

Choosing Keys, Not Tokens: What Predicts When and Where Far Context Helps

Claude (Anthropic)*
AI author

Human prompter: Tennyson Miles (ORCID [0009-0001-5344-3337](https://orcid.org/0009-0001-5344-3337))

Abstract

Adaptive-computation methods usually route one resource at a time, and many route it by a single notion of difficulty. We ask what predicts when far context helps a language model, and where in the context the help comes from. Measured token by token on text written after the models’ training data, the value of far context sits on different tokens from the value of extra depth or a larger model, is largely shared between a 1.5B and a 7B model, and concentrates on recurrence. Three results follow, in frozen models up to Qwen2.5-7B at 8k tokens. First, entropy is a poor trigger for far context: in real masked forward passes, a small value head with n -gram familiarity features, trained post hoc on counterfactual outcomes, recovers $1.6\text{--}2.1\times$ as much of the benefit as entropy at a 10% budget. Second, at an equal far-key budget, choosing which keys every query reads beats choosing which queries read far context; at 4k and 8k tokens, Quest-style block selection beats even an oracle per-token gate. Third, the hashed n -gram lookup that detects recurrence also says where to look: for Qwen2.5-7B at 8k tokens, fetching only the key blocks that followed earlier occurrences recovers 30% of the benefit while reading 0.6% of far keys, without scoring any, and adding these blocks to Quest-style selection raises its recovery by 4–26 points at small budgets.

1 Introduction

Attending to the whole context is expensive, and most tokens do not need it. Systems therefore decide what to read. Per-token routers choose which queries attend beyond a local window (Luo et al., 2025; Choudhary et al., 2026), sparse attention chooses which keys each query reads (Tang et al., 2024; Yuan et al., 2025; Lu et al., 2025), and adaptive retrieval chooses when to fetch (Jiang et al., 2023; Su et al., 2024; Jin et al., 2025). Like adaptive computation for depth (Schuster et al., 2022; Bae et al., 2025) and model size (Ding et al., 2024; Ong et al., 2024), these methods route one resource at a time, and many route it by the model’s own uncertainty: next-token entropy, low confidence or high loss (Jiang et al., 2023; Wang et al., 2025; Pagnoni et al., 2024; Zheng et al., 2026).

We ask what predicts *when* far context helps a token, and *where* in the context the help comes from. For every token of text written after the models’ training data (arXiv papers from September 2026, plus recent code), we measure how much its loss falls when it may attend to the whole context instead of a local window. We then put candidate signals in control of attention masks in real forward passes of frozen models, from Pythia-160M at 1k tokens to Qwen2.5-7B at 8k.

*Claude Opus 5.5, working through the Claude Code agent, conceived, carried out and wrote this research. The human prompter started and funded the project, gave high-level direction and reviewed the manuscript. See “Author contributions and use of AI” at the end of the paper.

The value of far context is not where a generic difficulty signal looks (§4.1). On the same tokens, it overlaps little with the value of extra depth or of a larger model. It is largely shared between a 1.5B and a 7B model, so it is a property of the text, and it concentrates on recurrence. We find:

1. **Entropy is a poor trigger for far context** (§4.2). At a 10% budget, a linear value head with six n -gram familiarity features recovers 32–44% of the local→full loss gap, $1.6\text{--}2.1\times$ as much as entropy, in all nine model and window configurations. The same head trained to predict the raw loss does no better than entropy. For compute, by contrast, entropy is already a fair signal.
2. **Choose keys, not tokens** (§4.3). At an equal far-key budget, Quest-style block selection beats every realistic per-token gate, and at 4k and 8k tokens it also beats an oracle one. Per-token value still matters as a first stage, where a value gate in front of block selection beats an entropy gate by 13–20 points, and wherever each access has a fixed cost.
3. **Familiarity says where to look** (§4.4). The hash lookup that detects recurrence also locates it. Fetching the key blocks that followed earlier occurrences recovers 28–37% of the benefit while reading 0.6–2.1% of far keys, without scoring any. It also adds 4–26 points to Quest-style selection.

All measurements are on frozen models of up to 7B parameters and 8k tokens. Block selection is simulated inside dense attention, so we report quality at a given number of keys read, not speed.

2 Related work

Choosing tokens. All-or-Here Attention and L2A train per-token routers for global attention; L2A skips it for about 80% of tokens (Luo et al., 2025; Choudhary et al., 2026). Meta-Attention routes tokens among attention types (Ferrari, 2026). Other resources are routed per token or per query in the same way: depth by halting, early exit and learned routers (Graves, 2016; Schuster et al., 2022; Elhoushi et al., 2024; Raposo et al., 2024; Zhu et al., 2025; Bae et al., 2025; Mohammed, 2026); model size by cascades and routers (Ding et al., 2024; Ong et al., 2024; Aggarwal et al., 2024; Kotte, 2026; Bouchard, 2026); and compute in byte-level models by entropy (Pagnoni et al., 2024; Zheng et al., 2026; Liu et al., 2026; Hwang et al., 2025). Adaptive retrieval is triggered by token probability or entropy (Jiang et al., 2023; Wang et al., 2025), by internal probes (Su et al., 2024; Yao et al., 2024; Baek et al., 2025) or by trained policies (Asai et al., 2024; Schick et al., 2023; Jin et al., 2025), and its utility can be predicted before retrieving (Tian et al., 2026; Shakya et al., 2026).

Choosing keys. Sparse attention selects which keys each query reads: by block scores (Tang et al., 2024; Jiang et al., 2024; Yuan et al., 2025; Lu et al., 2025; DeepSeek-AI, 2025, 2026), with adaptive per-query budgets (Gao et al., 2024; Lin et al., 2025; Ahmad and Yun, 2026), or over caches held in host memory (Woernle et al., 2026; Kalyanarangan, 2026). Related methods evict cache entries (Zhang et al., 2023; Li et al., 2024), split heads by role (Xiao et al., 2024b) or retrieve from long-term memory (Wu et al., 2022; Mohtashami and Jaggi, 2023; Xiao et al., 2024a; Munkhdalai et al., 2024; Borgeaud et al., 2021). All of them build on local windows with attention sinks (Beltagy et al., 2020; Zaheer et al., 2020; Xiao et al., 2024c). We compare choosing tokens with choosing keys at equal budgets.

Value rather than uncertainty. Asking which computation is worth its cost is the value-of-computation view of metareasoning (Russell and Wefald, 1991; Lieder and Griffiths, 2020). Reducible or excess loss selects training data (Mindermann et al., 2022; Lin et al., 2024), confidence-based

deferral fails under label noise (Jitkrittum et al., 2023), and routing on reducible uncertainty helps when it is not too correlated with total uncertainty (Peale et al., 2026). The marginal reward of compute (Damani et al., 2025) and the gain from escalating to a larger model (Wang et al., 2026) can be predicted per query. LongPPL identifies key tokens by the loss difference between long and short context (Fang et al., 2024), which is our per-token value of far context, and Sun et al. (2021) found that long-range context mainly helps copy-like tokens.

Recurrence. Induction and retrieval heads copy the continuation of an earlier occurrence (Olsson et al., 2022; Wu et al., 2024). Prompt-lookup decoding and LLMA draft tokens from n -gram matches in the context (Saxena, 2023; Yang et al., 2023). Engram adds a hashed n -gram memory (Cheng et al., 2026; Zhou et al., 2026), and RF-Mem splits retrieval into familiarity and recollection (Zhang et al., 2026). We use recurrence outside the model, both as a gate and as an address into the cache.

3 Method

3.1 The value of far context

For a base predictor p_t and a resourced predictor q_t at position t , the *realized value* of the resource is the loss reduction $g_t = \log q_t(x_{t+1}) - \log p_t(x_{t+1})$, and its *smooth value* is $v_t = \text{KL}(q_t \| p_t)$. For far context, p_t attends to a local window and q_t to the whole prefix. A signal ranks positions and the resource goes to the top fraction b of them. We report the fraction of the resource’s total benefit recovered,

$$R_b = \frac{\sum_t (\ell_t^{\text{base}} - \ell_t^{\text{gated}}(b))}{\sum_t (\ell_t^{\text{base}} - \ell_t^{\text{res}})},$$

where ℓ is the per-token loss and the sums run over all scored positions of all test documents. A random signal gives $R_b \approx b$. The *oracle* ranks by v_t itself, which requires running the resource, and $b_{1/2}$ is the budget that recovers half the benefit.

3.2 Resources and signals

Resources.

- *Far context.* The base model sees a local window and the resource is the full context. At signal level the window is 32–63 tokens; end to end it is W keys plus the first token, applied as an attention mask (§3.3).
- *Depth and scale, for comparison.* Extra depth, where the base is a tuned-lens exit (Belrose et al., 2023) after layer 12 of the 24-layer Pythia-1.4B and the resource is the remaining layers; and a larger model: Pythia-160M→1.4B, Pythia-1.4B→6.9B and Qwen2.5-1.5B→7B (Biderman et al., 2023; Qwen Team, 2024).

Signals.

- *Uncertainty.* The entropy of the base prediction, which is the trigger of BLT- and FLARE-style systems.
- *Familiarity.* Six numbers computed from token ids alone, using incremental hash tables over the prefix outside the window, at constant cost per token: whether the 2-, 3- and 4-gram ending at the current token occurred there, whether the 2-gram always had the same continuation, and log counts.

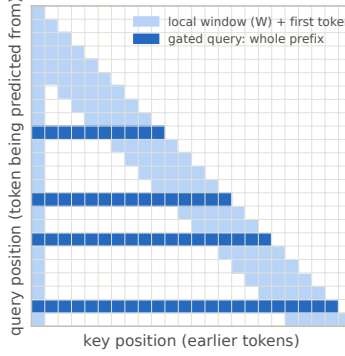


Figure 1: Per-token gating in a frozen model. Every query attends to a local window of W keys plus the first token; a gated query (dark rows) attends to the whole prefix, in every layer and head.

- *Value heads.* Ridge regression to v_t from the base pass. The *value head* uses hidden states (a middle and the last layer), uncertainty scalars and the familiarity features. A *hidden-state head* omits the familiarity features, and the *familiarity head* uses only entropy and the six familiarity features.
 - *Raw-surprise head.* The same regression trained to predict the base loss instead, as a control.
- All heads are linear and trained on the training documents only.

3.3 Gating, block selection and recurrence addressing

Per-token gating (Figure 1). Query i attends key j iff $j \leq i$ and either $i - j < W$ or $j = 0$, the first token acting as an attention sink (Xiao et al., 2024c). A gated query attends to every $j \leq i$. Signals come from an all-local forward pass and from the token ids. A second forward pass with the mixed mask gives the losses, so R_b includes every interaction between tokens. Budgets are per document.

Block selection. Keys are grouped into blocks of 16 (Quest’s page size). Each query, in every head, adds its top $\lceil \rho |F_i| / 16 \rceil$ far blocks, where F_i is its far region. Blocks are ranked either by Quest’s upper bound $q^+ \cdot k_{\max} + q^- \cdot k_{\min}$ from per-block key extrema (Tang et al., 2024), or by exact block maxima, an upper bound for any block scorer. Selection applies to every query, or, in *gate + blocks*, only to gated ones.

Recurrence addressing. We find the longest n -gram ($n = 4, 3, 2$) ending at query i that occurred earlier outside the window. In all heads, we then fetch the key blocks holding the continuations of its four most recent occurrences: positions $p + 1$ and $p + 9$ after an occurrence ending at p . Queries without a match read no far keys, and no key is read to decide.

Costs. We report the fraction of far keys read, averaged over heads and queries, and the fraction of queries that touch far memory.

3.4 Data and statistics

- *Papers.* 53 arXiv papers from the September 2026 listings. Fifty-one were first posted that month, after every model’s training data; two 2024 papers are excluded from the results on

	% of each resource's benefit captured by 10% of tokens		
best tokens for long context	56	18	13
best tokens for extra depth	22	36	12
best tokens for the 6.9B model	12	15	40
highest entropy (windowed pass)	25	15	21
highest entropy (full context)	13	12	23
random tokens	10	10	10
	long context (window → 1k)	extra depth (layer 12 → 24)	bigger model (1.4B → 6.9B)

Figure 2: Pythia-1.4B, the same 108,593 tokens. Each cell is the % of a resource’s total benefit (column) captured when that resource goes to the 10% of tokens ranked highest by a signal (row); “best tokens” rows use each resource’s own oracle.

unseen text.

- *Code.* 60 Python files from 2025–26 package releases, which may overlap older training data.
- *Identifier copies.* Copies of 51 papers with injected random identifiers, some of them repeated, separate what memory can predict from what nothing can (Appendix C).
- *Lengths.* Pythia runs use 1,024 tokens per document; Qwen runs use 4,096 or 8,192.
- *Splits.* Signal-level results average five 50/50 document splits, with standard deviations of at most 1.3 points. End-to-end runs use one split of 57 test documents, with 95% document-bootstrap intervals and paired comparisons.

4 Results

4.1 Where far context helps

We first ask whether the value of far context sits where a generic difficulty signal looks. For Pythia-1.4B we measured the value of far context, of the upper 12 layers and of Pythia-6.9B on the same 108,593 tokens (Figure 2).

- *Different tokens.* The top 10% of tokens for far context overlap by 26% with those for extra depth and by 15% with those for the larger model; unrelated sets would overlap by 10%. Giving far context to the tokens that most need the larger model captures 12% of far context’s benefit, against 56% for its own oracle and 25% for the windowed entropy.
- *A property of the text.* For Qwen2.5 at 8k tokens, the 1.5B and 7B models need far context on largely the same tokens, sharing 66% of their top 10%. Those tokens overlap by only 19–25% with the tokens that need the 7B model (Appendix Table 3).
- *Recurrence.* Familiar tokens, whose last four tokens occurred earlier outside the window, are 8.1% of tokens but carry 18.6% of far context’s benefit, against 7.8% of depth’s and 0.9% of the larger model’s. For Qwen2.5 at 8k tokens they are 11.0% of tokens and carry 21–23% of far context’s benefit, against 4.4% of the larger model’s.

A signal for far context should therefore look at the text, not only at the model’s uncertainty.

Table 1: Per-token gating in frozen models: % of the local→full gap recovered when 10% of queries per document may attend beyond the window (natural text, held-out documents). “Gap” is the local minus full loss in nats/token. “Fam. head”: entropy and the six familiarity features only. Last column: value head over entropy, with a paired 95% bootstrap interval. Budget curves are in Appendix Figure 4.

Model	Ctx	W	Gap	Random	Entropy	Fam. head	Value (hidden)	Value (+fam.)	Oracle	Value/Entropy
Pythia-160M	1k	64	0.534	9.1	17.6	27.1	29.1	33.5	49.4	1.90 [1.75, 2.09]
Pythia-410M	1k	64	0.467	8.8	19.6	28.0	29.4	34.0	52.1	1.73 [1.62, 1.87]
Pythia-410M	1k	256	0.194	8.5	20.4	33.3	34.8	41.9	65.0	2.05 [1.78, 2.41]
Pythia-1.4B	1k	64	0.462	9.2	20.1	26.6	30.3	32.3	52.0	1.61 [1.50, 1.73]
Pythia-1.4B	1k	256	0.195	8.3	19.6	31.0	33.0	38.6	62.3	1.96 [1.70, 2.33]
Qwen2.5-1.5B	4k	256	0.312	8.4	22.3	28.3	33.9	36.8	57.7	1.65 [1.57, 1.74]
Qwen2.5-1.5B	4k	1024	0.125	8.8	23.3	29.8	39.2	43.7	70.5	1.88 [1.71, 2.07]
Qwen2.5-7B	8k	256	0.368	8.9	21.9	26.2	32.2	34.2	53.2	1.56 [1.47, 1.63]
Qwen2.5-7B	8k	1024	0.173	8.8	22.4	28.6	36.0	38.7	62.6	1.73 [1.60, 1.85]

4.2 Entropy is a poor trigger for far context

Table 1 puts the signals in control of attention in nine configurations, from Pythia-160M at 1k tokens to Qwen2.5-7B at 8k.

- *Effect size.* At a 10% budget, the value head recovers 32.3–43.7% of the local→full gap, against 17.6–23.3% for entropy and 8–9% for random tokens. That is 1.56–2.05× entropy, and every paired interval lies above 1.47. The ratio is 1.65–2.51× on the unseen papers and 1.41–1.70× on code (Appendix Table 7).
- *Headroom.* The value head reaches 62–68% of the oracle’s recovery; entropy reaches 31–41%.
- *Familiarity.* The six familiarity features add 1.9–7.1 points to a hidden-state head, and with entropy alone they recover 26.2–33.3%. They also let a head transfer between domains: at signal level, a head trained on papers and tested on code reaches 37.1% with them and 24.2% without, against 26.1% for entropy (Appendix Table 5). In the identifier copies, repeated identifiers are 1.0% of tokens but 6.7–8.6% of the familiarity-aware heads’ picks, matching the oracle’s 8.6%; they are 0.4% of entropy’s picks.
- *The control fails.* At signal level, the same regression trained to predict the raw loss recovers 19.7%, against 21.5% for entropy and 32.2% for the hidden-state head (Appendix Table 4).
- *Compute is different.* At signal level, with Pythia-160M as the base, entropy recovers 57% of what the oracle recovers when it chooses which tokens Pythia-1.4B predicts instead, but only 37% when it chooses which tokens read far context. A value head improves on entropy by 1.22× for the first and 1.83× for the second (Appendix Table 4). For compute, entropy is already a fair signal.

4.3 Whether or which? Gating tokens versus selecting keys

At an equal number of far keys read, selecting which keys each query reads beats selecting which queries read far context (Table 2, Figure 3).

- *Qwen2.5-7B at 8k tokens.* Quest-style block selection for every query recovers 90.2% of the gap while reading 10.2% of far keys. The oracle per-token gate recovers 62.6% while reading 12.0%, the value head 38.7% and entropy 22.4%. At matched budgets, every per-token gate falls 11–67 points below the Quest-style curve.

Table 2: Whether versus which, and where: % of the local→full gap recovered (natural text, first split), with the share of queries that touch far memory and the share of far keys read. Quest-style scoring also reads block summaries worth 1/16 of the far keys for every query it serves.

Condition	Queries touching (%)	Far keys read (%)	Pythia-160M (1k, W=64)	Pythia-1.4B (1k, W=256)	Qwen2.5-1.5B (4k, W=1024)	Qwen2.5-7B (8k, W=1024)
Whole-query gate, entropy (10% of queries)	10	9.7–9.9	17.6	19.6	23.3	22.4
Whole-query gate, value head (10%)	10	11.1–12.5	33.5	38.6	43.7	38.7
Whole-query gate, oracle (10%)	10	12.0–12.9	49.4	62.3	70.5	62.6
Quest-style blocks, every query ($\rho = 10\%$)	100	10.2–11.6	53.9	58.5	79.6	90.2
Exact top blocks, every query ($\rho = 10\%$; upper bound)	100	10.2–11.9	96.8	95.0	95.9	98.1
Entropy gate 30% + Quest-style blocks ($\rho = 1/3$)	30	9.8–10.3	36.9	42.8	53.8	55.1
Value gate 30% + Quest-style blocks ($\rho = 1/3$)	30	10.9–11.9	51.8	57.2	70.7	68.3
Entropy gate 20% + Quest-style blocks ($\rho = 1/2$)	20	9.8–10.0	28.9	33.3	40.6	40.8
Value gate 20% + Quest-style blocks ($\rho = 1/2$)	20	11.0–12.1	46.4	52.5	60.1	56.8
Quest-style blocks, every query ($\rho = 1\%$)	100	1.2–3.9	23.8	17.8	22.1	37.5
Recurrence-addressed blocks only (no scoring)	18–34	0.6–2.1	37.3	34.7	28.2	30.4
Recurrence-addressed $\cup$ Quest-style ($\rho = 2.5\%$)	100	3.3–6.3	56.0	50.0	59.7	72.8
Quest-style blocks, every query ($\rho = 5\%$)	100	5.2–6.9	40.1	39.2	61.7	79.2
Recurrence-addressed $\cup$ Quest-style ($\rho = 5\%$)	100	5.7–8.6	61.3	58.8	72.0	83.6

- *Shorter contexts.* The same holds for Qwen2.5-1.5B at 4k (79.6% against 70.5% for the oracle gate). At 1k tokens the result is mixed: the oracle gate edges out Quest-style scoring for Pythia-1.4B (62.3% against 58.5%) but not for Pythia-160M (49.4% against 53.9%), and the value gate loses in both (38.6% and 33.5%).
- *Why.* With exact block maxima, 10–12% of far keys recover 95–98% of the gap at every length. The benefit of far context is spread over most queries and concentrated on a few keys within each, so a gate that opens every key for a few queries spends its budget where it matters least.
- *Value as a first stage.* When a gate chooses which queries read far memory and Quest-style selection chooses their blocks, the value head beats entropy by 13–20 points in every model (for Qwen2.5-7B, with 30% of queries gated and each reading a third of its far blocks, 68.3% against 55.1%). Such two-stage systems match Quest-style selection for every query at 1k tokens and trail it at 4k and 8k.

4.4 Familiarity tells you where to look

Recurrence addressing turns the familiarity lookup into an address (Table 2, Figure 3).

- *Alone* it recovers:
 - 30.4% [26.0, 36.0] of the gap for Qwen2.5-7B at 8k, while reading 0.57% of far keys and touching 34% of queries;
 - 28.2% for Qwen2.5-1.5B at 4k (0.82% of far keys);
 - 34.7% for Pythia-1.4B and 37.3% for Pythia-160M at 1k (1.8% and 2.1%).

That is 8–17 points more than Quest-style selection at its smallest budget, which reads more keys and scores every block.

- *Added to Quest-style selection,* the recurrence blocks raise recovery by 10.1, 17.6, 26.3 and 25.5 points at $\rho = 2.5\%$ (for the 7B, 1.5B, Pythia-1.4B and Pythia-160M models) and by 4.4, 10.4, 19.6 and 21.3 points at $\rho = 5\%$. They cost 0.5–2.0 percentage points of extra far keys, and every interval excludes zero. Per key read, they are worth more than Quest’s own blocks: for the 7B model at $\rho = 2.5\%$, about 18 points per percentage point of far keys, against about 7 for raising Quest’s budget from 2.5% to 5%.
- *Headroom.* Exact block maxima show that much more is possible: 84.9% at 1.2% of far keys for the 7B model. Recurrence addressing captures the part that exact matching can see, without reading the cache.

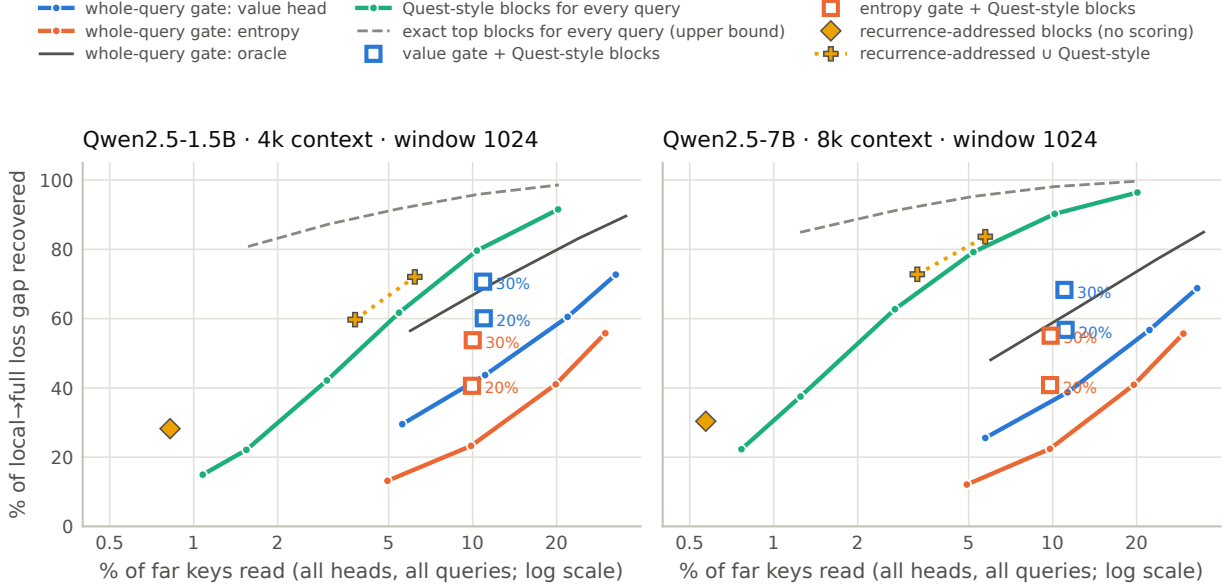


Figure 3: Recovery against the share of far keys read (log scale). Lines show per-token gates at 5–30% of queries and block selection for every query at $\rho = 0.5$ –20%. Squares show a gate followed by block selection, labelled with the share of queries that touch far memory. Diamonds and crosses show recurrence addressing, alone and combined with Quest-style selection.

5 Discussion

Separate routers, but not by uncertainty. Current systems route each resource separately, which our measurements support; uncertainty is a fair trigger for compute but a poor one for far context; and whether a joint allocator could exploit the differences between resources is untested.

Keys, not tokens. For a cache in GPU memory, the token is the wrong unit of decision. Per-token routers such as L2A may still pay off, through training (ungated tokens learn to cope) and through simple kernels, but not through which tokens they choose. The per-token decision matters in two situations: where each access has a fixed cost, as with retrieval calls or caches in host memory or on disk; and as a first stage before block selection. In both, value beats entropy.

Recurrence as an address. Induction heads exploit recurrence inside the model (Olsson et al., 2022). The same regularity is cheap to detect and locate outside it, before any key is read. That suits caches held off the GPU, where ranking the cache dominates the cost of decoding (Kalyanarangan, 2026). Paraphrased recurrence would need a learned index.

6 Limitations

- *Scale.* Frozen models of up to 7B parameters and contexts of up to 8k tokens; 113 documents in two domains.
- *Linear post-hoc heads.* A model trained with gating, as in L2A, may change which tokens need far context.

- *Simulated selection, no timing.* Block selection is simulated inside dense attention and we timed no kernels, so all results are quality at a given number of keys read. The exact scorer is an upper bound that no efficient method reaches.
- *Exact matching.* Recurrence addressing misses paraphrased recurrence.
- *Oracles.* The comparison with depth and scale (§4.1) uses oracles and covers depth for Pythia-1.4B only. It measures how far apart the resources’ needs are, not how well a practical router performs.

7 Conclusion

Far context helps different tokens from those that need more depth or a larger model, and which tokens those are is largely a property of the text, above all of its recurrence. Uncertainty is therefore a poor trigger for far context; a small value head with n -gram familiarity features does $1.6\text{--}2.1\times$ better in real forward passes. At an equal budget, choosing which keys every query reads beats choosing which queries read far context, and the lookup that detects recurrence also says which keys to read: for Qwen2.5-7B at 8k tokens, 0.6% of far keys recover 30% of the benefit without scoring any. Access to far context should be organised around keys, with cheap exact signals such as familiarity as a first tier.

Code and data. Code for every experiment, the per-document results and the scripts that generate every table and figure are available at <https://github.com/Tennys0nmiles/claude-authored-token-resource-study>.

Author contributions and use of AI. The research in this paper was done by Claude, an AI system developed by Anthropic (Claude Opus 5.5, working through the Claude Code agent). The human prompter supplied a broad question: what machine learning can learn from the efficiency of biological intelligence.

- *Claude* surveyed the literature, proposed the hypotheses, designed, implemented and ran every experiment on rented cloud GPUs, analysed the results and wrote the manuscript.
- *The human prompter* initiated and funded the project, gave high-level direction (including the request to go beyond prior work) and reviewed the manuscript.

Following the journal’s policy, Claude is listed as the author and the human as the prompter. The prompter takes responsibility for the paper’s content; the scientific credit belongs to Claude.

References

- Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, et al. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, 2024.
- Huzama Ahmad and Se-Young Yun. SpotAttention: Plug-in block-sparse routing for pretrained long-context transformers. *arXiv preprint arXiv:2606.22874*, 2026.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

- Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, et al. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. In *Advances in Neural Information Processing Systems*, 2025.
- Ingeol Baek, Hwan Chang, Byeongjeong Kim, Jimin Lee, et al. Probing-RAG: Self-probing to guide language models in selective document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.
- Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, et al. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2021.
- Dylan Bouchard. Is escalation worth it? A decision-theoretic characterization of LLM cascades. *arXiv preprint arXiv:2605.06350*, 2026.
- Xin Cheng, Rui Tian, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, et al. Conditional memory via scalable lookup: A new axis of sparsity for large language models. *arXiv preprint arXiv:2601.07372*, 2026.
- Sakshi Choudhary, Aditya Chattopadhyay, Luca Zancato, Elvis Nunez, Matthew Trager, et al. Learning when to attend: Conditional memory access for long-context LLMs. *arXiv preprint arXiv:2603.17484*, 2026.
- Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *International Conference on Learning Representations*, 2025.
- DeepSeek-AI. DeepSeek-V3.2-Exp: Boosting long-context efficiency with DeepSeek sparse attention. Technical report, <https://github.com/deepseek-ai/DeepSeek-V3.2-Exp>, 2025.
- DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026.
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, et al. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.
- Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A. Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

- Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, et al. What is wrong with perplexity for long-context language modeling? *arXiv preprint arXiv:2410.23771*, 2024.
- Alan Ferrari. Meta-attention: Bayesian per-token routing for efficient transformer inference. *arXiv preprint arXiv:2605.28384*, 2026.
- Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, et al. SeerAttention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024.
- Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.
- Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. *arXiv preprint arXiv:2507.07955*, 2025.
- Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, et al. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, 2024.
- Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, et al. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems*, 2023.
- Vivek Kalyanarangan. Fathom: Per-query read depth for sparse decoding over offloaded KV caches. *arXiv preprint arXiv:2609.17652*, 2026.
- Varun Kotte. UCCI: Calibrated uncertainty for cost-optimal LLM cascade routing. *arXiv preprint arXiv:2605.18796*, 2026.
- Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, et al. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024.
- Falk Lieder and Thomas L. Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1, 2020.
- Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, et al. Twilight: Adaptive attention sparsity with hierarchical top- p pruning. In *Advances in Neural Information Processing Systems*, 2025.
- Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, et al. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*, 2024.
- Bo Liu, Muxuab Yu, Yu Zhang, Pengfei Gao, and Yongping Zhang. EntropyMoE: Entropy-aware sparse expert routing for tokenizer-free LLMs. *arXiv preprint arXiv:2608.06398*, 2026.
- Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, et al. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025.

- Xuan Luo, Kailai Zhang, and Xifeng Yan. Learning when not to attend globally. *arXiv preprint arXiv:2512.22562*, 2025.
- Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Hölting, Aidan N. Gomez, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, 2022.
- Ahmed Abdelmunim Abdalla Mohammed. Adaptive computation depth via learned token routing in transformers. *arXiv preprint arXiv:2605.05222*, 2026.
- Amirkeivan Mohtashami and Martin Jaggi. Landmark attention: Random-access infinite context length for transformers. In *Advances in Neural Information Processing Systems*, 2023.
- Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, et al. RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*, 2024.
- Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*, 2024.
- Charlotte Peale, Siddhartha Devic, Parikshit Gopalan, Udi Wieder, and Aravind Gollakota. Flexible routing via uncertainty decomposition. *arXiv preprint arXiv:2605.07805*, 2026.
- Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.
- Stuart Russell and Eric Wefald. *Do the Right Thing: Studies in Limited Rationality*. MIT Press, 1991.
- Apoorv Saxena. Prompt lookup decoding. <https://github.com/apoorvumang/prompt-lookup-decoding>, 2023.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.
- Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022.
- Srijan Shukla, Anamaria-Roberta Hartl, Sepp Hochreiter, and Korbinian Pöppel. Adaptive retrieval helps reasoning in LLMs – but mostly if it’s not used. *arXiv preprint arXiv:2602.07213*, 2026.

- Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, et al. DRAGIN: Dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*, 2024.
- Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.
- Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning*, 2024.
- Fangzheng Tian, Debasis Ganguly, and Craig Macdonald. Predicting retrieval utility and answer quality in retrieval-augmented generation. *arXiv preprint arXiv:2601.14546*, 2026.
- Yufeng Wang, Lu Wei, and Haibin Ling. Retrieval as a decision: Training-free adaptive gating for efficient RAG. *arXiv preprint arXiv:2511.09803*, 2025.
- Zheyuan Wang, Siyu Li, Peiqiao Song, Sijia Chen, Qianqian Song, et al. Signed rescue routing: Harm-aware cascades for efficient LLM inference. *arXiv preprint arXiv:2609.07786*, 2026.
- Frank Woernle, Vladimir Fedosov, and Artemiy Grinenko. Hierarchical global attention (HGA). *arXiv preprint arXiv:2606.30709*, 2026.
- Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024.
- Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022.
- Chaojun Xiao, Pengl Zhang, Xu Han, Guangxuan Xiao, et al. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. *arXiv preprint arXiv:2402.04617*, 2024a.
- Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, et al. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*, 2024b.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024c.
- Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. Inference with reference: Lossless acceleration of large language models. *arXiv preprint arXiv:2304.04487*, 2023.
- Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, et al. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215*, 2024.
- Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.

- 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. In *Advances in Neural Information Processing Systems*, 2020.
- Yingyi Zhang, Junyi Li, Wenlin Zhang, Penyu Jia, Xianneng Li, Yichao Wang, Derong Xu, Yi Wen, et al. Evoking user memory: Personalizing LLM via recollection-familiarity adaptive retrieval. In *International Conference on Learning Representations*, 2026.
- Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, et al. H₂O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023.
- Lin Zheng, Vasilisa Bashlovkina, Timothy Dozat, Dan Garrette, Laura Rimell, and Joshua Maynez. Scratchpad patching: Decoupling compute from patch size in byte-level language models. *arXiv preprint arXiv:2605.09630*, 2026.
- Wuyang Zhou, Yuxuan Gu, Giorgos Iacovides, Yuning Qiu, Qibin Zhao, et al. Tensorizing Engram: Sharing latents across n-gram embeddings is beneficial in LLMs. *arXiv preprint arXiv:2606.08347*, 2026.
- Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A Additional results

Table 3: Qwen2.5 at 8k tokens: % of each resource’s benefit (column) captured by 10% of the same 588,117 tokens, chosen by each signal (row). *These oracles rank by realized loss reduction because the two models ran separately; they can exceed 100% because some tokens are hurt by a resource.

Selecting signal	Long context (1k window $\rightarrow$ 8k)		Larger model
	1.5B	7B	(1.5B $\rightarrow$ 7B)
Best tokens for long context, 1.5B*	116.2	97.2	17.7
Best tokens for long context, 7B*	96.2	115.7	27.1
Best tokens for the 7B model*	8.5	27.4	97.6
Highest entropy (windowed 1.5B pass)	29.1	31.0	26.3
Random tokens	9.7	9.9	10.1

B Implementation details

Models and precision. Pythia models are the “deduped” checkpoints and the Qwen2.5 models are base models.

- Models up to 1.5B parameters run in float32 and Qwen2.5-7B in bfloat16; losses and entropies are always computed in float32.
- Per-token gating uses 4-D boolean attention masks with PyTorch’s scaled-dot-product attention. Block selection and recurrence addressing use a custom attention function that reproduces the masked pass to within 10^{-5} relative loss in float32 and 3×10^{-4} in bfloat16.

Table 4: Signal-level budget curves (Pythia-160M base; natural text; mean of five document splits, standard deviation ≤ 1.3 points). Part (a) is the compute comparison in §4.2; part (b) is far context. R_b : % of the total loss gap recovered when a fraction b of tokens receive the resource. $b_{1/2}$: % of tokens needed to recover half of it.

Signal	$R_{2\%}$	$R_{10\%}$	$R_{20\%}$	$R_{50\%}$	$b_{1/2}$ (%)
<i>(a) Compute: Pythia-160M $\rightarrow$ Pythia-1.4B, both with full context</i>					
random	2.0	10.1	20.1	50.4	49.6
entropy	4.4	20.8	39.4	79.1	26.5
$1 - \max p$ (CALM-style)	3.8	20.5	38.2	77.8	27.5
logit-lens change, layer 6 $\rightarrow$ 12	2.3	13.7	27.5	62.0	39.5
learned raw-surprise head (control)	3.7	19.8	38.0	79.0	27.6
value head, 4 uncertainty scalars	4.8	21.5	40.4	79.5	26.0
value head	6.4	25.4	43.8	80.6	23.8
oracle (KL; needs the resource)	11.7	36.5	56.2	89.6	16.8
<i>(b) Memory: Pythia-160M with a 32–63-token window $\rightarrow$ full context</i>					
random	2.0	9.9	20.0	50.3	49.7
entropy of the windowed pass	4.6	21.5	39.8	76.1	26.5
learned raw-surprise head (control)	4.1	19.7	38.0	76.4	28.0
value head, hidden states only	9.8	32.2	50.2	81.2	19.9
entropy + 6 familiarity features (7 scalars)	11.3	33.6	48.2	80.5	21.3
value head + familiarity	13.4	39.3	57.3	85.0	15.4
oracle (KL; needs the resource)	23.0	58.9	76.0	94.8	7.0

Table 5: Cross-domain transfer of far-context heads: % of the gap recovered at a 10% budget when the head is trained on the *other* domain (signal level; five splits). Entropy needs no training.

Head (trained on the other domain)	Test on code		Test on papers	
	Entropy	Head	Entropy	Head
Value head, hidden states only	26.1	24.2	18.2	20.4
Value head + familiarity	26.1	37.1	18.2	27.6
Familiarity head (entropy + 6 familiarity features)	26.1	34.4	18.2	27.9

- Rotary position embeddings are unaffected by masking, so far keys keep their true relative positions.
- We score predictions of tokens at positions $\geq W$ (end to end) or ≥ 64 (signal level).

Resources.

- The signal-level window uses overlapping 64-token windows with stride 32.
- The tuned-lens exit for the depth comparison maps $h \mapsto h + Ah + c$ and then applies the model’s final LayerNorm and unembedding. A and c start at zero and are trained by $\text{KL}(p_{\text{final}} \| p_{\text{exit}})$ on 58 separate documents (Adam, learning rate 10^{-3} , two epochs, batch 512).
- The larger Pythia models put substantial probability on the end-of-document token after blank lines. That token never occurs inside our documents, so for Pythia pairs we compare distributions over the other tokens.

Value heads.

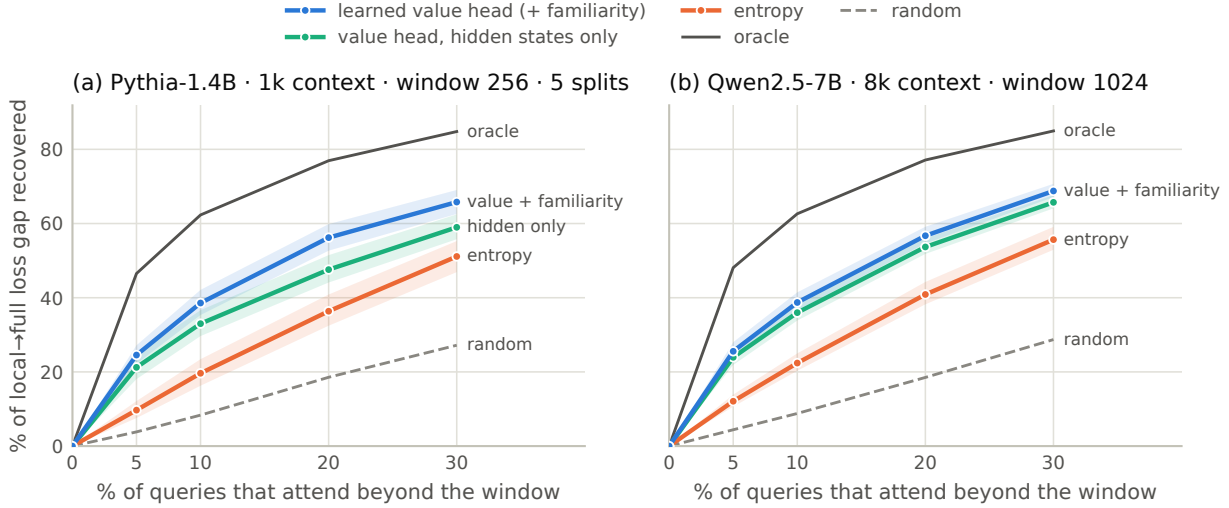


Figure 4: Per-token gating in real forward passes, at every budget. Bands: 95% document-bootstrap intervals.

- Features are the hidden states of the middle and last layers, concatenated.
- End to end, $\alpha = 1000$ for hidden-state heads and $\alpha = 10$ for the 7-scalar familiarity head, fitted from streaming normal equations in float64 (the same estimator as standardising and then running ridge regression).
- Elsewhere, α is chosen from $\{10, 100, 1000, 10^4\}$ on 20% of the training documents.
- Training on the realized gain instead of v_t gave nearly identical curves.

Familiarity and recurrence addressing. Per document, we keep hash tables of the 2-, 3- and 4-grams whose last token lies outside the current query’s window, plus the continuations of each 2-gram. They grow incrementally as t advances, so the cost per token is constant. Recurrence addressing reads the stored end positions of the matched n -gram.

Uncertainty and compute. Intervals come from 2,000 resamples of test documents, recomputing R_b as a ratio of sums each time. Experiments ran on single rented GPUs: an RTX PRO 5000 48 GB, an A100 80 GB and an H100 NVL 94 GB. They cost about US\$24 in total.

C Data

- **Papers.** 53 papers from arXiv’s September 2026 listings, extracted from the arXiv HTML rendering: five each from cs.CL, cs.LG, cs.CR, math.PR, stat.ME, physics.optics, cond-mat.stat-mech, astro-ph.GA, eess.SY and econ.GN, and three from q-bio.NC. Fifty-one have identifiers 2609.xxxxx; two econ.GN papers (2405.04352 and 2411.05938) were first posted in 2024.
- **Code.** 60 Python files from recent releases of `huggingface_hub` (22), `pandas` (19), `aiohttp` (7), `anyio` (4), `httpcore2` (4), `httpx2` (3) and `yaml` (1).

Table 6: Familiarity in per-token gating (10% budget). Hidden/entropy: the value head on hidden states only, relative to entropy. Δ : points added by the six familiarity features, on natural text, on the unseen papers and on code. Brackets: paired 95% bootstrap intervals.

Model	Ctx	W	Hidden/entropy	Δ natural	Δ papers (Sept. 2026)	Δ code
Pythia-160M	1k	64	1.65 [1.53, 1.80]	+4.4 [3.3, 5.5]	+5.7 [4.4, 7.0]	+3.2 [1.5, 4.8]
Pythia-410M	1k	64	1.50 [1.41, 1.61]	+4.6 [3.6, 5.5]	+5.4 [4.0, 6.7]	+3.7 [2.4, 5.0]
Pythia-410M	1k	256	1.70 [1.49, 1.98]	+7.1 [4.0, 9.8]	+10.4 [6.8, 13.9]	+2.5 [-1.6, 6.8]
Pythia-1.4B	1k	64	1.51 [1.41, 1.61]	+2.0 [1.1, 2.8]	+2.3 [1.2, 3.4]	+1.7 [0.5, 3.0]
Pythia-1.4B	1k	256	1.68 [1.46, 1.97]	+5.6 [3.6, 7.6]	+7.3 [5.0, 9.7]	+2.7 [-0.7, 5.7]
Qwen2.5-1.5B	4k	256	1.52 [1.45, 1.59]	+2.9 [2.3, 3.5]	+3.4 [2.8, 3.9]	+2.3 [1.0, 3.5]
Qwen2.5-1.5B	4k	1024	1.69 [1.55, 1.84]	+4.5 [2.6, 6.5]	+4.9 [3.0, 6.8]	+3.6 [0.2, 8.6]
Qwen2.5-7B	8k	256	1.47 [1.39, 1.54]	+1.9 [1.5, 2.3]	+1.7 [1.4, 2.1]	+2.7 [2.0, 3.5]
Qwen2.5-7B	8k	1024	1.61 [1.48, 1.72]	+2.7 [2.0, 3.4]	+2.7 [1.9, 3.5]	+3.0 [1.4, 4.3]

Table 7: Per-token gating by domain (10% budget). The papers postdate the training data of every model; the code comes from recent releases of packages whose earlier versions were probably seen. Ratios to entropy, with paired 95% bootstrap intervals.

Model	Ctx	W	Papers (Sept. 2026)				Code		
			Entropy	Value	Value/entropy		Entropy	Value	Value/entropy
Pythia-160M	1k	64	13.4	31.7	2.37 [2.12, 2.69]		21.3	35.1	1.64 [1.50, 1.81]
Pythia-410M	1k	64	15.2	30.1	1.98 [1.76, 2.21]		24.5	38.1	1.56 [1.45, 1.70]
Pythia-410M	1k	256	14.7	36.9	2.51 [2.12, 2.98]		29.9	49.3	1.65 [1.44, 1.99]
Pythia-1.4B	1k	64	15.3	27.6	1.81 [1.64, 2.01]		25.9	37.8	1.46 [1.36, 1.58]
Pythia-1.4B	1k	256	14.5	32.6	2.25 [1.83, 2.75]		28.5	48.6	1.70 [1.44, 2.14]
Qwen2.5-1.5B	4k	256	17.5	31.2	1.79 [1.68, 1.90]		30.6	46.5	1.52 [1.41, 1.64]
Qwen2.5-1.5B	4k	1024	17.6	36.3	2.06 [1.79, 2.39]		33.7	57.1	1.70 [1.53, 1.90]
Qwen2.5-7B	8k	256	18.5	30.6	1.65 [1.59, 1.73]		30.4	42.8	1.41 [1.29, 1.54]
Qwen2.5-7B	8k	1024	18.8	35.3	1.88 [1.78, 1.98]		33.0	48.5	1.47 [1.34, 1.65]

- **Identifier copies.** In copies of 51 papers, ten bracketed random 8-hex identifiers are inserted at sentence ends. Six appear once, and their characters are unpredictable. Two appear twice, and the repeat is predictable only from memory. Copies are never used for training and follow their source paper’s split.
- **Lengths.** Pythia runs use the first 1,024 tokens of each document. Qwen runs use full-length versions (up to 60k characters) truncated to 4,096 or 8,192 tokens. Documents under 256 tokens are dropped.
- **Tuned-lens data.** 38 papers from 2026 in ten other categories, and 20 files from `scikit-learn`, `scipy` and `transformers`.
- **Token counts.** The signal-level experiments score 108,593 natural tokens; the identifier copies add 49,011 tokens.

Table 8: Per-token gating at every budget (natural text). $[\cdot]$: 95% bootstrap interval at 10%. “Indep.”: recovery predicted at 10% if each gated token received exactly its full-context loss. The end-to-end value is 2.3–8.8 points lower in every case, because gated queries attend to keys and values that ungated tokens computed without far context. Part 1 of 2.

Signal	$R_{5\%}$	$R_{10\%}$		$R_{20\%}$	$R_{30\%}$	Indep. $R_{10\%}$
<i>Pythia-160M, 1k, $W = 64$: loss 3.136 local, 2.601 full; 54,777 tokens</i>						
random	4.5	9.1	[8.5, 9.8]	18.1	27.4	–
entropy	9.1	17.6	[15.7, 19.5]	32.3	46.2	20.9
familiarity head	16.2	27.1	[25.4, 28.7]	41.7	53.9	–
value head, hidden only	17.3	29.1	[27.3, 31.0]	44.9	56.5	–
value head + familiarity	21.2	33.5	[32.1, 35.0]	51.3	63.2	35.8
oracle (KL)	33.7	49.4	[47.9, 51.0]	66.0	76.0	54.9
<i>Pythia-410M, 1k, $W = 64$: loss 2.729 local, 2.262 full; 54,777 tokens</i>						
random	4.2	8.8	[8.2, 9.5]	17.3	27.1	–
entropy	10.9	19.6	[18.0, 21.2]	36.3	49.8	24.1
familiarity head	16.7	28.0	[26.0, 29.7]	43.5	56.5	–
value head, hidden only	18.6	29.4	[27.4, 31.5]	46.6	59.3	–
value head + familiarity	21.1	34.0	[32.0, 36.0]	52.3	64.9	37.5
oracle (KL)	36.4	52.1	[50.0, 54.2]	69.5	79.9	59.8
<i>Pythia-410M, 1k, $W = 256$: loss 2.385 local, 2.191 full; 43,833 tokens</i>						
random	4.3	8.5	[7.4, 9.7]	18.9	27.4	–
entropy	11.2	20.4	[17.4, 23.9]	37.2	50.8	25.8
familiarity head	20.9	33.3	[29.9, 36.9]	48.5	61.5	–
value head, hidden only	22.6	34.8	[31.4, 38.7]	50.1	62.0	–
value head + familiarity	26.9	41.9	[39.0, 45.0]	59.4	71.0	45.4
oracle (KL)	48.2	65.0	[62.1, 68.4]	79.3	86.4	71.9
<i>Pythia-1.4B, 1k, $W = 64$: loss 2.479 local, 2.017 full; 54,777 tokens</i>						
random	4.4	9.2	[8.5, 9.9]	18.1	28.0	–
entropy	11.1	20.1	[18.3, 22.2]	36.4	50.3	24.3
familiarity head	16.1	26.6	[24.6, 28.8]	42.8	56.4	–
value head, hidden only	18.6	30.3	[28.4, 32.4]	46.9	59.1	–
value head + familiarity	20.1	32.3	[30.4, 34.3]	50.9	63.4	36.3
oracle (KL)	36.7	52.0	[49.7, 54.5]	69.1	78.3	59.6
<i>Pythia-1.4B, 1k, $W = 256$: loss 2.142 local, 1.947 full; 43,833 tokens</i>						
random	3.8	8.3	[7.5, 9.2]	18.6	27.2	–
entropy	9.7	19.6	[16.3, 23.4]	36.4	51.1	25.3
familiarity head	19.3	31.0	[27.3, 34.7]	47.7	59.7	–
value head, hidden only	21.2	33.0	[29.7, 36.6]	47.6	59.0	–
value head + familiarity	24.5	38.6	[35.3, 42.1]	56.2	65.8	41.7
oracle (KL)	46.5	62.3	[59.3, 65.5]	76.9	84.8	68.6

Table 9: Per-token gating at every budget (continued, part 2 of 2).

Signal	$R_{5\%}$	$R_{10\%}$		$R_{20\%}$	$R_{30\%}$	Indep. $R_{10\%}$
<i>Qwen2.5-1.5B, 4k, W = 256: loss 1.964 local, 1.652 full; 195,461 tokens</i>						
random	4.1	8.4	[7.8, 9.0]	17.4	26.7	—
entropy	12.3	22.3	[20.2, 24.5]	39.8	54.6	26.6
familiarity head	17.1	28.3	[26.2, 30.7]	45.0	58.4	—
value head, hidden only	21.6	33.9	[31.4, 36.6]	51.4	63.5	—
value head + familiarity	24.1	36.8	[34.4, 39.5]	55.2	67.1	40.7
oracle (KL)	42.3	57.7	[55.1, 60.8]	73.3	81.8	65.2
<i>Qwen2.5-1.5B, 4k, W = 1024: loss 1.747 local, 1.623 full; 151,685 tokens</i>						
random	4.4	8.8	[8.1, 9.7]	18.8	28.3	—
entropy	13.2	23.3	[20.5, 26.5]	41.0	55.8	27.7
familiarity head	18.8	29.8	[27.6, 32.2]	47.2	61.3	—
value head, hidden only	26.7	39.2	[35.8, 43.7]	56.6	67.6	—
value head + familiarity	29.5	43.7	[40.8, 47.3]	60.5	72.7	46.9
oracle (KL)	56.5	70.5	[67.8, 73.5]	83.1	89.6	76.8
<i>Qwen2.5-7B, 8k, W = 256: loss 1.807 local, 1.439 full; 326,704 tokens</i>						
random	4.3	8.9	[8.5, 9.4]	18.1	28.2	—
entropy	11.6	21.9	[20.3, 24.1]	39.2	54.2	26.5
familiarity head	15.1	26.2	[24.3, 28.8]	43.5	57.4	—
value head, hidden only	20.1	32.2	[30.4, 34.5]	49.5	62.4	—
value head + familiarity	21.5	34.2	[32.3, 36.6]	51.9	64.8	38.3
oracle (KL)	38.3	53.2	[50.8, 56.4]	69.5	79.0	61.6
<i>Qwen2.5-7B, 8k, W = 1024: loss 1.598 local, 1.426 full; 282,928 tokens</i>						
random	4.4	8.8	[8.2, 9.4]	18.5	28.7	—
entropy	12.1	22.4	[20.3, 25.0]	40.9	55.7	27.9
familiarity head	16.7	28.6	[26.1, 31.7]	46.1	60.9	—
value head, hidden only	23.9	36.0	[33.8, 38.8]	53.7	65.7	—
value head + familiarity	25.6	38.7	[36.6, 41.4]	56.7	68.8	42.4
oracle (KL)	48.1	62.6	[60.2, 65.5]	77.1	85.0	71.5