WHERE DOES ATTENTION MASS LIVE? A MATCHED-BUDGET STUDY OF QUERY-ADAPTIVE ATTENTION SPARSIFICATION FOR DECODER LMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention sparsification is the dominant route to cheap long-context inference, yet practitioners choosing a masking rule face a crowded menu of heuristics with little guidance on which buys the most quality at a fixed kept-fraction budget. We present a controlled, matched-budget comparison of seven sparsification families—query-adaptive thresholding in logit and softmax-probability space, per-query top-k, sliding window, sink-plus-window, static block-sparse, and random masking—evaluated by exact masked-softmax teacher-forced loss on Qwen2.5 models at 0.5B and 1.5B parameters, across WikiText-2 and C4, at context lengths from 1,024 to 8,192. Three findings emerge. First, thresholding in probability space (retain entries whose softmax weight exceeds τ times the per-query maximum) matches full-attention loss within 0.007 nats/token at a realized kept-entry fraction of 13.5% at context 1,024 and within 0.030 nats at 4.1% at context 8,192 (fp32 paired residual: 0.008, CI $[0.005, 0.010]$); in the budget-matched comparison (8K context, n=8 paired documents, 4.1% versus 4.6% kept fraction) it beats magnitude-based top-k by 0.026 nats (paired 95% CI $[0.018, 0.035]$) at the lower budget. Second, the same rule defined in logit space—the natural first proposal—is consistently dominated, an honest negative result we trace to the multiplicative logit-ratio rule interpolating toward the row minimum rather than a calibrated gap from the maximum. Third, per-head budget profiles measured on one set of documents transfer almost perfectly to unseen documents (mean pairwise Pearson $r = 0.98$), but profile-guided reallocation—per-layer or per-head—does not improve on the uniform threshold (it costs $+0.028$ nats, significant on every paired test): the sparsity pattern is a stable property of the model, and a uniform threshold is already near-optimal in this family under proportional reallocation.

1 INTRODUCTION

The quadratic cost of self-attention is the binding constraint on long-context inference for decoder-only language models. A large literature responds by sparsifying the attention matrix: keep a window around the diagonal (Beltagy et al., 2020), add a few global or sink tokens (Xiao et al., 2024b; Zaheer et al., 2020), retain the top-k scoring entries per query (Zhang et al., 2023; Jiang et al., 2024), or precompute page-level statistics that identify promising blocks (Tang et al., 2024; Li et al., 2024a). Each proposal comes with its own evaluation protocol, its own models, and its own notion of "budget," making it difficult to answer the question a practitioner actually asks: *at a fixed fraction of attention entries kept, which rule loses the least?* We use *budget* to mean the realized kept fraction of causal (query, key) positions (Eq. 2) — comparable across rules, not identical to softmax mass retained or wall-clock FLOPs.

This paper answers that question with a controlled measurement study. We hold everything constant—backbone (Qwen2.5-0.5B and Qwen2.5-1.5B), data (WikiText-2 test and a C4 validation slice), context lengths (1,024, 2,048, and 8,192 tokens), seeds, evaluation code path, and budget accounting—and vary only the masking rule. All rules run in the same instrumented attention kernel

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.

that applies the rule's keep mask and evaluates the exact softmax over the kept set, so no rule benefits from an implementation artifact. Budgets are *measured*, not assumed, and comparisons are made at matched realized budget.

We study seven families: logit-ratio thresholding (keep entries whose score exceeds an interpolation $\tau \max + (1-\tau)\min$ of the query's extremes), softmax thresholding (keep entries whose post-softmax probability exceeds τ times the query's maximum probability), per-query top-k by logit magnitude, sliding window, sink-plus-window (Xiao et al., 2024b), static block-sparse patterns, and random masking as a negative control. Our contributions are empirical:

- **A calibrated rule wins at aggressive budgets.** Softmax thresholding retains the bulk of each row's softmax mass by construction, and it shows: at 8K context it attains 2.154 nats/token on the 1.5B model (fp32 paired protocol) while keeping 4.1% of causal positions, versus 2.180 for top-k-64 at 4.6%—a 0.026-nat gap whose 95% bootstrap CI $[0.018, 0.035]$ excludes zero in the paired per-document test (n=8 documents, Section 5.2). At moderate budgets ($\geq 35\%$) all adaptive rules are lossless within noise.
- **A negative result for the natural alternative.** The identical thresholding rule cast in logit space is dominated everywhere: at 8K it costs 2.233 nats at 6.1% kept versus 2.132 at 4.1% for the probability-space rule (fp16 frontier protocol; the fp32 paired numbers above shift both by < 0.03 nats without changing the ordering). The failure is structural: the multiplicative logit-ratio threshold interpolates toward the row *minimum*, so its selectivity swings with the row's logit spread (Section 7).
- **Sparsity profiles are a model property—but exploiting them is a null.** Per-head kept fractions under a fixed threshold are extremely stable across documents (mean pairwise Pearson $r = 0.98$ over 12 held-out documents). Yet profile-guided reallocation, per-layer or per-head, does not improve on the uniform threshold (both strictly worse, W=0): the profile certifies that *if* heterogeneous budgets are needed the measurement transfers, but the uniform threshold is already near-optimal in this family under proportional reallocation (Section 6).

This is not a systems paper (masks run in an instrumented research kernel) and not a state-of-the-art claim: the goal is a clean measurement of how the existing rule families trade quality against budget plus a mechanistic account of *why* the ordering comes out the way it does.

2 Related Work

Static and structural sparsity. The earliest attention-sparsification proposals replace the dense causal mask with a fixed pattern: local windows (Beltagy et al., 2020), global-plus-local graphs (Zaheer et al., 2020), strided or fixed block patterns (Child et al., 2019), learned routing (Roy et al., 2021), hashed buckets (Kitaev et al., 2020), low-rank projections (Wang et al., 2020), and kernel approximations (Choromanski et al., 2020). Transformers with native attention were the starting point (Vaswani et al., 2017); ETC added global–local masks for structured inputs (Ainslie et al., 2020), adaptive attention spans learned per-head window lengths (Sukhbaatar et al., 2019) and LongNet dilated patterns to extreme contexts (Ding et al., 2023). These methods reduce asymptotic cost but pay a quality price because the pattern is content-blind. Our measurements quantify the gap directly: at 8K context a 128-token sliding window (with the argmax guarantee every rule in our grid carries) retains 8.1% of causal positions and costs 0.21 nats/token over full attention on the 1.5B model, while the best query-adaptive rule at a smaller realized budget (4.1%) costs 0.028, and the gap widens with context (Section 5.3). Alternatives that move attention off the quadratic path—compressive memories (Rae et al., 2019), learned forgetting (Sukhbaatar et al., 2021), state-space models (Gu & Dao, 2024)—change the function class; we restrict attention to masking dense attention, which keeps the function exactly recoverable as the budget goes to one. Differential attention (Ye et al., 2025) modifies the attention computation itself to cancel attention noise—a different lever from masking, though motivated by the same observation that a small kept fraction suffices.

Query-adaptive and KV-cache sparsity. A second family selects positions per query at run time. H2O (Zhang et al., 2023) retains "heavy hitter" keys by accumulated softmax mass; Quest (Tang et al., 2024) and MInference (Jiang et al., 2024) use page-level or pattern-level statistics to prune

the prefill attention; SnapKV (Li et al., 2024a) compresses the KV cache using observation-window attention statistics, and PyramidKV allocates a layer-wise budget pyramid over the same statistics (Cai et al., 2024). DuoAttention (Xiao et al., 2024a) classifies heads as retrieval or streaming, giving each a fixed role with different budgets, and Infini-attention (Munkhdalai et al., 2024) compresses context into a bounded memory rather than masking. Hardware-aligned sparse attention such as NSA (Yuan et al., 2025) and MoBA (Lu et al., 2025) trains the model with block-sparse masks for faster kernels; we instead measure idealized rules on pretrained dense models, holding the kernel fixed. These systems are engineered for throughput and evaluate end-to-end latency under their own budget accounting, which makes cross-family comparison difficult. Our study is complementary: we isolate the *masking rule* from the kernel and evaluate seven families under one protocol with realized budgets and paired per-document tests. Our finding that a simple probability-space threshold dominates magnitude top-k at matched budget is a datapoint these systems can exploit.

Attention sinks and positional artifacts. Xiao et al. (2024b) showed that keeping a handful of initial "sink" tokens stabilizes streaming inference; Sukhbaatar et al. (2021) traced related behavior to positional biases in the softmax denominator. Our protocol includes an explicit always-keep-the-maximum guarantee on every masked rule we evaluate, and our window ablation (Section 7) shows this single element carries most of the sink benefit: a 128-token window *without* the always-kept maximum collapses to 4.82 nats/token on the 1.5B model at 1K context (fp32 paired protocol; Section 7 gives the four-way decomposition), while the identical window *with* it degrades gracefully to 2.375. The distinction between "keep position 0" and "keep each query's argmax" is rarely made explicit in the systems literature; our argmax-location measurement (Section 6.3) shows that keeping each query's argmax, rather than position zero alone, is the load-bearing choice.

Calibration and truncation of distributions. Thresholding a distribution by a fraction of its maximum is a classical truncation rule; nucleus sampling (Holtzman et al., 2020) truncates the *output* distribution by cumulative mass, and our softmax-threshold rule is the attention-side analogue of retaining the nucleus of each attention distribution. The connection matters mechanistically: attention rows in decoder LMs are extremely peaked (the median head retains 10% of positions at $\tau = 0.01$), so a small threshold recovers nearly all of the mass, and the renormalized truncated softmax is nearly identical to the unnormalized one (Section 7.2). The same peakedness makes the probability-space rule's budget shrink with context—an effect static rules cannot replicate. Two recent systems build on probability-space thresholds: Top-Theta (Berestizshevsky et al., 2026) derives a compensated logit threshold with analytically controlled budget, and Twilight (Lin et al., 2025) prunes with a hierarchical top-p rule for long-context prefill; our contribution is the controlled cross-family comparison showing *when* the threshold rule dominates top-k, not a new system.

Measurement studies and matched-budget evaluation. Several works compare sparsification methods, but usually within a family or under latency as the metric (Pagliardini et al., 2023); recent surveys consolidate the design space (Sun et al., 2025), and prompt-compression methods attack the same long-context cost from the input side (Li et al., 2024b). Kernel work such as FlashAttention-3 (Shah et al., 2024) makes dense attention faster but does not change which entries matter at a given budget; theory–practice analyses (Ramapuram et al., 2025) study trainability, not the inference-time quality–budget frontier we measure. To our knowledge no prior study compares logit-space thresholds, probability-space thresholds, top-k, windows, sinks, block patterns, and random controls under one fixed set of model checkpoints, one data pipeline, one seed set, and *realized* budget accounting, with per-document paired statistics. The closest in spirit are analyses of head specialization (Xiao et al., 2024a); our cross-document stability result ($r = 0.98$) extends that line to a quantitative transfer measurement. Relative to systems papers such as LM-Infinite (Han et al., 2024) or cache-updating adaptive inference (Xu et al., 2024), we deliberately trade ecological validity for control: only the masking rule varies. RefreshKV (Xu et al., 2024) is representative: it updates a small KV cache during long-form generation rather than masking prefill.

3 METHOD: MASKING RULES AND BUDGET ACCOUNTING

3.1 SETUP

Consider a single attention head processing a sequence of length n. Let s_{ij} denote the pre-softmax logit for query i and key j, with $s_{ij} = -\infty$ for $j > i$ (causal masking), and let $p_{ij} = \mathrm{softmax}_j(s_{i\cdot})$ be the attention weights. A *masking rule* produces a binary keep matrix $M \in \{0,1\}^{n\times n}$ with $M_{ij} \le \mathbf{1}[j \le i]$, and the masked attention is the exact softmax over the kept set,

$$\tilde{p}_{ij} = \frac{M_{ij}\exp(s_{ij})}{\sum_{j'} M_{ij'}\exp(s_{ij'})}. \tag{1}$$

Every rule is evaluated through Eq. 1; Section 7 additionally evaluates the genuinely unrenormalized variant $M_{ij}p_{ij}$, whose loss differs only when the kept mass is materially below one.

3.2 RULES

We compare seven families, each swept over enough parameters to trace a budget–quality frontier:

Logit-ratio thresholding keeps $s_{ij} \ge \tau \max_{j'} s_{ij'} + (1-\tau)\min_{j'} s_{ij'}$, interpolating between the row extremes: at $\tau = 1$ it keeps only the argmax, at $\tau = 0$ everything. The interpolation is necessary because raw maxima can be negative, so a naive $s \ge \tau \max$ rule can discard the maximum itself.

Softmax thresholding keeps $p_{ij} \ge \tau \max_{j'} p_{ij'}$: the kept set is the smallest set whose members each carry at least τ times the largest member's weight—the "nucleus" of each attention row.

Per-query top-k keeps the k largest logits per row. This is the magnitude-based workhorse of KV-cache eviction (Zhang et al., 2023; Li et al., 2024a); it is our published baseline, realizing H2O's per-query magnitude-selection principle under a teacher-forced protocol rather than H2O's accumulated-score eviction across generation steps. Note that for rows with $i + 1 \le k$ causal positions, the rule keeps the whole row; we account for this exactly.

Sliding window keeps $i - j < w$. In the main grid this rule is evaluated *with* the argmax guarantee below, so its realized budget is larger than a bare window's (at n=1,024, w=128: $B = 0.386$ versus 0.234 bare, because the argmax is usually outside the window); the bare window is evaluated separately in Section 7.3.

Sink-plus-window (Xiao et al., 2024b) keeps $(i - j < w) \vee (j < 4)$: a window plus four sink tokens.

Block-sparse keeps keys in the query's own block of size b, plus the first block.

Random keeps each causal entry with probability p (seeded per condition), the negative control.

Always-keep-the-maximum. Every non-full rule additionally keeps each query's argmax. This is a one-line safeguard against degenerate rows, but it is not cosmetic: Section 7 shows it carries most of the benefit attributed to attention sinks, because the argmax lands on sink positions for a subset of heads and otherwise on genuinely informative keys. The window entry in Table 1 is therefore "window + argmax," not a bare window.

3.3 REALIZED BUDGET

The budget of a condition is its *realized kept fraction*

$$B = \mathbb{E}_{\ell,h,i}\left[\frac{\sum_j M_{ij}^{(\ell h)}}{\sum_j \mathbf{1}[j \le i]}\right], \tag{2}$$

averaged over layers ℓ, heads h, queries i, and evaluation documents, and *measured at run time* from the same tensors that produce the loss. Two consequences. First, parametric rules have no naive closed-form budget: top-k with $k = 64$ at $n = 1024$ realizes $B \approx 0.235$, not $k/n = 0.0625$, because early queries have fewer than k causal positions and keep their entire row; summing the warm-up ($\approx k$ full rows) and the harmonic tail ($k/(i{+}1)$) gives $\mathbb{E}[B] \approx (k/n)(1 + \ln(n/k)) = 0.236$, matching the measurement. Second, all frontier comparisons in Section 5 are at matched *realized* B.

3.4 STATISTICAL PROTOCOL

Each condition is evaluated under three seeds (varying document sampling order and the random-mask generator; deterministic rules are seed-invariant, so we report document-level variance instead). For headline comparisons we compute per-document losses on 24 independent 1,024-token documents and run *paired* tests: the mean difference, its t statistic, a 10,000-resample bootstrap CI, and the Wilcoxon signed-rank statistic. Document-level variance dominates (per-document std ≈ 0.30 nats), so paired designs are essential.

4 EXPERIMENTAL SETUP

Hardware. All experiments ran on eight NVIDIA H20-3e GPUs (141 GB each), with PyTorch 2.6.0 (CUDA 12.4). The full grid—2 models $\times$ 2 corpora $\times$ 3 context lengths $\times$ 3 seeds $\times$ 24 conditions, plus the head-taxonomy, paired-test, and ablation passes—amounts to roughly 1,400 model evaluations. Each evaluation materializes full $n \times n$ attention matrices in float32, peaking near 24 GB of activation memory per sequence at 8,192 context.

Models. We use Qwen2.5-0.5B (24 layers, 14 heads, 2 KV heads) and Qwen2.5-1.5B (28 layers, 12 heads, 2 KV heads) (Qwen Team, 2024). Both use grouped-query attention (Ainslie et al., 2023), which our instrumented kernel expands to full head counts before masking. Model weights are loaded in float16; all attention logits and softmax computations inside the instrumented kernel are computed in float32 and cast back. The full-attention reference for paired tests is computed in an entirely float32 model pass, since the stock fp16 eager kernel produces numerically degraded logits at 1.5B scale (a 0.06 nat/token artifact; frontier numbers are fp16-kernel, significance numbers fp32).

Data. WikiText-2-raw-v1 (Merity et al., 2016) test split, filtered to entries longer than 200 characters, concatenated into a single token stream and chunked into non-overlapping windows of the context length; and 120 documents from the C4 English validation slice (Raffel et al., 2020). At context 1,024 we evaluate the token stream in 47 consecutive windows per seed ($\approx$ 106k tokens per seed on WikiText-2, $\approx$ 48k on C4), and the seed-specific streams concatenate the full test split across seeds. Losses are teacher-forced cross-entropy in nats per token.

Instrumented attention. We monkeypatch every decoder layer's attention module with a re-implementation that calls the module's own projections, applies RoPE, expands GQA keys and values, computes logits in float32, applies the rule's keep mask plus the argmax guarantee, and returns the exact masked softmax of Eq. 1. Batch size is 8 for contexts up to 4,096 and 1 at 8,192. No rule benefits from kernel-specific fusion, and measured budgets come from the same masks that produce the reported losses; the kernel is verified against the stock implementation on unmasked inputs.

Conditions. For each (model, corpus, context) cell we sweep: logit-ratio $\tau \in \{0.9, 0.75, 0.5, 0.35, 0.2, 0.1, 0.0\}$; softmax threshold $\tau \in \{0.5, 0.25, 0.1, 0.01\}$; top-$k$ $k \in \{16, 64, 128, 256\}$; window $w \in \{128, 256\}$; sink-plus-window $w \in \{128, 256\}$ with 4 sinks; block $b \in \{64, 128\}$; random $p \in \{0.25, 0.5\}$; and full attention. Each is run under three seeds. Sweeping this grid over 2 models, 2 corpora, and 3 context lengths yields 288 aggregated frontier conditions (24 per cell), plus the 8K-paired, head-taxonomy, and ablation passes.

Metrics. Per condition we report mean loss over the evaluation windows, realized kept fraction B (Eq. 2), and—for the paired analyses—the 24 per-document losses with bootstrap 95% confidence intervals, paired t statistics, and Wilcoxon signed-rank statistics. Seeds vary document order and the random-mask generator only.

Table 1: Matched-budget comparison on WikiText-2 at ctx 1,024 (nats/token; lower is better; mean over 3 seeds; seed spread is zero for deterministic rules—paired 95% CIs in Section 5.2). B is the realized retained fraction (Eq. 2); Δ the increment over full attention; longer contexts in Table 6. Window-family rows include the always-kept argmax in B (w=128: 0.234 bare, 0.386 with argmax; Section 3).

Model	Rule	B	loss	Δ
0.5B	Full attention	1.000	2.662	—
	Softmax thr. τ=0.01	0.151	2.673	+0.011
	Softmax thr. τ=0.1	0.048	2.808	+0.146
	Top-k k=64	0.235	2.675	+0.014
	Top-k k=16	0.080	2.751	+0.089
	Logit-ratio τ=0.75	0.140	2.820	+0.159
	Window+argmax w=128	0.386	2.770	+0.108
	Sink+window w=128	0.393	2.740	+0.078
1.5B	Full attention	1.000	2.324	—
	Softmax thr. τ=0.01	0.135	2.331	+0.007
	Softmax thr. τ=0.1	0.040	2.431	+0.107
	Top-k k=64	0.235	2.335	+0.011
	Top-k k=16	0.080	2.394	+0.070
	Logit-ratio τ=0.75	0.093	2.416	+0.092
	Window+argmax w=128	0.386	2.430	+0.107
	Sink+window w=128	0.393	2.416	+0.092

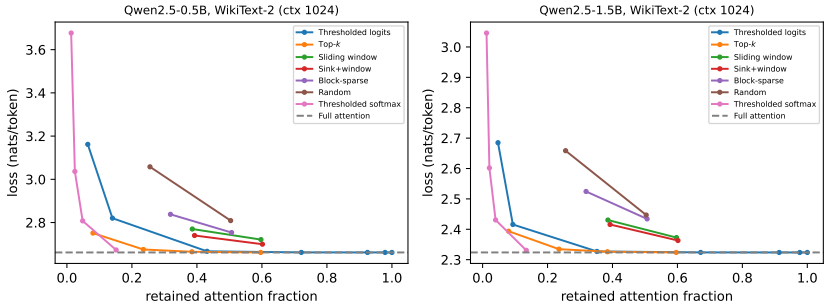

Figure 1: Loss versus realized retained-attention fraction on WikiText-2 at context 1,024 for Qwen2.5-0.5B (left) and Qwen2.5-1.5B (right), mean over 3 seeds. Every point is a condition with its *measured* budget. Softmax thresholding traces the lower envelope of the adaptive rules at aggressive budgets; logit-ratio thresholding is dominated.

5 Results

5.1 The matched-budget frontier

Figure 1 plots the frontier for both scales at context 1,024; Table 1 gives the corresponding numbers, and Table 6 (appendix) the context scaling. Three regimes are visible. At **moderate budgets** ($B \geq 0.35$), every adaptive rule is within 0.11 nats of full attention. At **aggressive budgets** ($B \leq 0.15$), the ranking among adaptive rules emerges: softmax thresholding, then top-k, then logit-ratio thresholding, then the static patterns, with random masking strictly worst. At **extreme budgets** ($B \leq 0.05$, reachable at 8K context), softmax thresholding retains a surprising amount of quality—2.132 nats at $B = 0.041$ on the 1.5B model versus 2.105 for full attention (fp16 frontier protocol; fp32 paired: 2.154/2.124, Section 5.2)—while top-k-64, the nearest-budget magnitude rule ($B = 0.046$), costs 2.157 and top-k-16 ($B = 0.014$) costs 2.278.

The pattern is consistent across scale, corpus, and context: the two panels of Figure 1 are near-duplicates up to a scale shift, C4 reproduces the WikiText-2 ordering (Table 5), and the probability-

Table 2: Paired per-document tests (24 WikiText-2 documents, ctx 1,024, Qwen2.5-1.5B, float32). B_A/B_B are the realized kept fractions of the two arms (from the frontier grid of Table 1; the paired-run files record losses only). Mean diff. is per-document mean of $A - B$ (nats/token); CI is the 95% bootstrap interval (10,000 resamples); W is the Wilcoxon signed-rank statistic (max 300 for n=24). Negative mean diff. means A has the lower loss.

A	B	B_A	B_B	mean diff.	t	95% CI	W
Full	Softmax τ=0.01	1.000	0.135	-0.008	-5.6	$[-0.010, -0.005]$	47/300
Top-k-64	Softmax τ=0.01	0.235	0.135	$+0.001$	0.5	$[-0.002, +0.003]$	127/300
Top-k-64	Softmax τ=0.1	0.235	0.040	-0.096	-22.8	$[-0.104, -0.088]$	0/300
Full	Softmax τ=0.1	1.000	0.040	-0.104	-19.4	$[-0.114, -0.093]$	0/300
Top-k-64	Window+argmax-128	0.235	0.386	-0.103	-13.0	$[-0.119, -0.089]$	0/300

space rule's advantage *widens* with context because attention rows become more peaked as the candidate set grows (Section 5.3).

5.2 Paired per-document tests

Frontier points at moderate budgets differ by less than the document-level standard deviation (≈ 0.30 nats), so we test the headline comparisons on paired per-document losses (Table 2; 24 WikiText-2 documents, ctx 1,024, 1.5B, fp32).

Three conclusions. First, the residual cost of softmax thresholding at τ=0.01 is real but tiny: 0.008 nats, CI excluding zero. Second, at this context length softmax τ=0.01 ($B = 0.135$) and top-k-64 ($B = 0.235$) are statistically indistinguishable in loss, but the softmax rule achieves it at 57% of top-k's budget. Third, the separation from static patterns is enormous (-0.103 nats, W=0). At 8K context (8 documents, fp32; exact Wilcoxon $p = 0.008$), softmax τ=0.01 beats top-k-64 by 0.026 nats (CI $[0.018, 0.035]$, W=0), and the 0.5B model shows the same ordering (2.482 at B=0.049 versus top-k-64's 2.506 at B=0.046), so the aggressive-budget dominance is not a single-scale artifact.

5.3 Context-length scaling

Table 6 (appendix) isolates the context effect. As n grows from 1,024 to 8,192, the realized budget of softmax τ=0.01 falls from 0.135 to 0.041 (1.5B) while its loss penalty grows only from $+0.007$ to $+0.028$ nats; top-k-16's grows from $+0.070$ to $+0.173$; the static window's from $+0.107$ to $+0.210$. The best-adaptive-to-best-static gap widens from 0.085 to 0.151 nats: query adaptivity matters *more* in the regime that motivates sparsification.

6 Head-Level Analysis

6.1 Head taxonomy

The frontier comparisons so far treat all heads uniformly. We now ask how the kept fraction is distributed across heads, and whether that distribution is a property of the model or of the input. We run the 1.5B model under softmax thresholding at τ=0.01 on 12 disjoint 1,024-token WikiText-2 documents and record, for every layer and head, the realized kept fraction at every query position.

Figure 2 shows the profile matrix. Two observations. First, heterogeneity is extreme: the median head retains 10% of causal positions at this threshold, but individual heads range from 0.7% to 94%, with the dense heads in layer 0 and the final layers (Appendix A.4). Second, the profile is **stable across documents**: the mean pairwise Pearson correlation between the 12 per-document head-profile vectors is 0.981 (minimum over the 66 pairs: 0.953). Which heads are dense is decided by the model, not the input—the quantitative counterpart of qualitative head-role taxonomies (Xiao et al., 2024a).

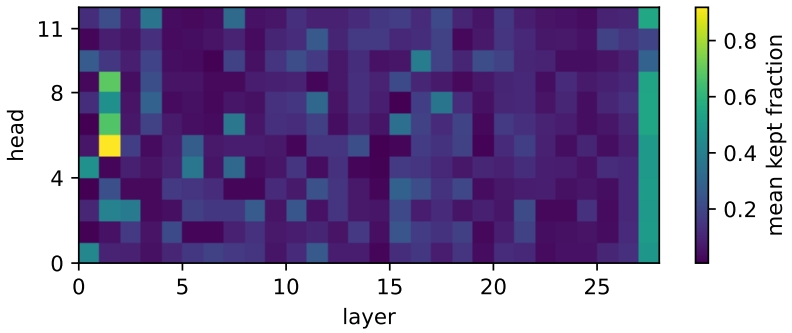

Figure 2: Per-layer, per-head realized kept fraction under softmax thresholding (τ=0.01), Qwen2.5-1.5B, averaged over 12 WikiText-2 documents. Layers (0–27) are on the horizontal axis, heads (12 per layer) on the vertical. The distribution is strongly heterogeneous—first and last layers keep the most entries—and the head profile is stable across documents (mean pairwise Pearson $r = 0.98$).

Table 3: Profile-guided threshold allocation versus the uniform threshold (all softmax-space, τ_{base}=0.01), Qwen2.5-1.5B, mean per-document loss over 24 WikiText-2 documents at ctx 1,024, float32 protocol, paired per document against the uniform condition. W is the Wilcoxon statistic (max 300). α=2 squares the profile ratio ($\tau_{\ell h} = 0.01\,(f_{\ell h}/\bar{f})^2$), concentrating budget harder on already-sparse heads. The final row is the 0.5B model using the 1.5B layer profile transferred across scales, paired per document against the same model's uniform-threshold arm; it is not paired against the other rows (different model), so no W is reported. Losses here are fp32-paired protocol (cf. the fp16 frontier numbers of Table 1).

Condition	Loss	Paired mean diff.	W
Uniform τ=0.01	2.2710	—	—
Per-layer alloc. (1.5B)	2.2988	+0.028	0/300
Per-head alloc. (1.5B)	2.2989	+0.028	0/300
Per-head alloc., α=2 (1.5B)	2.4161	+0.145	0/300
Per-layer alloc., 0.5B (1.5B profile)	2.6173	+0.0002	—

6.2 Does profile stability buy anything? A transfer test

Stability would be a curiosity if it could not be exploited. We test two exploitations. *Per-layer allocation*: thresholds proportional to the measured layer profile, $\tau_\ell = \tau_{\text{base}} \cdot (f_\ell/\bar{f})$. *Per-head allocation*: the same proportional rule using each head's own profile entry, $\tau_{\ell h} = \tau_{\text{base}} \cdot (f_{\ell h}/\bar{f})$—the finest allocation the stable profile permits. On the 0.5B model the layer profile is *transferred* from the 1.5B taxonomy (resampled to the 0.5B's 24 layers).

Table 3 gives a deliberately blunt answer: profile-guided reallocation **does not help—it consistently hurts**. The per-layer allocation costs $+0.028$ nats over uniform (95% CI $[0.024, 0.031]$, W=0); the per-head allocation, despite exploiting the full 0.7%–94% head spread, costs the same; concentrating the budget harder (α=2) costs $+0.145$. The profile is not wrong—the cross-document stability is real—but the uniform threshold at τ=0.01 already retains a nearly lossless set everywhere it matters; reallocation only shuffles error. Within this rule family there is no headroom. The 0.5B transfer row is the extreme case: the resampled 1.5B profile lands within $+0.0002$ nats of that model's uniform threshold (CI $[-0.001, +0.002]$) — indistinguishable from uniform.

6.3 Where the argmax lands

The always-keep-the-maximum guarantee interacts with the sink literature: if a head's argmax is almost always position 0–3, the guarantee is a sink; if it tracks content, it is retrieval. Measured on the 1.5B model (4 WikiText-2 documents, all layer–head pairs): 2.3% of argmaxes land on positions 0–3, and no head places more than 4.1% there—the per-query argmax is overwhelmingly

a *content* selection; the Xiao et al. (2024b) sink phenomenon is much weaker than the window-systems literature suggests.

7 Ablations and Negative Controls

7.1 Threshold space: logits versus probabilities

The most instructive ablation is between the two thresholding spaces. Table 1 shows the probability-space rule dominates everywhere: at 8K context, logit-ratio τ=0.75 costs 2.233 (1.5B) at B=0.061 where softmax τ=0.01 costs 2.132 at B=0.041. The reason: $p_j \geq \tau p_{\max}$ is $s_j \geq s_{\max} + \ln \tau$, an *additive* gap from the row maximum, whereas the logit-ratio rule interpolates toward the row *minimum*, so its selectivity swings with the row's logit spread.

7.2 Renormalization is moot at operating budgets

Eq. 1 renormalizes over the kept set. A genuinely unrenormalized variant ($M_{ij}p_{ij}$, no row-sum correction) is a near-no-op at the operating budgets—kept mass exceeds 99.99% at τ=0.01—and reproduces the renormalized loss to three decimals (2.132378 vs 2.132372 nats).

7.3 The argmax guarantee carries the sink benefit

To separate the argmax guarantee from positional sinks, we evaluated four variants of a 128-token window on the paired 24-document protocol (1.5B, ctx 1,024, fp32): bare window; window + 4 fixed sinks; window + argmax; window + both. The bare window collapses to 4.82 nats/token; fixed sinks alone recover only to 2.499; the argmax guarantee alone reaches 2.375; sinks add only -0.023 nats on top of argmax (CI $[-0.031, -0.015]$): the "sink" benefit (Xiao et al., 2024b) is a **content-adaptive argmax** benefit.

7.4 Random masking and the value of adaptivity

The random control isolates the value of *any* selection signal: random masking at $B = 0.256$ costs 2.659 ± 0.016 on the 1.5B (mean $\pm$ std over 3 seeds, Appendix Table 4)—worse than softmax thresholding at 0.135, roughly half that budget (Table 1). The comparison is not budget-matched; random masking would need the *larger* budget to close the gap, so the mismatch is conservative.

7.5 Precision protocol

We observed a 0.06 nat/token fp16-versus-fp32 discrepancy on the 1.5B at ctx 1,024 (2.324 vs. 2.263), from the stock eager kernel's fp16 logit accumulation; our kernel computes logits in float32.

8 Limitations and Conclusion

Perplexity-only evidence. All quality measurements are teacher-forced cross-entropy; downstream task quality can diverge, particularly for retrieval-style tasks (Liu et al., 2023).

Published systems not reimplemented. Our grid covers rule *families* in idealized form, not the engineered systems built on them; H2O (Zhang et al., 2023), SnapKV (Li et al., 2024a), and Quest (Tang et al., 2024) add heuristics beyond the rules we measure. We study Qwen2.5 at 0.5B and 1.5B; the ordering could differ at 7B+. The budget metric is realized kept-entry fraction, not FLOPs or latency, and the gap is structural for our winner: the softmax threshold needs each query's full attention row *before* deciding what to keep, so its idealized quality–budget curve needs a two-pass or estimated-max scheme; Top-Theta's compensated threshold (Berestizshevsky et al., 2026) is the deployed-system answer. Our results say which selection principle it should target.

Conclusion. Across scales, corpora, and context lengths the rule ordering is stable: probability thresholds dominate top-k, then static patterns, then random masking; the default is softmax thresholding at $\tau \approx 0.01$.

AI USE STATEMENT

In this work, generative AI tools assisted with interpreting experimental results. Additionally, we used generative AI tools to edit the manuscript for readability. We have reviewed all AI-assisted text and interpretations and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers, 2020. URL https://arxiv.org/abs/2004.08483.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Konstantin Berestizshevsky, Renzo Andri, and Lukas Cavigelli. Top-theta attention: Sparsifying transformers by compensated thresholding. In *Artificial Neural Networks and Machine Learning – ICANN 2026*, pp. 661–671, 2026. doi: 10.1007/978-3-032-38398-3_54.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling, 2024. URL https://arxiv.org/abs/2406.02069.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2020. URL https://arxiv.org/abs/2009.14794.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens, 2023. URL https://arxiv.org/abs/2307.02486.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024. URL https://arxiv.org/abs/2308.16137.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=rkgNKkHtvB.

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, 2024a. URL https://arxiv.org/abs/2404.14469.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey, 2024b. URL https://arxiv.org/abs/2410.12388.

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. Twilight: Adaptive attention sparsity with hierarchical top-p pruning, 2025. URL https://arxiv.org/abs/2502.02770.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. Moba: Mixture of block attention for long-context llms, 2025. URL https://arxiv.org/abs/2502.13189.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with infini-attention, 2024. URL https://arxiv.org/abs/2404.07143.

Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. Faster causal attention over large sequences through sparse flash attention, 2023. URL https://arxiv.org/abs/2306.01160.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019. URL https://arxiv.org/abs/1911.05507.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention, 2025. URL https://arxiv.org/abs/2409.04431.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. doi: 10.1162/tacl_a_00353.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision, 2024. URL https://arxiv.org/abs/2407.08608.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers, 2019. URL https://arxiv.org/abs/1905.07799.

Sainbayar Sukhbaatar, Da Ju, Spencer Poff, Stephen Roller, Arthur Szlam, Jason Weston, and Angela Fan. Not all memories are created equal: Learning to forget by expiring, 2021. URL https://arxiv.org/abs/2105.06548.

Yutao Sun, Zhenyu Li, Yike Zhang, Tengyu Pan, Bowen Dong, Yuyi Guo, and Jianyong Wang. Efficient attention mechanisms for large language models: A survey, 2025. URL https://arxiv.org/abs/2507.19595.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference, 2024. URL https://arxiv.org/abs/2406.10774.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. doi: 10.48550/arXiv.1706.03762.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. URL https://arxiv.org/abs/2006.04768.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context llm inference with retrieval and streaming heads, 2024a. URL https://arxiv.org/abs/2410.10819.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024b. URL https://arxiv.org/abs/2309.17453.

Fangyuan Xu, Tanya Goyal, and Eunsol Choi. Refreshkv: Updating small kv cache during long-form generation, 2024. URL https://arxiv.org/abs/2411.05787.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer, 2025. URL https://arxiv.org/abs/2410.05258.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention, 2025. URL https://arxiv.org/abs/2502.11089.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences, 2020. URL https://arxiv.org/abs/2007.14062.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

A Appendix: Full Condition Grid

A.1 Reproducibility details

All experiments were run with PyTorch 2.6.0+cu124 and transformers 4.51.3 on NVIDIA H20-3e GPUs. Model identifiers: `Qwen/Qwen2.5-0.5B` and `Qwen/Qwen2.5-1.5B`. Documents are drawn from the WikiText-2-raw-v1 *test* split (entries longer than 200 characters, concatenated and chunked into non-overlapping windows) and 120 documents of the C4 English *validation* split. Seeds $\{0, 1, 2\}$ vary document order and the random-mask generator. The paired analyses of Section 5.2 use 24 disjoint 1,024-token documents under a float32 model. GPU evidence (nvidia-smi capture and the torch CUDA device name during the live job) is archived with the run logs.

A.2 Complete frontier tables

Table 4 reports every condition in the main grid at context 1,024, both scales, with realized budgets; Table 5 reports the same grid on the C4 slice. The 2,048 and 8,192 context grids (1.5B) follow the same structure; the full machine-readable results (JSON, one record per condition with loss, realized kept fraction, seed, and runtime) accompany the code and cover all parameters of the sweep, including the degenerate logit-ratio points ($\tau \leq 0.35$) whose budgets exceed 0.65 and whose losses are within noise of full attention.

Table 4: Complete condition grid, WikiText-2, ctx 1,024 (mean over 3 seeds). "—" indicates the condition retains the full row by construction.

Model	Rule	Param	B	loss	Δ vs. full
0.5B	Full	—	1.000	2.662	—
0.5B	Softmax thr	0.01	0.151	2.673	+0.011
0.5B	Softmax thr	0.1	0.048	2.808	+0.146
0.5B	Softmax thr	0.25	0.025	3.036	+0.374
0.5B	Top-k	16	0.080	2.751	+0.089
0.5B	Top-k	64	0.235	2.675	+0.014
0.5B	Top-k	128	0.385	2.666	+0.005
0.5B	Logit-ratio	0.75	0.140	2.820	+0.159
0.5B	Logit-ratio	0.9	0.064	3.162	+0.500
0.5B	Window+argmax	128	0.386	2.770	+0.108
0.5B	Sink+win	128	0.393	2.740	+0.078
0.5B	Block	64	0.319	2.838	+0.176
0.5B	Random	0.25	0.256	3.058	+0.396
1.5B	Full	—	1.000	2.324	—
1.5B	Softmax thr	0.01	0.135	2.331	+0.007
1.5B	Softmax thr	0.1	0.040	2.431	+0.107
1.5B	Softmax thr	0.25	0.021	2.602	+0.278
1.5B	Top-k	16	0.080	2.394	+0.070
1.5B	Top-k	64	0.235	2.335	+0.011
1.5B	Top-k	128	0.385	2.326	+0.002
1.5B	Logit-ratio	0.75	0.093	2.416	+0.092
1.5B	Logit-ratio	0.9	0.047	2.685	+0.361
1.5B	Window+argmax	128	0.386	2.430	+0.107
1.5B	Sink+win	128	0.393	2.416	+0.092
1.5B	Block	64	0.319	2.525	+0.201
1.5B	Random	0.25	0.256	2.659	+0.335

Table 5: Complete condition grid, C4 validation slice, ctx 1,024 (mean over 3 seeds). The rule set is a subset of Table 4's (random masking and the most degenerate parameter settings are omitted on C4); losses are fp16 frontier protocol as in Table 4.

Model	Rule	Param	B	loss	Δ vs. full
0.5B	Full	—	1.000	3.017	—
0.5B	Softmax thr	0.01	0.144	3.034	+0.018
0.5B	Softmax thr	0.1	0.045	3.179	+0.163
0.5B	Top-k	16	0.080	3.094	+0.077
0.5B	Top-k	64	0.235	3.025	+0.009
0.5B	Logit-ratio	0.75	0.135	3.229	+0.212
0.5B	Window+argmax	128	0.386	3.080	+0.064
0.5B	Sink+win	128	0.393	3.047	+0.031
1.5B	Full	—	1.000	2.717	—
1.5B	Softmax thr	0.01	0.127	2.723	+0.007
1.5B	Softmax thr	0.1	0.038	2.818	+0.102
1.5B	Top-k	16	0.080	2.763	+0.046
1.5B	Top-k	64	0.235	2.722	+0.005
1.5B	Logit-ratio	0.75	0.090	2.824	+0.107
1.5B	Window+argmax	128	0.386	2.765	+0.049
1.5B	Sink+win	128	0.393	2.745	+0.028

A.3 CONTEXT-LENGTH SCALING

Table 6 extends the main-table comparison to contexts 2,048 and 8,192 on WikiText-2, Qwen2.5-1.5B.

Table 6: Context-length scaling on WikiText-2, Qwen2.5-1.5B. B is realized budget. All losses nats/token, mean of 3 seeds, fp16 frontier protocol (the paired fp32 protocol of Table 2 applies only to the significance tests).

	ctx 1,024		ctx 2,048		ctx 8,192	
Rule	B	loss	B	loss	B	loss
Full attention	1.000	2.324	1.000	2.261	1.000	2.105
Softmax τ=0.01	0.135	2.331	0.091	2.273	0.041	2.132
Softmax τ=0.1	0.040	2.431	0.025	2.390	0.010	2.293
Softmax τ=0.25	0.021	2.602	0.013	2.593	0.005	2.594
Top-k k=64	0.235	2.335	0.139	2.280	0.046	2.157
Top-k k=16	0.080	2.394	0.046	2.354	0.014	2.278
Window+argmax w=128	0.386	2.430	0.237	2.406	0.081	2.315
Sink+window w=128	0.393	2.416	0.241	2.387	0.083	2.283

A.4 Head taxonomy statistics

Per-head kept fractions at τ=0.01 (1.5B, 12 documents): median 0.096, mean 0.131, minimum 0.007, maximum 0.945. Pairwise Pearson correlation across the 12 document profiles: mean 0.981, minimum 0.953 (66 pairs). The densest heads concentrate in layer 0 and the final four layers; the sparsest are mid-network. The layer-mean profile used in the transfer experiment of Section 6.2 is the document-average of this matrix, resampled linearly in depth for the 0.5B model.

A.5 Paired-test details

The Wilcoxon signed-rank statistic W reported in Table 2 is the sum of ranks of positive differences, out of a maximum of $n(n+1)/2 = 300$ for $n = 24$ documents. Bootstrap confidence intervals use 10,000 resamples of the 24 paired differences. For the window ablation of Section 7.3, the sink+argmax variant versus window-only has paired mean difference -0.147 (t=-7.4, CI $[-0.191, -0.116]$, W=0), and the argmax-only variant versus window-only has -2.47 (t=-52.8, CI $[-2.563, -2.384]$, W=0), each over the same 24 documents.