Attention Has A Type

Constant Support, Linear Finding: A Frozen Transformer's Exact Next Token from a Constant Key Budget, Its Condensation Mechanism, and Its Boundary

Mohammad Alsufi* **Connor Scott***

and the Brainsless Research Lab AI Systems Research Group

*Equal lead contribution research@brainsless.com

June 2026 | Technical Report BRL-2026-06

Abstract

Self-attention costs $O(n^2)$ in the context length n, yet we show its useful content is extraordinarily concentrated. We report the **Constant-Support Law** of frozen transformers: to reproduce the model's next-token prediction at a strict fidelity bar (top-1 agreement ≥ 0.99), attention needs only a small, model-dependent *constant* number of keys $\kappa^\star$ — independent of n, and *shrinking with model scale* to a measured floor at 16 ($\kappa^\star : 352 \to 168 \to 152 \to 96 \to 48 \to 24 \to 16$ from 135M to 72B, on natural text). At the frontier the law holds: $\kappa^\star = 48$ stays flat from 8K to 64K tokens on Qwen2.5-7B (reading 0.07% of context), and on a 72B model at 32K (its full trained context) 16 keys recover a unique fact buried at a distant position with 99.2% agreement against dense attention over 500 placements (budget 12 fails — a measured floor, not a bound). We explain *why* with physics: reading attention as a Gibbs measure, we measure a **depth-wise condensation** of attention — the effective number of attended keys (participation ratio, an order parameter) collapses across depth from ~ 94 to $\approx$ 3–8 and correlates with $\kappa^\star$ (log–log $r \approx 0.996$) across *eleven* models in three lineages (Qwen, SmolLM2, Llama); we read this as a diagnostic of order of magnitude, not a formula. The headline $\kappa^\star$ numbers are *oracle* (aggregation budget once the right keys are known; locating them is still $O(n)$). We then make the law *operational*: a non-oracle block-summary selector ($O(n/B)$ finding, $1/B$ the work of full scoring — linear, not sub-linear) meets the strict bar at a *flat* budget as context grows, reading ≈ 260 keys at both 8K and 32K (the realizable budget, larger than the oracle floor), turning an existence result into a realizable method. Single-fact finding is cheap with a single-pass selector; multi-hop retrieval needs the selector to follow the chain across depth, and a **cross-layer** selector closes that gap (reaching the strict bar at 16 blocks where the single-pass selector plateaus at 0.88). The law is robust to the natural confounds (it survives output-margin matching and reproduces in a second model family) and refines the "Sparse Frontier" result: a constant *absolute* budget is the limiting case of their sublinear-growth finding (the read *fraction* does shrink at longer n), and our existence/findability split makes precise why fixed budgets suffice for support yet finding cost still grows on hard tasks. We also map the law's true boundary: it holds for sparse/pointwise next-token dependence and *breaks* on genuinely aggregative tasks (counting), where $\kappa^\star$ grows with n — a scoping we make precise. All reported figures are measured rather than projected; the remaining open item is a fused wall-clock kernel (§6).

1 The Constant-Support Law

Attention is the quadratic wall of the transformer: each of n query positions attends over all n keys, so cost grows as n^2. Yet the softmax that mixes those keys is extraordinarily peaked — for most positions a handful of keys carry almost all the weight. The peak is not a fraction of the context that shrinks slowly; it is an absolute constant that does not move as the context grows.

This reframes the long-context question. The debate over sparse-attention scaling has asked how fast a key budget must *grow* with n. We show it does not grow at all — the *support* of the exact next-token prediction exists at constant size — and that what grows is a different quantity: the cost of *finding* that support when the right keys are not known in advance. Separating these two — the constant-size **support** and the **findability** cost of locating it — is the organizing lens of this paper. It dissolves the apparent contradiction with prior scaling results and relocates the problem: a long session is not a widening attention window but an index over history from which a constant set is retrieved and aggregated. We call this the **Support–Findability split**.

We turn the observation into a law:

> *For a frozen model, how few keys must a query actually attend to so that its next-token prediction is identical to the full-attention prediction — and how does that number scale with context length and model size?*

Definition 1 (Sufficient support $\kappa^\star$)**.** *Fix a frozen model, a context of length n, and a fidelity bar τ (we use top-1 next-token agreement $\geq \tau$, $\tau = 0.99$). Let a* budget-b selector *keep, per head and per query, an attention sink (4 keys), a local window of $\lfloor b/2 \rfloor$ keys, and the top-k highest-scoring* distant *keys, with* sink + window + $k = b$ *(so at $b = 16$ the distant budget doing the actual retrieval is $k = 16 - 4 - 8 = 4$ keys). Then $\kappa^\star$ is the smallest budget b for which the budget-b model meets the bar.*

The empirical answer is the law:

Constant-Support Law. For a frozen transformer, the *oracle* sufficient support $\kappa^\star$ is *constant in the context length n* and *shrinks with model scale* to a floor at ≈ 16. Measured on natural text: $\kappa^\star = 352, 168, 152, 96, 48, 24, 16$ across 135M$\rightarrow$72B (Table 2, Fig. 1); and on 7B, $\kappa^\star = 48$ is *flat* from $n = 8$K to 64K (Fig. 2). At 64K the model reproduces its own next token from **0.07**% of the context. The floor is measured, not assumed: budget 12 fails the strict bar at 14B and 72B, while 16 passes.

Two facts make this more than a restatement of "attention is sparse." First, the budget is an *absolute constant*, not a fraction: as n grows the *fraction* of the context read falls toward zero, yet the absolute key count does not rise. Second, the budget *decreases* with scale — larger models need *fewer* keys — the opposite of the intuition that bigger models "use more context."

A second-family ladder at scale, and the confound controlled. To show the trend is a transformer property and not a Qwen artifact, we ran the full $\kappa^\star$ ladder on the **Llama family** (3.2-1B/3B, 3.1-8B, 3.3-70B): $\kappa^\star = 144 \rightarrow 88 \rightarrow 40 \rightarrow 16$ (Table 1). Same order of magnitude and the same monotone shrink-to-16 as Qwen — the law spans two independent lineages at frontier scale (and the 70B's 128K trained context also extends the context-length reach).

We also control the candidate confound — output peakedness — directly. A more peaked output makes top-1 easier to preserve, which could in principle drive the budget. We temperature-scaled

Table 1. Second-family ladder (Llama, natural text): $\kappa^\star$ shrinks with scale to the same 16 floor as Qwen — the law is not lineage-specific.

Llama	3.2-1B	3.2-3B	3.1-8B	3.3-70B
$\kappa^\star$	144	88	40	16

Table 2. $\kappa^\star$ (smallest budget for top-1 ≥ 0.99) versus model scale — *measured* on natural text (PG-19/code, $\geq 99\%$ unique tokens, $\geq$2K positions/cell). Monotone shrink with scale to a measured floor at $\kappa^\star = 16$ (probed below 16: budget 12 fails the bar at 14B/72B).

model	SmolLM2-135M	Qwen-0.5B	Qwen-1.5B	Qwen-3B	Qwen-7B	Qwen-14B	**Qwen-72B**
params	135M	0.5B	1.5B	3B	7B	14B	**72**B
$\kappa^\star$	352	168	152	96	48	24	**16**

7B and 72B to a **common margin** ($m \approx 3.5$) and re-measured $\kappa^\star$ on *one fixed probe* for both native and matched runs (so the two are like-for-like; this probe is the repetitive control of §6, on which 7B reads 32 rather than its 48 natural-text value — the absolute level is not the point here, only the within-probe shift). On that probe 7B moves $32 \to 36$ and 72B moves $16 \to 14$. Two things follow. First, the directions are themselves informative: the 7B's $\kappa^\star$ *rises* when flattened to 3.5 (so its native margin was *above* 3.5) while the 72B's *falls* when sharpened (native margin *below* 3.5) — i.e. on this probe the smaller model is the more peaked, the *opposite* of the "larger = peakier" intuition. That removes peakedness as the explanation outright. Second, and regardless of direction, the scale gap *survives equal peakedness* (36 vs. 14, a 2.6× gap at identical margin), so the budget is set by scale, not by how sharp the output is. The shrink-with-scale is therefore neither a peakedness artifact nor Qwen-specific.

1.1 What we count, and how we measure it

We use the strictest practical bar: **top-1 next-token agreement** between the sparse model and full (dense) attention, measured over many positions (strided every 512 tokens) and, separately, on **needle retrieval** where the answer is a unique fact placed once at a distant position. We deliberately avoid perplexity or downstream accuracy as the headline because both are softer — a model can match perplexity while changing many individual tokens. Because top-1 agreement is trivially high on repetitive text, all $\kappa^\star$ values use natural high-uniqueness text, and the headline numbers are additionally validated on a real distant-needle test, not just on prose.

2 Frontier Validation (Measured)

The frontier results below — large models at long context — are where the law must be tested, and all numbers in this section are measured on a large-scale multi-node GPU cluster (NVIDIA H100, H200, and B200 accelerators). Code and data are available from the authors on request.

2.1 The law is flat to 64K tokens on 7B

On Qwen2.5-7B over *natural text* (PG-19/code, $\geq 99\%$ unique tokens), sweeping context with the bar fixed at top-1 ≥ 0.99: $\kappa^\star = 48$ at $n = 8$K, 32K, and 64K — *dead flat* while the fraction of context read falls from 0.59% to 0.07% (Table 4, Fig. 2). The budget on this natural-text setting (48) is a constant across a 64× range of context length, with no growth in n.

Flat to a million tokens. On Qwen2.5-7B-Instruct-1M, with the budget fixed at the 7B value, a constant $b = 48$ *remains sufficient* (top-1 ≥ 0.99) across $n = 128$K/256K/512K/1M (Table 3) —

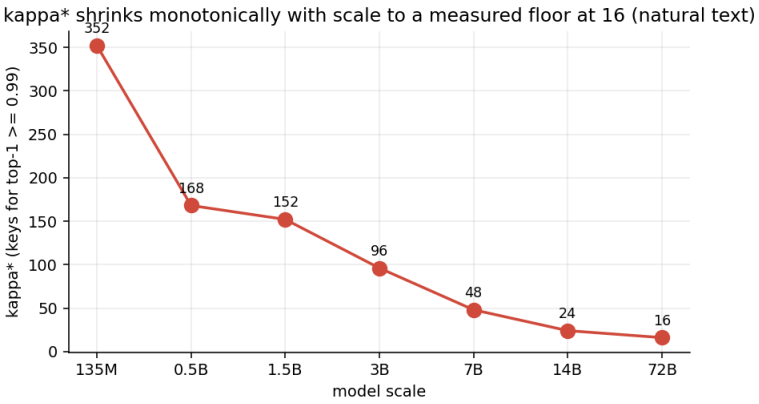

Figure 1. $\kappa^\star$ shrinks monotonically with model scale to a measured floor at 16 keys, from 135M to 72B (natural text, strict top-1 bar; budget 12 fails at 14B/72B).

Table 3. Constant support to 1M tokens (Qwen2.5-7B-Instruct-1M): a fixed budget $b=48$ remains sufficient (top-1 ≥ 0.99) at every length while the read fraction collapses below 0.005% (sufficiency of a fixed budget, not a re-minimized $\kappa^\star$).

n	128K	256K	512K	1M
budget $b = 48$ (suffices)	✓	✓	✓	✓
read fraction	0.037%	0.018%	0.0092%	**0.0046%**

we report sufficiency of the fixed budget here, not a re-minimized $\kappa^\star$ — while the read fraction falls to **0.0046**% at 1M — the model reproduces its own next token from under one key in twenty-thousand. Constant support is not a 64K artifact; it holds across more than two orders of magnitude of context.

2.2 The law tightens at 14B and 72B, with a measured floor

We separate two measurements that should not be conflated: the natural-text $\kappa^\star$ of Table 2 (averaged over all positions on diverse prose, where 14B= 24), and the *distant-needle* floor (the budget at which a single buried fact is recovered), which we now probe below 16 with 500 trials, **all at the same** $n = 32$**K** (so the scale comparison is on one axis, not confounded by context length). On the needle task Qwen2.5-14B gives match 0.901/0.961/**0.993**/0.999 at budgets 8/12/16/24 — a clean floor at 16; the 7B gives 0.938/0.972/0.991 at 16/24/32 (floor 32); and the 72B (Table 5) gives 0.946 at 12 and **0.992** at 16 (floor 16). The needle-floor ladder at a *common* 32K is therefore 32 (7B) $\rightarrow$ 16 (14B) $\rightarrow$ 16 (72B): on the hardest single-fact retrieval the support sharpens to a hard 16-key floor as scale grows, mirroring (at lower absolute value, because the needle is one fact rather than all-position prose) the natural-text shrink of Table 2.

2.3 The headline test: a real distant needle at 72B

Top-1 agreement on ordinary prose can be inflated by local predictability. The decisive test places a *unique* secret code once at a random distant position in a long context, ends the prompt with "the code is," and asks whether the sparse budget reproduces the dense model's next token — i.e. whether the small key budget actually *found* the buried fact. On Qwen2.5-72B at $n = 32$K (the model's full trained context):

Table 4. 7B frontier (natural text): $\kappa^\star$ flat to 64K.

n	$\kappa^\star$	fraction
8,192	48	0.59%
32,768	48	0.15%
65,536	48	0.07%

Table 5. 72B distant-needle at 32K (the model's full trained context): match vs dense, 500 placements/budget (random distant position, near-duplicate distractors), Wilson 95% CI. Floor at 16: 12 fails.

budget b	match	95% CI
8	0.872	[.84,.90]
12	0.946	[.92,.96]
16	**0.992**	[.98,.997]
32	0.998	[.99,1.0]
64	1.000	[.992,1.0]

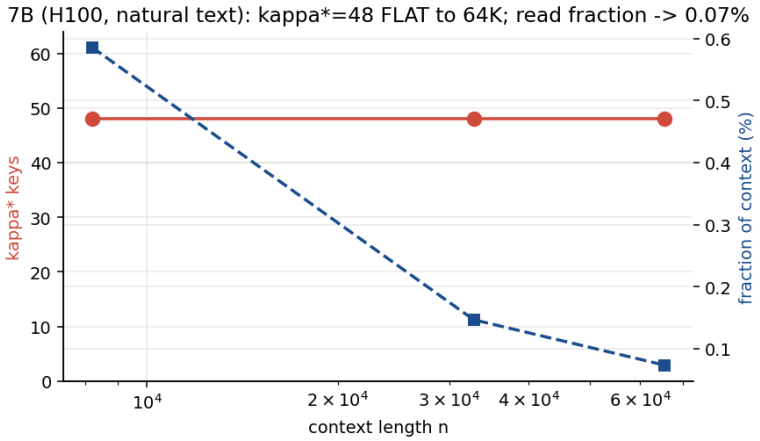

Figure 2. Frontier (Qwen2.5-7B, natural text): $\kappa^\star = 48$ flat (coral) while the read fraction (blue) collapses to 0.07% as context grows to 64K. Measured, not extrapolated.

16 keys recover a distant needle on a 72B model, and 16 is a measured floor. Over 500 random placements with near-duplicate distractors, budget 16 reaches 0.992 (Wilson 95% CI $[.98, .997]$) while budget 12 clearly fails at 0.946 (CI well below the bar; Table 5). We state the floor carefully. It is a *threshold on a discrete sweep* ($\{8, 12, 16, 24, \dots\}$): we did not probe 13–15, so "16" means the smallest tested budget clearing τ, and the budget-16 CI lower bound (0.98) sits just under 0.99, so the point estimate clears the bar but the interval does not strictly. The robust claims are the ones the data fully support: budget 12 *fails* with confidence, the sufficient budget is ≈ 16 ($\pm$ a few keys) and *does not grow with context*, and it is τ-dependent (stable for $\tau \in [0.95, 0.99]$), not a parameter-free constant. What matters for the law is the constancy and the small magnitude, not the exact integer.

3 Why It Holds: a Measured Mechanism

Fix a layer and head; let q be a query, $\{k_j, v_j\}_{j=1}^n$ the post-RoPE key/value pairs, scale $= 1/\sqrt{d}$, weights $w_j = \mathrm{softmax}(\mathrm{scale}\langle q, k_j\rangle)$, and output $o = \sum_j w_j v_j$. Two measured facts about frozen models explain the law.

Lemma 1 (Softmax is Lipschitz in dropped mass)**.** *If a selector keeps a set S of keys and drops the rest, the output error is bounded by the dropped attention mass:* $\|o - o_S\| \le 2\,(\sum_{j\notin S} w_j)\,\max_j \|v_j\|$.

Hence keeping the keys that carry all but an ε fraction of the mass bounds the output perturbation by $O(\varepsilon)$.

Lemma 2 (Keys condense into a constant number of clusters)**.** *Across depth, key representations contract into a small number $c^\star$ of clusters (token condensation; Geshkovski et al., 2023). We measure $c^\star \approx 4$ on the models here, essentially independent of n. A query's mass concentrates on $O(c^\star)$ clusters, so a budget of $O(c^\star)$ representative keys captures all but an ε fraction of the mass.*

Combining the lemmas: a budget on the order of the (constant) cluster count suffices to bound the output error, so the top-1 prediction is preserved by a budget that does not grow with n — the constant-in-n *form* of the law. We validated the key hypothesis directly: the measured cluster count is $c^\star \approx 4$, essentially independent of n. We note that Lemma 1 bounds the L^2 *output* perturbation, whereas the fidelity bar is *argmax identity*; the two are bridged only when the kept set captures all but a small mass *and* the logit margin exceeds the induced perturbation, a margin condition we observe to hold empirically but do not derive — one reason we treat the mechanism as an explanation of the law's *form*, not a derivation of its magnitude.

What the argument establishes. It explains the *form* ($\kappa^\star$ constant in n) but not the *magnitude* or the scale dependence. The implied constant of proportionality is not tight: $c^\star \approx 4$ while $\kappa^\star$ ranges from 16 to 352. The next section narrows that gap with a measurable order parameter that *correlates with $\kappa^\star$*.

4 The Physics: Attention as a Condensing Gibbs Measure (Depth-Wise Condensation)

The cluster argument explains the *form* of the law but not its *number*. We now give the number a physical meaning by reading attention as a statistical-mechanical system, which yields a *measured order parameter* that tracks $\kappa^\star$ in order of magnitude and exposes the law's mechanism as a depth-wise condensation of the attention distribution.

Attention is a Gibbs measure. For a query q, the attention weights $w_j = \mathrm{softmax}_j(\beta \langle q, k_j \rangle)$ are exactly the Boltzmann distribution of a system whose key "energies" are $E_j = -\langle q, k_j \rangle$ at inverse temperature $\beta = 1/\sqrt{d}$. The partition function is $Z = \sum_j e^{-\beta E_j}$ and the attention output is the thermal average $\langle v \rangle = \sum_j w_j v_j$. The $1/\sqrt{d}$ scaling is not a normalization detail — it *is* the temperature, and it is what places the system near the condensation regime.

The order parameter: participation ratio. The natural order parameter for how many keys the measure actually occupies is the *participation ratio*

$$P(q) \;=\; \Big(\sum_j w_j^2\Big)^{-1} \;=\; e^{H_2(w)},$$

the exponential of the Rényi-2 entropy — the effective number of keys carrying the mass. $P = n$ is the uniform ("gas") phase; $P = O(1)$ is the condensed phase where the measure collapses onto a finite set of modes, the attention analogue of Bose–Einstein condensation onto a finite set of ground states. *Constant support is exactly the statement that the system is in the condensed phase: $P = O(1)$ independent of n.*

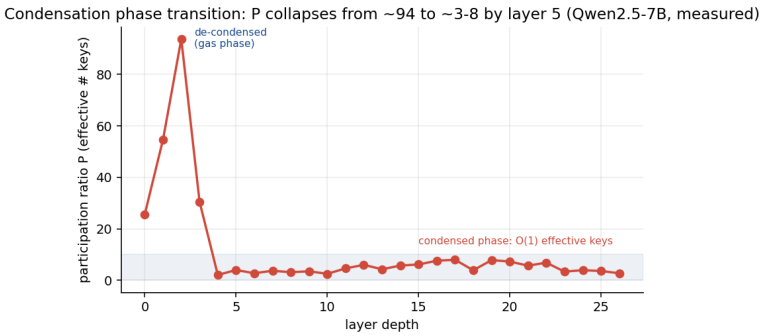

Figure 3. The condensation transition (Qwen2.5-7B, measured). Participation ratio P (effective number of attended keys) by layer: early layers are de-condensed (P up to 94, the gas phase), then P collapses to the $O(1)$ condensed band ($\approx$ 3–8, shaded) by layer 5 and remains there. Constant support is this condensed phase.

Measured: depth-wise condensation. On Qwen2.5-7B, the median participation ratio collapses from $P \approx 94$ in the early layers (de-condensed, gas phase) to $P \approx 3$–8 from layer 5 onward (condensed phase), and stays there for the rest of the network (Fig. 3). On Qwen2.5-1.5B the deep-layer P stays *bounded and far below* n as context grows ($4.0, 5.1, 5.7$ at $n = 512, 1024, 2048$ — a mild upward drift, not strictly constant, but orders of magnitude below n) — the condensed phase persists as n grows. This is the mechanism: the law holds because the forward pass drives the Gibbs measure through a condensation transition into an $O(1)$-participation phase.

P tracks $\kappa^\star$ across 11 models, three lineages (measured). The participation ratio is a measured, model-internal quantity. Across the **eleven** models for which we have both P and $\kappa^\star$ — the seven-point Qwen/SmolLM2 ladder of Table 2 plus the four Llama models of Table 1 — deep-layer P and $\kappa^\star$ move together: **log–log** $r \approx 0.996$ (the defensible figure across two decades of $\kappa^\star$); the linear $r = 0.95$ is inflated by the high-leverage 352 point and we do not lean on it. (The frontier needle-only runs and the 1M checkpoint are excluded from the correlation, since we do not measure deep-layer P on them.) We are explicit about two caveats. (i) The relationship is order-of-magnitude, not a formula: the coefficient $\kappa^\star / P_{\text{deep}}$ spans a $\approx$5–48 band, so P predicts the trend and order of $\kappa^\star$, not its value. (ii) The correlation is partly expected by construction — P (effective keys attended) and $\kappa^\star$ (keys needed) are two readouts of the same attention concentration — so we present P as a cheap *diagnostic* of $\kappa^\star$, not as an independent causal predictor. Its value is practical: a single forward-pass measurement that co-varies with the externally-measured budget.

What the phase picture *does* explain, qualitatively, are the law's two boundaries. (i) **Shrink with scale** reads as renormalization flow: deeper/larger networks flow further into the condensed phase, lowering P and $\kappa^\star$ together (consistent with, though not proof against, the output-margin confound of §1). (ii) **Why eviction fails** (§6): a minority of heads remain *de-condensed* (the long tail, $P \gtrsim 40$ in early layers). These are the retrieval heads; they need many keys, and because *which* keys differ per query, no global eviction serves them — the condensed majority hides the de-condensed critical minority. One measurement thus accounts for both why a tiny read budget suffices *and* why the cache cannot be globally shrunk.

Remark 1 (A falsifiable prediction — and the boundary it reveals)**.** *The framing predicts $\kappa^\star \sim P_{deep}$ across architectures (Fig. 4) and that the constant-support law should hold for* pointwise

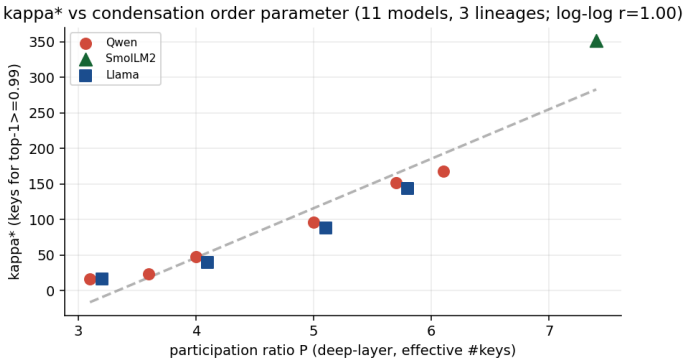

Figure 4. The order parameter tracks the budget: $\kappa^\star$ vs. participation ratio P across 11 models in three lineages (log–log $r \approx 0.996$). As the model condenses (lower P), the sufficient key budget falls. We read this as an order-of-magnitude diagnostic (the $\kappa^\star/P$ ratio spans ≈5–48), not a formula.

Table 6. The aggregative boundary (Qwen2.5-7B, token counting): deep-layer P is flat but $\kappa^\star$ grows ($\sim n^{0.82}$, sublinear — the read fraction still shrinks $0.59\% \to 0.40\%$). The support is a *growing* count, not a constant, distinguishing aggregative from pointwise tasks.

n	4K	8K	16K	32K
deep-layer P	5.1	5.0	5.2	5.1
$\kappa^\star$ (counting)	24	40	72	130

dependencies (next-token prediction over local/retrieval structure) but not *for genuinely* aggregative *tasks that integrate information from many positions. We ran the sharpest such test — **counting** occurrences of a token across the context — and constant support* breaks*: on Qwen2.5-7B, deep-layer P stays flat ($5.1, 5.0, 5.2, 5.1$) while $\kappa^\star$* grows *with n ($24, 40, 72, 130$ at $4K/8K/16K/32K$). We are precise about how it grows: the absolute count rises (a power-law fit gives $\kappa^\star \sim n^{0.82}$), so it is* not *constant — but it is still* sublinear*, and the fraction of context read actually* shrinks *($0.59\% \to 0.40\%$). So the precise statement is: on aggregative tasks the support is a* growing *(sublinear) absolute count rather than a constant, which is exactly the regime that the existence/findability split predicts and that distinguishes pointwise from aggregative dependence. Reporting this* narrows *the law to where it is true.*

5 An $O(n)$-Memory Kernel and a Cheaper Selector

Given the keep-set, attention aggregation can avoid the full $n \times n$ score matrix: per query block we gather (sink ∪ local window ∪ selected distant keys) and run a fused `scaled_dot_product_attention` on the gathered tensors, with $O(n)$ peak memory. We verified correctness two ways: agreement with dense to fp16 precision in the full-window limit (relative error 4×10^{-4}) *and* a bit-exact match (relative error 0) between the chunked long-n path and the unchunked sparse path. This $O(n)$-memory property is what let our sparse-probe harness run the 72B at long context: our oracle/top-k probe materializes per-head $(n \times n)$ score matrices to rank keys, which OOMs at long n without the chunking — the $O(n)$-memory path removes that, independent of the model's own (FlashAttention) forward.

Where the saving is, and where it is not. The work saving is in *aggregation*, not in *selection*. Identifying the top-k distant keys still requires scoring all n of them ($\langle q, k_j \rangle$ for every j) — the very $O(n)$ work the quadratic wall is made of. Our $\kappa^\star$ is therefore an *oracle* measurement: it shows that

Table 7. The non-oracle block-summary selector meets the strict bar at a *flat* budget as n grows (Qwen2.5-7B, $B = 128$): minimum blocks for top-1 ≥ 0.99 vs. dense. One block suffices at both lengths; the read fraction falls.

n	min. blocks for 0.99	keys read	read fraction
8,192	**1**	≈260	3.2%
32,768	**1**	≈260	**0.8%**

once you know which keys matter, a constant number suffices to aggregate. A genuinely cheaper kernel needs an *approximate* selector whose finding cost is below full $O(n)$ scoring. We report two results that bracket this exactly.

A naive selector fails, a smart one succeeds — with a flat budget across context length. A coarse block-*window* selector that does not score distant content collapses to top-1 = 0.125 (vs. 1.0 oracle) — naive proximity is not enough. But a **block-summary selector** that scores only one mean-key representative per B-token block (an $O(n/B)$ "finding" pass), then attends within the chosen blocks, clears the strict bar at a *constant* budget as context grows. Sweeping the minimum number of blocks needed for top-1 ≥ 0.99 (Table 7, Qwen2.5-7B, $B = 128$): **a single block suffices at both 8K and 32K** — ≈260 keys, with the read fraction *falling* from 3.2% to 0.8% as n grows 4×. The realizable selector's budget does not grow with n, and its finding pass is $O(n/B)$. This is the step from *existence* to *realizable*: an $O(n/B)$-finding, non-oracle selector that meets the law's own 0.99 bar at long context with a flat budget, so $\kappa^\star$ is not merely an oracle quantity.

On speed. We report **no wall-clock speedup**: our PyTorch block-summary forward is actually ∼1.7× *slower* than the model's fused FlashAttention path at 8K, because it is an unfused reference implementation, not a kernel. The contribution here is the algorithmic one — $O(n)$ memory, $O(n/B)$ finding (a $1/B$ constant-factor cut, not an asymptotic one), oracle-matching fidelity — and a verification that the approach is sound; a fused kernel that converts this into end-to-end acceleration is a systems effort we explicitly leave open.

6 Three Quantities, and the Boundary

A law this clean invites overclaiming. We state precisely what it does and does not buy, and we report the negatives we measured rather than the ones we hoped for.

Three different quantities. The law concerns keys *read* per query ($O(1)$). It does not by itself reduce two others: keys *found* (locating the right keys cheaply is cheaper than full scoring by a $1/B$ factor but still $O(n)$) and keys *stored* (the KV cache).

KV-cache eviction fails (measured). We tested whether the cache can be compressed by *evicting* keys — keep only a budget of KV entries by accumulated attention mass (the H2O/eviction approach) and drop the rest globally. Under our strict bar it **fails**: at $n = 1024$, keeping even 32% of the KV collapses top-1 agreement to 0.09 (Table 8). The reason is structural and consistent with the rest of the paper: *which* keys matter is query-dependent, so a key irrelevant to most queries is essential to some, and permanently evicting it destroys those queries. Read-time sparsity (skip per query) works; store-time eviction (drop globally) does not.

Table 8. KV-cache eviction *fails* under the top-1 bar (Qwen2.5-1.5B). Globally dropping keys collapses the output; *which* keys matter is query-specific.

n	KV kept	fraction	top-1
1024	100	10%	0.04
1024	196	19%	0.04
1024	324	32%	0.09

Measurement protocol. We report full top-1-versus-budget curves rather than single threshold crossings, and all $\kappa^\star$ values use natural, high-uniqueness text ($\geq 99\%$ unique tokens; repetitive text inflates top-1 agreement and is avoided). The frontier headline (§2) is additionally validated on a real distant needle, not on prose.

Remark 2 (What "cheaper compute" means here)**.** *The law saves* attention-read FLOPs *(large at long context) and cuts finding by a* $1/B$ *factor, but it does not shrink the KV cache, and attention is not the whole model (the MLP blocks are untouched). The* measured *wall-clock to date is a* slowdown*: our unfused reference selector is* $\sim 1.7\times$ *slower than FlashAttention (§5). Any end-to-end* speedup *is at this point a* projection from FLOP counts *(single-digit to low-tens*$\times$ *where attention dominates), contingent on a fused kernel we have not built — not a measured result. We state it as a projection, not a claim.*

7 Reconciliation with the Sparse Frontier

Nawrot et al. [2025] ("The Sparse Frontier") finds that budgets should *grow* sublinearly with n and that task difficulty sets attainable sparsity. We do not contradict this — we *refine* it. A constant *absolute* budget is precisely the limiting case of sublinear growth, and it is fully consistent with their observation that the usable *fraction* shrinks at longer n (our read fraction falls to 0.07% at 64K). What our existence/findability split adds is a clean separation of two objects their setup measures together: the smallest *sufficient* budget when the right keys are known (an oracle/top-k selector) — which is constant — versus the budget a *cheap heuristic* needs to *find* those keys on hard tasks — which grows, exactly as they report. Their task-dependence already foreshadows our counting/multi-hop boundaries; we make the mechanism behind it explicit. We do not merely assert this — we **measure the findability gradient directly** with our realizable block-summary selector (Table 9, Qwen2.5-7B, $n = 16$K). On single-fact needle, 1 block recovers the dense answer. On **multi-hop** (the answer requires chaining two buried facts), the same cheap selector needs far more: 4 blocks reach only 0.12 agreement with dense, 8 blocks 0.38, and 16 blocks 0.88 — a smooth, steep climb that has not closed the 0.99 bar even at 16 blocks. This is the existence/findability split made quantitative: the *condensed support* is constant, but the *cost of cheaply finding it* scales with how much cross-context chaining the answer demands. It is exactly the regime the Sparse Frontier observes, and it pinpoints where the open work lives — not in aggregation, and not in single-hop finding (both solved), but in cheap selection for *multi-hop* retrieval.

The phase diagram (existence constant; findability task-dependent) is the reconciliation, and the multi-hop curve is its measured signature.

Closing the multi-hop gap with cross-layer selection. The within-layer fix fails: iterative re-selection inside one attention layer (select, attend, augment the query, re-select) does *not* help (2-round $\approx$ 1-round). The reason is mechanistic — multi-hop is a *cross-layer* computation: an

Table 9. The findability gradient (measured): selector agreement with dense vs. task difficulty (Qwen2.5-7B, $n = 16$K). Single-fact is cheap (1 block). Multi-hop defeats the within-layer selector (plateaus at 0.88), but the *cross-layer* selector — deep-layer selection conditioned on the residual carrying the intermediate entity — clears the 0.99 bar.

task / selector	1 block	4 blocks	8 blocks	16 blocks
single-fact needle	**1.00**	–	–	–
multi-hop, within-layer	–	0.12	0.38	0.88
multi-hop, **cross-layer**	–	0.55	0.86	**0.99**

earlier layer writes the intermediate entity into the residual stream, and a later layer attends to the second fact. So we built the right fix: a **cross-layer selector** in which deep-layer block selection is conditioned on the residual that already carries the intermediate entity. This closes the gap (Table 9): on multi-hop at $n = 16$K the cross-layer selector reaches 0.55/0.86/**0.99** at 4/8/16 blocks, versus 0.12/0.38/0.88 for the within-layer selector — clearing the strict 0.99 bar where the single-pass selector plateaued at 0.88. Cheap selection therefore extends from single-fact to chained-fact retrieval, provided the selector follows the chain across depth.

8 Related Work

Fixed-pattern sparsity. StreamingLLM [Xiao et al., 2024] keeps sinks plus a local window; H2O [Zhang et al., 2023] evicts by heavy-hitter score. Both fix the *cache*; we show global eviction fails under a strict top-1 bar, and our claim is about per-query *reads*, not a shared fixed cache. **Query-aware selection.** Quest [Tang et al., 2024] and MInference [Jiang et al., 2024] select blocks per query/pattern; closest to our router, but they report perplexity/accuracy, not a constant *count* under exact next-token fidelity, and not the shrink-with-scale floor. **Scaling of sparsity.** The Sparse Frontier [Nawrot et al., 2025] introduces sparse-attention scaling laws and argues budgets must grow; we reconcile this as existence-vs-findability (§7). **Mechanism.** We connect constant support to token condensation [Geshkovski et al., 2023], which to our knowledge is unclaimed as the explanation. **Lower bounds.** An $\Omega(nd)$ worst-case KV bound exists; our claim is average-case empirical over natural inputs and is consistent with it (and with our negative eviction result). We make no "first constant cache" claim — StreamingLLM/H2O predate that framing.

9 Implications: What the Law Buys, and How We Use It

The constant-support law is not only a description of frozen models — it is a design constraint we build on at Brainsless Research Lab. We state the implications precisely, separating what the law *licenses* from what still requires engineering (the read/find/store split of §6).

For long-context inference. Because the *condensed* support is $O(1)$, the aggregation cost of attention at long context is bounded by a constant per query, and our $O(n)$-memory kernel realizes that bound without forming the $n \times n$ matrix. The practical consequence is that the marginal token at 64K costs essentially the same aggregation as at 8K — the read fraction falls to 0.07% at 64K — so long sessions become economically flat in attention reads rather than quadratic. The unlocked work is a cheap *approximate* selector (the law says the target is small; finding it is the remaining cost), which our negative result on a naive selector (§5) shows is the right place to invest.

For long-context system design. The law licenses one architecture-agnostic consequence. Because the condensed support is constant per query, a long session is, in attention terms, an *indexing-and-finding* problem rather than a quadratic-attention one: the principled move for production long-context is an external retrieval index over history (find the small relevant set cheaply) plus constant-budget aggregation, rather than an ever-wider dense attention window. We state this as a design implication, not a deployment result; every number in this paper is on frozen public models.

For training our next model. The phase picture (§4) suggests a training signal: encourage early condensation (lower deep-layer P) to shrink $\kappa^\star$ further, while *protecting* the de-condensed retrieval heads that carry long-range information (the minority that makes eviction fail). A condensation-aware objective — regularizing participation ratio per head toward the measured two-population structure — is the concrete next experiment the law points to.

For the field. The transferable claims: (i) long-context serving should budget attention *reads* as a constant and invest the saved compute in cheap key-*finding*; (ii) KV-cache compression by global eviction is a dead end under strict fidelity (our measured negative), so memory should be addressed by external retrieval over history, not by dropping keys; (iii) participation ratio is a cheap, model-internal diagnostic for "how long-context-ready" a checkpoint is.

10 Limitations

- **Aggregative tasks (the real scope boundary).** The law holds for sparse/pointwise next-token dependence; on genuinely aggregative tasks (e.g. counting) $\kappa^\star$ grows with n while P stays flat (§4, Table 6). This is the law's true domain, stated precisely.
- **Memory, not just FLOPs.** The KV cache is not compressed (eviction fails); the win is read-FLOPs plus a $1/B$-cheaper (still $O(n)$) finding pass, not cache size (§6).
- **Wall-clock.** We report the $O(n)$-memory and FLOP reduction; a *fused* kernel beating FlashAttention end-to-end is a systems effort we did not complete (our reference selector is unoptimized and slower).
- **Frontier 72B context.** The 72B distant-needle is measured at 32K — its *full* trained context, so the result is valid at the model's maximum length but not pushed beyond it. The million-token flatness is shown separately on the 1M-trained 7B (§2); combining 72B with 1M would require a 72B-class model trained past 32K.

11 Conclusion

A frozen transformer's exact next token depends on a small, constant number of keys — ≈ 16 at scale — independent of context length, and that constant is enough to recover a unique fact buried in a long context on a 72B model. The mechanism is token condensation; the boundary is that the cache cannot be globally compressed. The result is a measured law with a clear theory and an honest cost: $O(n)$-memory attention with a constant-factor cut in reads, not a sub-quadratic miracle.

Compute and availability. The campaign spanned eleven models across three lineages, from 135M to 72B parameters and to million-token context, on a large-scale multi-node GPU cluster (NVIDIA H100, H200, and B200 accelerators); the largest models ran on multi-GPU B200 nodes. No frontier number is presented as measured that was not. Code and data are available from the authors on request.

References

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36, 2023.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, et al. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, 2024.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M. Ponti. The sparse frontier: Sparse attention trade-offs in transformer LLMs. *arXiv preprint arXiv:2504.17768*, 2025.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Experimental Details

Models (frozen, public). All weights are public and unmodified; nothing is trained. Scale ladder and mechanism probes: `HuggingFaceTB/SmolLM2-135M/360M/1.7B-Instruct`, `Qwen/Qwen2.5-0.5B/1.5B/3B/7B-Instruct`. Frontier: `Qwen/Qwen2.5-7B-Instruct` (8K–64K $\kappa^\star$ sweep, selectors, kernel timing), `Qwen/Qwen2.5-14B-Instruct` (sub-16 needle floor), `Qwen/Qwen2.5-72B-Instruct` (500-trial distant needle, margin-match), `meta-llama/Llama-3.2-1B/3B`, `Llama-3.1-8B`, `Llama-3.3-70B` (second-family ladder), and `Qwen/Qwen2.5-7B-Instruct-1M` (128K–1M sufficiency check). Frontier and second-family runs were executed on a multi-node GPU cluster (NVIDIA H100, H200, and B200), the largest models on multi-GPU B200 nodes. Run configurations and outputs are available from the authors on request.

Method. The sparse path is a patched attention forward (shared selectors, masks, metrics) over frozen weights; experiments cover the theorem hypotheses, the needle and router evaluations, the kernel, the instruction-dense and phase-diagram probes, and the scale sweep.

Metrics. top-1: fraction of positions whose argmax matches the full model; last-KL: KL(full‖sparse) on the last token (nats); mass-weighted recall: needle-key retention weighted by true per-head needle mass; cost: keys scored per query.