per chunk. All sparse methods use dense attention for the first chunk and enable sparse attention from the second chunk onward. We keep K_{eq} comparable across sparse methods and align implementations as closely as possible; further details are in Appendix A.

Table 2: **Component-level inference speed on A100.** RF denotes Residual Forcing; "every call" and "per chunk" specify route-refresh frequency; Dense FA2-6 denotes FA2 with 6 candidate frames.

	Routing	Attention		DiT		End-to-end	
Method	ms ↓	ms ↓	× ↑	s ↓	× ↑	s ↓	× ↑
Dense FA2	0.00	96.74 ± 0.68	1.00	49.13 ± 0.12	1.00	72.72 ± 0.18	1.00
Dense FA2-6	0.00	50.81 ± 0.32	1.90	38.93 ± 0.09	1.26	62.30 ± 0.11	1.17
Light Forcing	27.07	41.79 ± 3.65	2.31	38.42 ± 0.81	1.28	61.96 ± 0.79	1.17
DFSAttn	31.63	66.83 ± 0.44	1.45	43.21 ± 0.25	1.14	67.84 ± 0.63	1.07
SpargeAttention	44.38	67.52 ± 0.14	1.43	43.49 ± 0.26	1.13	67.73 ± 0.54	1.07
RF-every call	27.55	45.31 ± 0.18	2.14	38.96 ± 0.14	1.26	62.48 ± 0.12	1.16
RF-per chunk	**5.51**	$\mathbf{23.45} \pm 0.18$	**4.12**	$\mathbf{34.91} \pm 0.20$	**1.41**	$\mathbf{58.41} \pm 0.25$	**1.25**

Table 3: **Progressive structure protection on 128 thirty-second videos.** Each row uses its own calibration. $K_{\text{eq}}^{\text{ref}}$ includes protected and residual tokens after deduplication.

	Generation quality ↑					Cost / boundary ↓		
Protection	Qual.	Sem.	Imag.	Dyn.	Total	$K_{\text{eq}}^{\text{ref}}$	Attn. ms	PM-BLEx
Mean only	79.39	71.47	61.66	**59.64**	77.81	33.50	22.19	**0.235**
+ alignment	80.82	**71.58**	66.71	40.05	78.97	33.20	**21.94**	0.295
+ axial halo	81.09	71.30	**68.44**	34.58	79.14	33.06	22.75	0.289
+ saliency (full)	**81.50**	70.59	68.23	41.51	**79.31**	**30.01**	23.45	0.284

5.2 Generation Quality

Table 1 shows that Residual Forcing preserves overall generation quality while achieving the lowest PM-BLEx among sparse methods at every duration. Its advantage is clearer on longer videos: at 30 and 60 seconds, it leads the sparse baselines in Total, image quality, subject and background consistency, and motion smoothness. It also improves boundary continuity, reducing PM-BLEx by 44.6% and 47.8% relative to Light Forcing at 5 and 30 seconds, respectively. Overall, Residual Forcing combines near-dense generation quality with stronger temporal continuity than the evaluated sparse baselines.

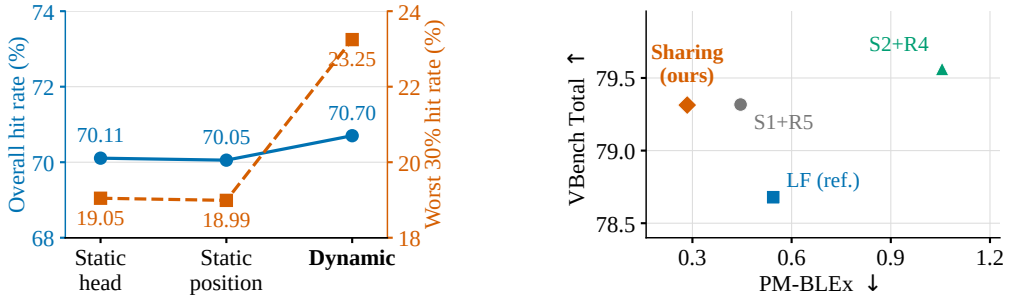
5.3 Inference Speed

Table 1 shows that Residual Forcing achieves the largest DiT speedup among sparse methods at all three durations: $1.36\times$, $1.41\times$, and $1.48\times$, respectively. Table 2 further separates routing, attention, DiT, and end-to-end costs, with Dense FA2-6 and RF-every call as context-reduction and route-refresh controls. This breakdown shows that routing is a substantial cost in the evaluated sparse methods and can outweigh attention execution, limiting the benefit of sparsity alone. RF-per chunk amortizes this overhead across denoising and context-update calls, improving both attention and generation latency. This highlights the value of route reuse alongside faster sparse execution. Appendix A.2 provides the timing protocol.

6 Ablations and Analysis

6.1 The Role of Structure Preservation

To assess the contribution of structure protection, we progressively add aligned blocks, axial halos, and shared salient keys, holding context selection, cache size, $\tau = .8$, and reuse fixed while calibrating each variant separately. On 128 30-second videos, full protection improves Total by 1.51 points and imaging quality by 6.57 points over mean routing, while reducing fixed-snapshot K_{eq} by 10.4% (Table 3). Alignment provides the largest quality recovery, whereas saliency yields the largest reduction in K_{eq}, suggesting complementary roles in preserving local interactions and con-



(a) Budget allocation

(b) Context sharing

Figure 5: **Budget allocation and context sharing.** (a) Overall and worst-30% hit rates for static and dynamic budget allocation methods. (b) Mean boundary excess versus VBench Total on 128 30-second videos. S1+R5 and S2+R4 use fixed candidates; LF is an external reference.

centrating the residual budget. However, retrieval and support construction offset the latency benefit of a smaller support. Together, these results support protection for better quality with fewer attended tokens, while emphasizing that support size and boundary excess alone do not establish speed or quality gains.

6.2 Head-Calibrated Query-Adaptive Budgets

To test whether adapting block counts to individual queries improves attention coverage, we compare dynamic budgets with counts fixed per head or query position, testing the same samples with the remaining routing components unchanged. A query-block evaluation counts as a hit when it retains at least 80% of candidate attention mass. Besides the overall hit rate, we measure each method's worst-30% mean hit rate. Figure 5(a) shows a modest overall increase, from 70.05–70.11% for the static baselines to 70.70%, but a larger tail improvement, from 18.99–19.05% to 23.25%. This contrast indicates that the query adaptation budget has an advantage over the static budget, and the advantage is mainly concentrated in the blocks that frequently fail to meet the coverage target.

6.3 Comparing Cross-Chunk Context Policies

For boundary continuity, we further evaluate the effectiveness of our cross-chunk context sharing and compare it with fixed Sink1+Recent5 (S1+R5) and Sink2+Recent4 (S2+R4) candidates, keeping the 12-frame cache, protection, thresholds, and reuse unchanged. Under the same cache capacity and candidate count, our sharing policy reduces PM-BLEx by 36.2% and 73.1% relative to S1+R5 and S2+R4, respectively, while maintaining similar Total scores (Figure 5(b)). S2+R4 attains a slightly higher Total (79.56) but substantially larger boundary excess, illustrating that aggregate quality can obscure local discontinuities. Sharing also lowers dynamic degree, indicating a trade-off between continuity and motion. Overall, our sharing policy substantially improves boundary consistency while preserving overall generation quality.

7 Conclusion

We proposed Residual Forcing, a training-free sparse-attention framework for autoregressive video diffusion. It protects structured local and shared-saliency support before residual routing, adapts computation with head-calibrated query-wise budgets, and shares context across chunk boundaries. Efficient indexed execution and route reuse translate these designs into practical inference speedups. Experiments show competitive long-video aggregate quality, improved boundary continuity, and consistent acceleration over the evaluated sparse baselines.

Reproducibility Statement

The support definitions and online selection rule are given in Section 4. Appendix A specifies generation, baseline adaptations, calibration, and metric settings. Section 5.3 separates fixed-cache profiling from full-video timing.

Ethics Statement

Residual Forcing accelerates a pretrained video generator and inherits its potential biases and capacity to produce misleading or harmful synthetic media. Faster inference may lower the cost of both beneficial and abusive uses. The present work studies computational efficiency and temporal behavior; it does not establish content authenticity, fairness, or safety improvements. Any public release should preserve applicable model and data terms and clearly identify generated media.

AI Use Statement

We used OpenAI Codex to assist with method development, code implementation, experimental design and execution, result analysis, literature searches, and mathematical derivations. Codex also assisted with manuscript drafting, translation and language editing, figure and table preparation, and LaTeX formatting. The authors checked the code through testing, cross-checked experimental results and statistical analyses, and reviewed the mathematical derivations. The authors take full responsibility for the final content of this work, including AI-assisted text, code, figures, claims, and results.

References

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion, 2024. URL https://arxiv.org/abs/2407.01392.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Shihao Han, Hao Yang, Xinting Hu, Xiaofeng Mei, Yi Jiang, and Xiaojuan Qi. Veda: Scalable Video Diffusion via Distilled Sparse Attention, 2026. URL https://arxiv.org/abs/2605.30325.

Jie Hu, Zixiang Gao, Yutong He, and Kun Yuan. DFSAttn: Dynamic Fine-grained Sparse Attention for Efficient Video Generation, 2026. URL https://arxiv.org/abs/2605.23445.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion, 2025. URL https://arxiv.org/abs/2506.08009.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models, 2023. URL https://arxiv.org/abs/2311.17982.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-KV: Hybrid KV Cache Compression for Efficient Autoregressive Video Diffusion Models, 2026. URL https://arxiv.org/abs/2605.09681.

Youngrae Kim, Qixin Hu, C. C. Jay Kuo, and Peter A. Beerel. MemRoPE: Training-Free Infinite Video Generation via Evolving Memory Tokens, 2026. URL https://arxiv.org/abs/2603.12513.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A Context-Aware Sparse Attention Mechanism for Efficient Long-Sequence Inference, 2025. URL https://arxiv.org/abs/2502.20766.

Haopeng Li, Yitong Li, Junsong Chen, Tian Ye, Haozhe Liu, Jincheng Yu, Duomin Wang, Ruihua Zhang, Zeke Xie, Enze Xie, and Song Han. Sol-Attn: Accelerating Video Generation Inference via On-the-Fly Attention Sparsification, 2026. URL https://arxiv.org/abs/2607.24027.

Shangran Lin, Lu Lu, Jian Chen, and Qiang Liu. RAPID: Reusing Attention Sparsity with Inter-step Adaptation for Efficient Video Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Lin_RAPID_Reusing_Attention_Sparsity_with_Inter-step_Adaptation_for_Efficient_Video_CVPR_2026_paper.pdf.

Anmin Liu, Ruixuan Yang, Huiqiang Jiang, Bin Lin, Minmin Sun, Yong Li, Chen Zhang, and Tao Xie. VecAttention: Vector-wise Sparse Attention for Accelerating Long Context Inference, 2026a. URL https://arxiv.org/abs/2603.29494.

Enhuai Liu, Yunke Wang, Yutong Wang, Changming Sun, and Chang Xu. LoSA: Near-Lossless Sparse Attention for Training-Free Video Diffusion Acceleration, 2026b. URL https://arxiv.org/abs/2608.12032.

Jiacheng Lu, Haoyi Zhu, Sipei Yi, Enze Xie, Yu Li, and Cheng Zhuo. Light Interaction: Training-Free Inference Acceleration for Interactive Video World Models, 2026. URL https://arxiv.org/abs/2605.31158.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light Forcing: Accelerating Autoregressive Video Diffusion via Sparse Attention, 2026. URL https://arxiv.org/abs/2602.04789.

Kasper Nordenram and Amelie Dittmann. ClusterAttention: A training-free speedup of bidirectional attention, 2026. URL https://arxiv.org/abs/2608.26965.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

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, 2024. URL https://arxiv.org/abs/2410.13720.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast Autoregressive Video Diffusion and World Models with Temporal Cache Compression and Sparse Attention, 2026. URL https://arxiv.org/abs/2602.01801.

Jinliang Shen, Lianghao Su, Zheming Li, Kang He, ZiLiang Lai, Yanbing Jiang, and Chengru Song. HeadCast: Casting Attention Heads for Efficient Autoregressive Video Generation, 2026. URL https://arxiv.org/abs/2607.20125.

Xuan Shen, Chenxia Han, Yufa Zhou, Yanyue Xie, Yifan Gong, Quanyi Wang, Yiwei Wang, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. DraftAttention: Fast Video Diffusion via Low-Resolution Attention Guidance, 2025. URL https://arxiv.org/abs/2505.14708.

Pardis Taghavi, Reza Langari, and Gaurav Pandey. Partition the Support, Reconstruct the Residual: Training-Free Sparse Attention for Video Generation and World Models, 2026. URL https://arxiv.org/abs/2608.18484.

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, 2025. URL https://arxiv.org/abs/2503.20314.

Jianzong Wu, Liang Hou, Haotian Yang, Xin Tao, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. VMoBA: Mixture-of-Block Attention for Video Diffusion Models, 2025. URL https://arxiv.org/abs/2506.23858.

Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and Adaptive Sparse Attention for Efficient Long Video Generation, 2025. URL https://arxiv.org/abs/2502.21079.

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse Forcing: Native Trainable Sparse Attention for Real-time Autoregressive Diffusion Video Generation, 2026. URL https://arxiv.org/abs/2604.21221.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block Sparse Attention with Antidiagonal Scoring, 2025. URL https://arxiv.org/abs/2503.16428.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation, 2025a. URL https://arxiv.org/abs/2509.22622.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate Video Generation with Sparse Attention via Semantic-Aware Permutation, 2025b. URL https://arxiv.org/abs/2505.18875.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *International Conference on Learning Representations*, 2025c. URL https://arxiv.org/abs/2408.06072.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. FlashInfer: Efficient and Customizable Attention Engine for LLM Inference Serving, 2025. URL https://arxiv.org/abs/2501.01005.

Shai Yehezkel, Shahar Yadin, Noam Elata, Yaron Ostrovsky-Berman, and Bahjat Kawar. Accelerating Text-to-Video Generation with Calibrated Sparse Attention, 2026. URL https://arxiv.org/abs/2603.05503.

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, 2024. URL https://arxiv.org/abs/2412.07772.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and Training-free Sparse Attention Accelerating Any Model Inference, 2025a. URL https://arxiv.org/abs/2502.18137.

Jintao Zhang, Kai Jiang, Chendong Xiang, Weiqi Feng, Yuezhou Hu, Haocheng Xi, Jianfei Chen, and Jun Zhu. SpargeAttention2: Trainable Sparse Attention via Hybrid Top-k+Top-p Masking and Distillation Fine-Tuning, 2026. URL https://arxiv.org/abs/2602.13515.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast Video Generation with Sliding Tile Attention, 2025b. URL https://arxiv.org/abs/2502.04507.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, 2018. URL https://arxiv.org/abs/1801.03924.

Qi Zhao, Qirui Li, Hanlin Tang, Yiduo Li, Zhen Guo, Cuifeng Shen, Chao Xu, Zhaosheng Chi, Xiaojin Lu, Kan Liu, Tao Lan, Lin Qu, and Xi Li. SCOPE: Subspace Clustering with Online Per-Head Top-K Estimation for Sparse Video Attention, 2026. URL https://arxiv.org/abs/2608.12780.

Xuzhe Zheng, Yuexiao Ma, Jing Xu, Xiawu Zheng, Rongrong Ji, and Fei Chao. HEART: Exploiting Head Heterogeneity in Sparse Attention for Video Diffusion, 2026. URL https://arxiv.org/abs/2605.14513.

A Experimental and Implementation Details

A.1 Generation and Evaluation Protocols

All methods use the same LongLive weights, prompts, and seed-42 noise initialization, generating 832×480 videos at 16 fps. Each chunk contains three latent frames, four denoising calls, and one context write. The 5/30/60-second settings generate 21/123/243 latent frames and retain 80/480/960 decoded frames, respectively, encoded as H.264 with CRF 18.

Evaluation uses 946 VBench prompts for 5-second videos and 128 refined MovieGen prompts for 30- and 60-second videos. Reported VBench scores are scaled to 0–100. Short-video evaluation uses the official dimension-specific prompt subsets and all nine Semantic dimensions; long-video evaluation uses all 128 prompts, with Semantic computed from overall consistency and temporal style. Both protocols use seven quality dimensions and combine normalized Quality and Semantic scores with weights 4:1; their aggregate scores are not directly comparable. PM-BLEx is evaluated separately.

A.2 Baseline Adaptations and Cache Policies

We adapt attention components to the common generator. In the table below, prefix+recent denotes the physical cache, including current noisy frames in the recent portion; candidate frames are a selected subset.

Table 4: **Deployed baseline configurations.** Counts are latent frames.

Method	Physical cache	Routing configuration
Dense FA2	$6 + 6$	Full attention over the readable cache.
Light Forcing	$1 + 11$	Native hierarchical frame–block routing, sparsity $s = .9$.
DFSAttn	$3 + 9$	First two and last four candidate frames; retention schedule $.3/.3/.2/.2/.1$.
SpargeAttention	$3 + 9$	First two and last four candidate frames; TopK $= .2$.
Residual Forcing	$6 + 6$	Six candidates per query block; conditional mass target $\tau = .8$.

Block partitioning and selection. Each baseline retains its own block layout and selection rule. Light Forcing uses contiguous 64-token Q/K blocks, hierarchical frame–block mean scoring, and a chunk-dependent TopK budget. Because 1,560 tokens per frame is not divisible by 64, its adapted frame selector uses exact frame means and admits blocks overlapping selected frames. We retain its cache and sparsity schedule without repeating its original post-training. DFSAttn uses 3D Hilbert ordering, scores 16-token tiles, and aggregates their softmax weights into 128-token Q/K blocks for TopK selection. Its five retention ratios cover four denoising calls and one context write; masks are refreshed at every call. SpargeAttention uses contiguous Q128/K64 blocks, mean-similarity TopK routing with a similarity safeguard, and native Q/K quantization on SM80. In contrast, Residual Forcing uses spatial 8×8 blocks, explicitly protects aligned blocks and individual halo/salient tokens, and selects a query-adaptive residual prefix. Thus, the methods differ in block geometry, scoring, and budget allocation; the reported K_{eq} compares their actual token support rather than native block counts.

Effective support. Table 1 measures K_{eq} on complete MovieGen prompt-ID1 trajectories at seed 42. We divide deduplicated valid key-token counts by 64 and average per query token over all chunks, layers, heads, and calls, including the initial dense chunk and excluding padding. This is a single-prompt diagnostic. All methods use this counting convention; SpargeAttention counts QK support before PV skipping. Budgets are comparable, not exactly equal. Table 3 instead uses a fixed full-window snapshot.

Timing protocol. Measurements use A100-PCIE-40GB GPUs, PyTorch 2.5.1, and CUDA 12.4. Table 1 reports DiT speedups over corresponding same-card Dense runs. Each method generates MovieGen IDs 0–4 in one process; ID0 is discarded as warm-up. Video timings average the remaining four prompts, with sample standard deviations in Table 2. DiT includes denoising, context

writes, scheduling, and cache operations; end-to-end timing additionally includes text processing and VAE decoding, excluding model loading, encoding, and file writing.

Attention profiling in Table 2 covers all 30 layers on two fixed full-window captures, with three warm-ups and eleven timed repetitions each. Combined latency includes routing, metadata, and attention execution. Routing reports median cost per forward. For Residual Forcing, this is estimated from the difference between fresh-route and cached-route medians; only RF-per chunk divides this cost by five. Independently measured component medians need not sum to the reported mean combined latency.

All speedups in Table 2 use the displayed Dense row as their common reference, computed from unrounded mean latencies. Measurements were collected across separate runs on A100-PCIE-40GB GPUs; this normalization differs from the matched-reference convention in Table 1. Dense FA2-6 uses full attention over a physical Sink2+Recent4 cache. These profiling measurements are separate from quality evaluation and trajectory-wide support counting.

A.3 Residual Configuration and Calibration

Each latent frame has a 30×52 grid (1,560 tokens), partitioned into 4×7 spatial blocks of at most 8×8 tokens. Shared-saliency retrieval uses $k_{\text{cand}} = 128$ and $k_{\text{prot}} = 64$; indexed attention uses FlashInfer. Calibration uses six prompts from the extended MovieGen set (IDs 997–1002), seed 42, and chunk indices 4/10/20/50/60/80 (both IDs and indices are zero-based): 36 first-denoising-call captures from dense trajectories under the deployed cache and candidate policy. Each protection variant is calibrated separately. For $\tau = .8$, the frozen array contains 360 FP32 thresholds, one per layer–head, fitted independently of the evaluation prompts.

For a token set T, define $\mu_A(T) = (1/|A|) \sum_{i \in A} \sum_{j \in T} p_{ij}$, where p_{ij} denotes dense attention probabilities over the full readable context. The target is a fraction of the six candidate frames' mass, attained jointly by protected and selected residual tokens. For J_A nonempty residual blocks, set $\mathcal{S}_A(0) = \mathcal{P}_A$ and $G_A(0) = 0$. The shortest qualifying proxy-ordered prefix is

$$k_A^*(\tau) = \min \{k \in \{0, \dots, J_A\} : \mu_A(\mathcal{S}_A(k)) \geq \tau \mu_A(\mathcal{C}_A)\}. \tag{7}$$

The implementation allows an absolute mass tolerance of 10^{-6} in this comparison. Its critical threshold is $\rho_A^*(\tau) = G_A(\max\{k_A^* - 1, 0\})$. We average first over query blocks $\mathcal{A}_n$ within capture n, then equally over captures:

$$\rho_{\ell h}^{(\tau)} = \frac{1}{N_{\text{cap}}} \sum_{n=1}^{N_{\text{cap}}} \frac{1}{|\mathcal{A}_n|} \sum_{A \in \mathcal{A}_n} \rho_{n,\ell,h,A}^*(\tau). \tag{8}$$

Online selection uses strict $>$, selecting at least one block when available, capped at J_A; empty residual sets use zero. Averaging thresholds provides a head prior, not a per-query coverage guarantee, including for later calls that reuse the route.

A.4 Frame Identities and Startup

All frame indices below are zero-based. At the full window, slots 0–5 are persistent sink frames, 6–8 recent history, and 9–11 the current chunk (Q0–Q2). Q1/Q2 use base $\{0, 8, 9, 10, 11\}$ and retrieve from $\{1, \dots, 7\}$. For the same spatial block, layer, and head, Q0 uses base $\{0, 7, 8, 9, 10\}$ and inherits the previous Q2 retrieval if it was in persistent slots 1–5; otherwise, it adds slot 6. Inheritance tracks physical frames, not reused sliding-cache slot numbers. Both retrieval score families are standardized across historical candidates using population standard deviation with a 10^{-6} floor and equal weights.

Growing from nine to twelve frames evicts no history, so inherited frames retain their original indices.

Table 5: **Startup schedule.** Each sparse chunk constructs one route and reuses it for four subsequent calls.

Readable frames	Attention	Candidate policy
3	Dense	All three frames.
6	Block–token sparse	All six frames.
9	Block–token sparse	Q0: $\{0, 3, 4, 5, 6, 7\}$; Q1/Q2: $\{0, 5, 6, 7, 8\}$ plus retrieval from 1–4.
12	Block–token sparse	Full-window policy above.

A.5 Complete PM-BLEx Protocol

We use LPIPS-Alex v0.1 on frames resized to width 256 with aspect ratio preserved. With zero-based decoded indices, define $d_t = \mathrm{LPIPS}(I_{t-1}, I_t)$. Boundaries start at index 9 with period 12; exclude the first four. For each remaining boundary b,

$$e_b = \max_{|u|\le 1} d_{b+u}, \qquad m_b = \mathrm{median}_{o\in\{-8,-6,-4,4,6,8\}} \left(\max_{|u|\le 1} d_{b+o+u} \right). \tag{9}$$

Include only boundaries whose required windows are all valid. Equation 6 normalizes by mean adjacent-frame LPIPS over the entire video, floored at 10^{-12}. Control windows occupy fixed non-boundary phases of the generation cycle. The 5/30/60-second videos contain 2/35/75 valid evaluated boundaries, respectively. Final scores average videos equally, independently of VBench aggregation.