UNDERSTANDING CONTEXT UTILIZATION VIA NEURON SINKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer language models apply the same computational blocks at every position, although causal masking limits the context available near sequence beginnings. We identify neuron sink, a phenomenon across diverse language models in which a sparse subset of MLP neurons exhibits disproportionately frequent strong activations near sequence beginnings. This pattern reappears at the start of a passage following an irrelevant prefix, suggesting that neuron sink depends on the availability of useful context rather than absolute position or context length. Ablating sink neurons selectively impairs prediction near sequence beginnings, demonstrating their functional importance for low-context processing. Beyond this role, we find evidence that neuron sink contributes to behavioral position bias. In a controlled model, transplanting the sequence-initial sink-neuron state to middle positions reduces first-over-middle retrieval bias, causally linking this activation bias to the primacy component of the lost-in-the-middle phenomenon. These findings uncover a mechanism for low-context processing and identify a potential target for mitigating position bias in language models.

1 INTRODUCTION

A causal language model applies the same Transformer blocks at every token position, even though the information available for prediction changes across the sequence (Vaswani et al., 2017). For example, consider "*She picked up her umbrella before stepping outside into the rain.*" When predicting "*umbrella*", the short prefix "*She picked up her*" provides few clues about the situation. By the time the model predicts "*rain*", the preceding context has established a scene that more strongly constrains the continuation. Shared weights may therefore support different internal computations as the available context changes. Are these differences reflected in which neurons become active?

To investigate this possibility, we compare the activation patterns of individual MLP neurons at different positions within natural-text sequences. Across pretrained language models, we find a sparse set of neurons whose strong activations occur far more often near sequence beginnings than at later positions (Figure 1b). We refer to these neurons as **sink neurons** and the associated phenomenon as **neuron sink**. Zeroing the sink neurons sharply increases next-token loss near the sequence beginning, while its effect at later positions is much smaller. These findings identify sink neurons as a sparse population whose activity and contribution to prediction are both concentrated near sequence beginnings.

We next ask: **what gives rise to neuron sink?** Because sink activity is concentrated near sequence beginnings, we first examine whether it arises from features associated with these positions: the presence of a BOS token, the distinctive text distribution at document openings, or the disproportionate attention that early tokens receive through attention sink. Our controls show that none of these features is required for neuron sink.

Returning to our motivating question of how neuronal activity varies with the context available for prediction, we then test whether sink activity reflects a shortage of useful preceding context. To this end, we take a passage from within a document and replace its preceding context with unrelated text of the same length. Neuron sink emerges at the boundary between the unrelated prefix and the target passage, but is absent at the same position with the original context. This recurrence suggests that sink neurons respond to a shortage of useful information in the preceding context: even deep within a long sequence, an unrelated prefix can leave a passage with little predictive context, much like a sequence beginning.

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

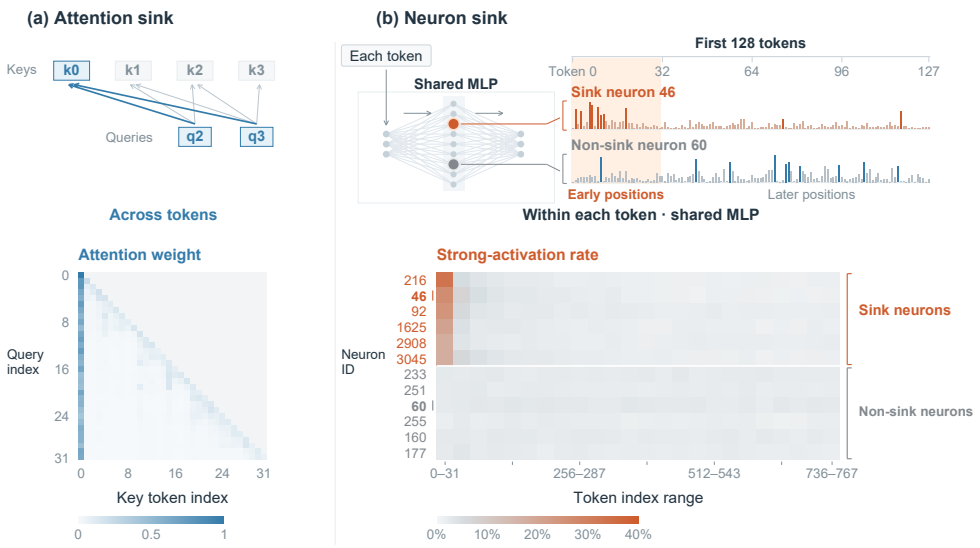

Figure 1: **Attention sink and neuron sink. (a)** Attention sink concentrates attention from many queries on a few early keys (Xiao et al., 2024). **(b)** Neuron sink concentrates strong activations of specific MLP neurons near sequence beginnings. The upper traces illustrate where larger activations occur in a sink and a nonsink neuron within one sequence. The lower heatmap reports strong-activation rates across 256 sequences, averaged within each 32-token interval. Measurements use layer 7 of Qwen3-0.6B-Base on RefinedWeb.

In KV retrieval tasks, their activity also increases at queries whose required evidence is absent. Our useful-context hypothesis predicts that sink activity should extend over more positions when useful context becomes available later. We test this in a synthetic copy task that starts with a block of random tokens and then repeats it. Lengthening the random block delays the first predictable token and generally broadens the initial sink peak. Together, these results suggest that sink activity tracks the availability of useful context rather than absolute position alone.

Because MLP outputs contribute to token representations, neuron sink can make the states of early tokens systematically different from those of later tokens. These differences may affect how the model subsequently uses information from the beginning and later parts of the context. This suggests a possible connection to the preference for beginning over middle information observed in the lost-in-the-middle phenomenon (Liu et al., 2024). In a controlled retrieval model, copying sink-neuron activations from beginning positions to matched middle positions reduces the first-over-middle accuracy gap without changing the input or model parameters. This provides causal evidence that sink-neuron activity can contribute to behavioral position bias in this controlled setting.

In summary, our contributions are:

1. **We identify neuron sink in pretrained language models.** Across model scales and families, a sparse set of MLP neurons becomes strongly active disproportionately often near sequence beginnings, and ablating these neurons primarily harms early-token prediction.
2. **We show that sink activity responds to useful preceding context.** Across natural-text discontinuities, controlled synthetic dependencies, and evidence-removal interventions, sink activity follows whether the preceding context contains information useful for the current prediction rather than absolute position alone.
3. **We provide causal evidence connecting neuron sink to behavioral position bias.** In a controlled two-layer model, copying beginning-position sink-neuron activations to the middle reduces the model's preference for beginning over middle information.

2 Related Work

2.1 Neuron-Level Specialization and Context Dependence

MLP neurons have been studied as key–value memories and content-selective features (Geva et al., 2021; Dai et al., 2022; Gurnee et al., 2023), as well as positional and confidence regulators (Gurnee et al., 2024; Stolfo et al., 2024). Voita et al. (2024) use activation–position mutual information to identify binary ReLU patterns in OPT, including broad ranges spanning hundreds to over a thousand

tokens. Our focus is strong-activation enrichment within the first few dozen tokens in gated-MLP models, and its recurrence when useful preceding context is disrupted. By varying context while holding position fixed, we connect this localized activity to contextual support for prediction, extending work on predictive context features (O'Connor & Andreas, 2021).

2.2 Attention Sinks and Activation Outliers

Attention sinks concentrate attention from many queries on early keys even when their content is not semantically important (Xiao et al., 2024). Proposed explanations include softmax normalization and learned key biases (Gu et al., 2025), and avoidance of excessive token mixing (Barbero et al., 2025); query-dependent output gating can strongly attenuate the first-token sink (Qiu et al., 2025). Massive activations are extreme residual-stream values that can act as implicit biases and attract attention (Sun et al., 2024). Neuron sink instead measures how often MLP coordinates before the output projection exceed their own magnitude thresholds across positions, distinguishing it from attention concentration and globally extreme activation magnitude.

2.3 Position Bias and Context Use

Language models often retrieve evidence near the beginning or end of a prompt more reliably than in the middle (Liu et al., 2024). Wu et al. (2025) analyze how causal masking, depth, and positional mechanisms shape this bias. Following their data-generation and paired evaluation protocol, we test whether beginning-associated MLP activations contribute to first-over-middle preference in a controlled two-layer model.

3 Preliminaries and Measuring Neuron Sink

3.1 MLP Neurons in Transformer Models

We measure MLP activation values immediately before the output projection in each Transformer block (Vaswani et al., 2017). A *neuron* is one coordinate of this intermediate vector: $m_{t,c}^{(\ell)}$ denotes its activation value at token position t, layer ℓ, and coordinate c. In a gated MLP, this is the product of the activated gate and the up-projected value (Shazeer, 2020); in an ungated MLP, it is the post-activation coordinate.

3.2 Definition and Measurement of Neuron Sink

Let r denote the token positions under examination, which may be a single position or a span of positions. Our measurement asks: *is a neuron much more active within r than it is on average across the sequence?*

Defining activation. For each neuron, we set a threshold $\theta_c^{(\ell)}$ at the $(1-q)$-quantile of its absolute activation values across calibration sequences and token positions, where q is a hyperparameter (default $q = 0.02$). We refer to activations whose absolute values meet or exceed this threshold as *strong activations*. We define the neuron as *active* at a token when such a strong activation occurs.

Relative activation rate. For a fixed neuron c in layer ℓ, $p_c^{(\ell)}(r)$ is its activation rate *within* r: the number of tokens in r at which this neuron is active, divided by the total number of tokens observed in r. We pool both counts across all evaluated sequences. The baseline $b_c^{(\ell)}$ is the same neuron's activation rate *across all positions*: the number of tokens at which it is active anywhere in the sequences, divided by the total number of tokens in those sequences. The *relative activation rate* divides the rate within r by this baseline:

$$F_c^{(\ell)}(r) = \frac{p_c^{(\ell)}(r)}{b_c^{(\ell)}}. \tag{1}$$

The $1\times$ baseline corresponds to the sequence-wide average; $F > 1$ means that the neuron activates more often in r.

Selecting sink neurons. We select sink neurons based on their activity over the first w tokens of each sequence, where w is the selection window size. Let ρ be the fraction of each layer's MLP neurons selected as sink neurons; the defaults are $w = 32$ and $\rho = 0.01$. In the main pretrained-LM analysis, we rank eligible neurons *within each layer* by the relative activation rate over the first w tokens. We select the highest-ranked neurons as *sink neurons*, with the number selected determined by ρ. Our default $\rho = 0.01$ selects 1% of the neurons in each layer for analysis; it does not imply that only 1% exhibit neuron sink. Unless otherwise stated, comparisons within a model



Figure 2: Neuron sink and its effect on prediction. Sink neurons are selected within each layer using the first 32 tokens. (a) Within-layer mean relative activation rate at the beginning in Qwen3-0.6B-Base; the dashed curve marks the selection cutoff. (b) Rates of the same neurons in consecutive 32-token windows, averaged within each layer; the first 512 tokens are shown. (c) Four Qwen3-Base scales. (d) Three model families. In (c,d), points denote layers, and diamonds and whiskers show the median and interquartile range. Activation-rate axes in (a,c,d) use a base-2 logarithmic scale. Dotted lines mark the $1\times$ sequence-wide baseline (zero on the $\log_2$ color scale in (b)). (e) Held-out next-token loss change after ablating sink neurons or 20 layer- and count-matched random neuron sets; the random mean and one standard deviation are shown in blue.

use sink neurons selected by this procedure, with neuron identities and activation thresholds held fixed across conditions. Appendix A specifies calibration, eligibility, ranking, and the experiment-specific exceptions.

4 A Sparse Subset of MLP Neurons Activates Preferentially Near Sequence Beginnings

Using the relative activation rate defined above, we examine where sink-neuron activity is concentrated and how this pattern varies across layers and models. We begin with Qwen3-0.6B, extend the analysis to other model scales and families, and then use ablation to assess the contribution of sink neurons to early-token prediction.

4.1 A Representative View of Neuron Sink

We evaluate Qwen3-0.6B-Base (Qwen Team, 2025) on 256 RefinedWeb sequences of 2,048 tokens each (Penedo et al., 2023). Following Section 3.2, we select 31 neurons in each of the 28 layers based on their relative activation rates in the first 32 tokens. All layers show elevated rates over this span (Figure 2(a)). The median across layers of the within-layer mean relative activation rate is $7.67\times$ the sequence-wide baseline.

To examine how activity changes with position, we evaluate these same 31 neurons per layer in consecutive 32-token windows. Their profiles peak sharply in the initial window and rapidly approach baseline at later positions (Figure 2(b)).

4.2 Widespread Across Layers, Scales, and Families

We repeat selection separately for each checkpoint: Qwen3-Base at 0.6B, 1.7B, 4B, and 8B parameters, followed by a family comparison of Qwen3-8B-Base, Llama-3.1-8B (Grattafiori et al., 2024), and Mistral-7B-v0.3 (Jiang et al., 2023). All four Qwen3 scales show elevated beginning activity across model depth (Figure 2(c)). Across-layer medians range from $7.00\times$ to $8.62\times$ the sequence-wide baseline; the strength varies between layers and is not monotonic in model size.

The pattern also appears in all three tested families, whose across-layer medians range from $4.74\times$ to $7.13\times$ the baseline (Figure 2(d)). Appendix B gives model and sampling details, together with individual-sequence and leave-one-out checks for Qwen3-0.6B.

4.3 Sink Neurons Causally Support Early Prediction

We next test whether sink neurons causally support early prediction. On held-out text, we use zero ablation (Heimersheim & Nanda, 2024), comparing next-token loss with and without zeroing the selected sink neurons immediately before the MLP down projection. The neuron set is fixed before this evaluation, and all selected coordinates are zeroed at every input position in all 28 layers during the same forward pass. We also evaluate 20 random neuron sets matched to the sink set's count in each layer. The readout is the increase in negative log-likelihood, $\Delta\mathrm{NLL}$, at each target position. Appendix C specifies the held-out data, controls, and loss aggregation.

As shown in Figure 2(e), ablating sink neurons increases loss sharply over the first 32 predictions, with an effect substantially larger than ablating any matched random neuron set. The effect then decays rapidly with position; from position 129 onward, it is small and falls within the variability of the random controls.

5 Sink Neurons Respond to Useful Preceding Context

Beginning-selective activity and its contribution to early prediction do not establish what drives neuron sink. We test whether limited useful context explains this activity by examining alternative explanations, comparing equally long related and unrelated prefixes, delaying predictability in a copy task, and removing evidence at late retrieval queries.

5.1 Testing Explanations Tied to Sequence Beginnings

We first test whether neuron sink requires an explicit boundary token, document-opening content, or conventional attention sink.

Not Determined by an Explicit Sequence-Boundary Token. To test whether an explicit boundary token is required, we compare the same text with and without a prepended `<|endoftext|>` in Qwen3-0.6B, evaluating the same first 32 text tokens.

We use the exact sink-neuron set from the main experiment in both conditions. Median relative activation rate is $7.67\times$ without the inserted token and $7.45\times$ with it, with all 28 layers remaining above the sequence-wide baseline. Thus, an explicit sequence-boundary token is not necessary for the response.

Not Reducible to Document-Opening Content. Titles, metadata, and introductory phrases may be overrepresented at document beginnings. Neurons selective for these patterns could therefore appear to prefer early positions. If this explains neuron sink, the response should largely disappear when ordinary text from inside a document starts a new input.

In Qwen3-0.6B, we compare two sets of 2,048-token inputs constructed from the same 256 long documents. Each set contains one segment per document: the first uses segments starting at the document beginning, while the second uses segments starting at randomly chosen positions within each document. We use an independently calibrated and selected set of 31 neurons per layer, fixing their identities and thresholds across the two conditions.

Median relative activation rate is $6.74\times$ at genuine document starts and $4.24\times$ after mid-document restart; the latter remains well above $1\times$. Document-opening content is therefore not required for neuron sink.

Neuron Sink Persists in an Attention-Sink-Free Checkpoint. Neuron sink could be a companion effect of attention sink: the model adaptations that make early tokens attract disproportionate attention might also produce their distinctive MLP activation pattern. If neuron sink requires the conventional attention sink, it should disappear in a model without that attention pattern.

We evaluate the two released checkpoints from Qiu et al. (2025): their standard-attention baseline and their elementwise SDPA-output-gated model. The source study reports a conventional first-token attention sink in the baseline and an attention-sink-free pattern in the gated checkpoint. On identical RefinedWeb inputs, both checkpoints retain an MLP population with elevated activation rates near the beginning: geometric-mean relative activation rates are $7.19\times$ and $5.12\times$, respectively. Neuron sink thus persists in the checkpoint reported to lack attention sink, arguing against an explanation that requires the latter.

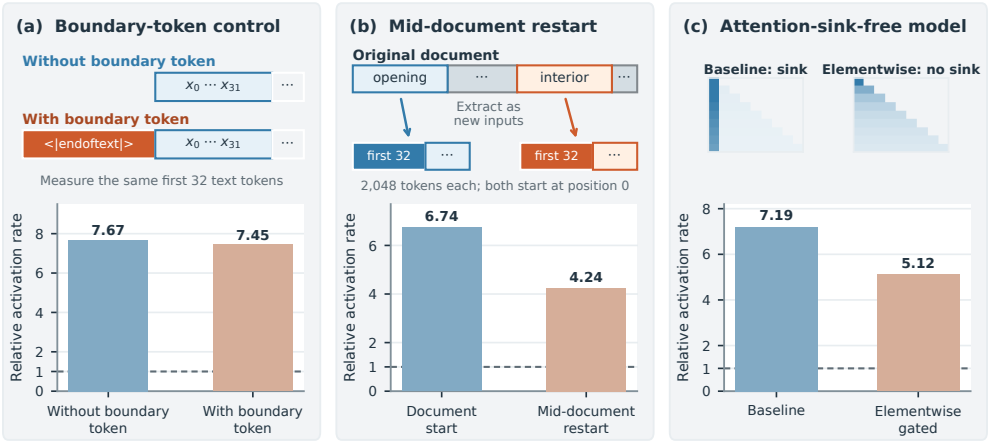

Figure 3: Controls for alternative explanations of neuron sink. (a) The main-experiment neurons retain elevated activation rates both without and with Qwen3-0.6B's explicit <|endoftext|> boundary token. (b) Sink neurons in each layer respond at document starts (6.74×) and at the start of inputs that begin mid-document (4.24×). Bars in (a–b) report the median across layers of arithmetic population-mean relative activation rate. (c) Both the standard-attention and elementwise SDPA-output-gated checkpoints from Qiu et al. (2025) retain neuron sink; bars show geometric means across layers. Attention maps schematically illustrate the source study's reported presence and absence of attention sink.

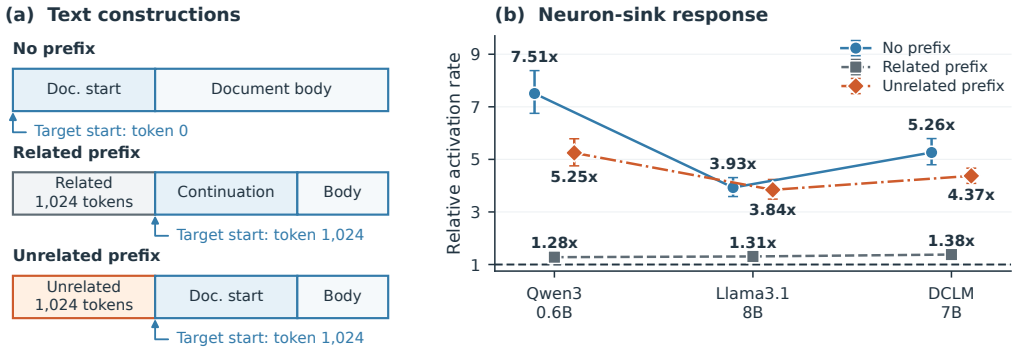

Figure 4: When the prefix is no longer relevant, the neuron-sink pattern observed at the beginning of a sequence reappears. (a, left) Arrows mark where evaluation begins. Without a prefix, we evaluate the beginning of a document at position 0. After a 1,024-token prefix, we evaluate either the next part of the same document or the beginning of a different, unrelated document. In both prefixed conditions, evaluation begins at position 1,024 (zero-based positions). (b, right) With approximately 1% of neurons per layer selected separately in each condition, the related-prefix continuation remains near baseline, whereas the unrelated-document start recovers much of the no-prefix response. Points are geometric relative activation rates, error bars are 95% layer-bootstrap intervals, and the dashed line marks the 1× sequence-wide baseline.

5.2 Prefix Relevance Supports a Useful-Context Explanation

The preceding controls leave open why sink activity favors early positions. Placing a document beginning after an unrelated prefix tests whether the response can recur away from the sequence origin. Comparing it with a same-document continuation after an equally long prefix further separates prefix relevance from position and length (Figure 4(a)). Both prefixed targets start at position 1,024, but their preceding text differs in the information it can provide for prediction.

With a related prefix, the pronounced neuron-sink response is largely absent; with an unrelated prefix, it reappears at the same late position (Figure 4(b)). A separate paired comparison fixes both target tokens and beginning-selected neurons: replacing the related prefix with unrelated text raises target-start activation rates by factors of 1.52 in Qwen3-0.6B and 1.69 in Llama-3.1-8B (Appendix E). These results motivate a useful-context explanation: sequence beginnings are one setting in which predictions have little support from preceding text, and similar activity can recur later when that support is limited.

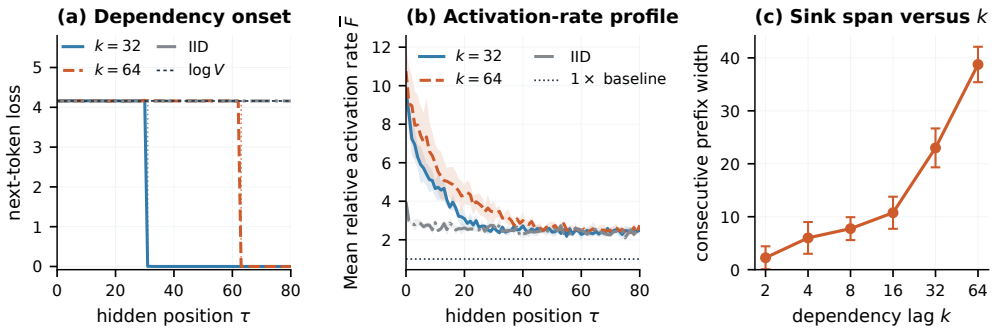

Figure 5: Delaying predictable tokens in the k-lag copy task. Lines and points show four-seed means; bands and error bars show one standard deviation across seeds. (a) Hidden position τ predicts token $\tau + 1$; loss drops at $\tau = k - 1$ in the copy tasks. (b) Mean relative activation rate $\overline{F}$ over neurons selected separately at each position; the dotted line marks the $1\times$ sequence-wide baseline. (c) Consecutive initial positions satisfying $\overline{F} \geq \mu_{\mathcal{B}} + 2\sigma_{\mathcal{B}}$, computed separately for each seed using its mean and standard deviation over positions 256–511. The span grows over $k = 2$–64.

5.3 Delaying the First Predictable Token Extends Sink Activity

The useful-context explanation predicts that delaying the first predictable token should extend elevated sink activity to later positions. We test this prediction in a k-lag copy task.

The first k tokens are sampled independently and uniformly; every later token copies the token k positions earlier. For $k = 4$, one possible sequence is

$$\underbrace{\texttt{A B C D}}_{\text{four random tokens}} \;\Big|\; \underbrace{\texttt{A B C D A B C D ...}}_{\text{copy from four positions earlier}} .$$

The first copied token is predictable once the model has read the initial four tokens. Increasing k delays this first predictable token. For each lag and each of four seeds, we train a separate two-layer decoder-only Transformer from scratch, with vocabulary size $V = 64$ and 512 prediction positions.

An independent and identically distributed (IID) control samples every token independently, so its preceding context never helps predict the next token.

We measure how far neuron sink extends. At each position τ, we select up to ten layer-0 neurons with the largest relative activation rates $F > 1$ and denote their arithmetic mean by $\overline{F}(\tau)$, plotted in Figure 5(b). Each neuron's F uses its sequence-wide activation rate as the baseline, as in Eq. 1.

The *sink span* counts consecutive initial positions satisfying $\overline{F}(\tau) \geq \mu_{\mathcal{B}} + 2\sigma_{\mathcal{B}}$, where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are the mean and population standard deviation of $\overline{F}$ over later positions $\mathcal{B} = \{256, \ldots, 511\}$. Counting stops at the first position below this threshold. We compute each seed's span separately before averaging; Appendix F gives the selection details.

Figure 5(a) verifies that next-token loss drops at the first predictable target. As predicted, the initial activation-rate peak generally broadens as this transition is delayed (Figure 5(b–c)): mean sink span rises from 2.25 positions at $k = 2$ to 38.75 at $k = 64$. IID remains unpredictable throughout but has only a brief initial elevation relative to its own late-region baseline. The span does not locate the predictability transition exactly, but its increase supports the prediction that delaying useful context extends elevated activity to later positions.

5.4 Extending the Explanation to Retrieval Queries

We next apply the useful-context explanation to predict sink-neuron responses beyond sequence or passage beginnings. A late query may have a long context containing many facts, yet lack the evidence needed for its answer. Beginning-selected neurons should therefore become more active when this evidence is unavailable. We test this extension in Qwen3-0.6B-Base by changing access to the required evidence while holding the model and late query fixed.

Key–value retrieval has been used to study context use (Liu et al., 2024). In our Novel-KV retrieval task, each 2,048-token prompt contains 408 random single-token key–value mappings followed by a query for one key. The queried mapping appears near the early, middle, or late part of the table. For example, a mapping $k^* \to v^*$ supports the final query for k^*.

For each of 2,000 base prompts, we form three conditions (Figure 6(a)). In *visible*, the queried mapping remains available. In *replaced*, only its key is changed to an unused key, so the original

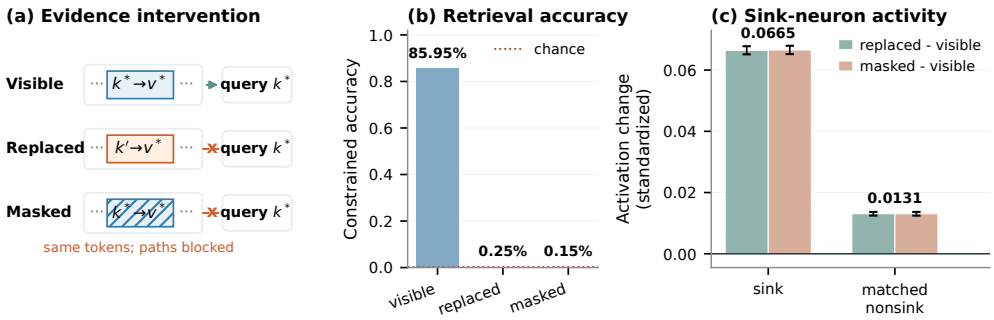

Figure 6: Evidence availability and sink-neuron activity at the query. (a) The required mapping is visible, replaced, or masked. (b) Retrieval accuracy falls to chance when evidence is unavailable. (c) Mean standardized activation magnitude increases more for sink neurons than for the matched nonsink population. Bars show the mean difference between unavailable and visible evidence for paired prompts; error bars are paired-bootstrap confidence intervals.

query no longer has a matching mapping. In *masked*, every prompt token is unchanged, but the queried mapping is blocked from attention by all subsequent tokens, including the final query. This blocks both direct access and relay through intervening tokens. We measure retrieval accuracy and sink-neuron activity at the final query position, comparing each unavailable-evidence condition with its own visible counterpart. Retrieval uses a restricted candidate set; accuracy falls from 85.95% with visible evidence to 0.25% and 0.15% after replacement and masking, respectively, close to the 0.245% chance level (Figure 6(b)).

At the final query position, we measure the mean absolute activation of a fixed sink population selected at natural-text beginnings, standardizing each neuron using its mean and standard deviation on separate calibration data. We compare this readout with a nonsink population matched by layer, count, and calibration activation statistics (Appendix G).

When the evidence is replaced or masked, mean standardized activation magnitude rises by about 0.0665 for sink neurons, versus 0.0131 for the matched nonsink population (Figure 6(c)). Thus, sink neurons respond more strongly to unavailable evidence than the matched controls. The similar results for replacement and masking support this interpretation, since masking leaves the prompt text unchanged.

This result supports extending the useful-context explanation to individual retrieval queries: beginning-selected neurons respond to the availability of evidence for the current prediction, even when its position and surrounding text remain unchanged under masking.

6 Copying Sink-Neuron Activations Reduces First–Middle Bias

The beginning-biased activity of sink neurons makes early-token representations systematically different from those at later positions. This raises the question of whether such representational differences contribute to position bias in retrieval, particularly the beginning-over-middle advantage observed in *lost in the middle* (Liu et al., 2024). To test this possibility, we copy beginning-position sink-neuron activations to middle positions in a controlled two-layer model and examine whether this reduces the first–middle retrieval gap. We report retrieval accuracies at both positions to distinguish gap reduction from improved middle retrieval.

6.1 Controlled Evaluation and Causal Intervention

We use a separate two-layer Transformer on a synthetic content–label retrieval task, following the data-generation and paired first–middle evaluation protocol of Wu et al. (2025). Appendix H gives model and selection details. Each input contains eight content–label pairs followed by a query:

$$(x_1, y_1, \ldots, x_8, y_8, x_q). \tag{2}$$

The two evaluation conditions share an underlying sampled context. In `first_vs_middle`, the query is generated from the first pair's original content class, and its content vector x_1 is copied to the fifth pair while both labels are retained; the target is y_1. In `middle_vs_first`, the construction is reversed: the query comes from the fifth pair's original class, x_5 is copied to the first pair, and the target is y_5. The query is a noisy realization of the selected class. Appendix H details the pairing.

We define first-over-middle preference as

$$\text{Bias}_{\text{first_vs_middle}} = \text{Acc}(\text{first_vs_middle}) - \text{Acc}(\text{middle_vs_first}). \tag{3}$$

Positive bias means that first-target accuracy exceeds middle-target accuracy in the paired evaluation. On 4,096 training-distribution examples used for neuron selection, we measure how the activations of each MLP neuron are distributed across positions. We then select eight sink neurons from the first Transformer block, whose activation distribution constitutes an internal position bias.

To test whether this internal position bias causally affects model behavior, we use a within-sequence variant of activation patching (Meng et al., 2022; Zhang & Nanda, 2024), copying beginning-position activations of the selected neurons to the corresponding middle positions within the same model, input, and forward pass:

$$m^{(0)}_{t(x_5),c} \leftarrow m^{(0)}_{t(x_1),c}, \qquad m^{(0)}_{t(y_5),c} \leftarrow m^{(0)}_{t(y_1),c}, \tag{4}$$

where $t(x_i)$ and $t(y_i)$ denote the token positions occupied by the content and label of pair i, respectively, and c indexes the selected sink neurons. We apply the replacement before the first-layer MLP output projection, leaving all other coordinates at that site unchanged. Clean and patched runs use the same inputs and weights. As a control, we apply the identical operation to five random nonsink neuron sets matched by layer and count.

6.2 Effect on the First–Middle Accuracy Gap

The sink patch reduces the first–middle accuracy gap by 2.00 percentage points, or 11.5% of the clean gap (Table 1). Middle-target accuracy increases by 1.54 points, while first-target accuracy decreases by 0.46 points. Random-neuron patches increase the gap by 1.75 points on average. The reduction therefore includes improved middle retrieval, with an average effect opposite to that of the random controls. Appendix H reports each control.

Table 1: Retrieval after activation copying. Accuracy is in %; Gap (First − Middle) and changes from Clean (arrows) are in percentage points. Random patch averages five sets.

Setting	First acc.	Middle acc.	Gap
Clean	84.28	66.87	17.41
Sink patch	83.82 (↓ 0.46)	**68.41** (↑ 1.54)	**15.41** (↓ 2.00)
Random patch	84.92 (↑ 0.64)	65.76 (↓ 1.11)	19.16 (↑ 1.75)

Together with the opposite average effect of matched random-neuron patches, these results support a causal contribution of sink neurons to first–middle bias in this controlled model.

7 Discussion and Conclusion

Sink Neurons Respond to Limited Useful Context. Neuron sink suggests that positional patterns in MLP activity can reflect the information available for prediction. Sequence beginnings provide little preceding context, but a long prefix can also offer limited support when it is unrelated to the target or lacks the required evidence. The prefix and retrieval experiments, together with the controlled predictability manipulation, support useful-context availability as an explanation for this activity.

Sink Activity Can Influence Retrieval Position Bias. The retrieval intervention extends the analysis from how preceding context supports prediction at a token to how information at that token is used by a later query. Copying beginning-associated sink-neuron activations to middle positions reduces first-over-middle preference in the controlled model. This establishes that MLP activity at context positions can contribute to a later retrieval asymmetry, motivating a neuron-level perspective on behavioral position bias.

Limitations and Future Directions. Three questions remain. First, prefix relevance, predictability, and evidence availability capture different aspects of useful context; their commonalities and variation across neurons need clarification. Second, sink neurons' computations and their effects on later retrieval remain unresolved. Interventions across layers could separate contributions to prediction from downstream information use. Third, the retrieval effect is partial and established in a controlled two-layer model. Tests in pretrained long-context LMs should assess improvements to middle-position retrieval, preservation of early-token prediction, and variation across tasks and context lengths.

Reproducibility Statement

The code is provided in the supplementary material. The models and datasets used in our experiments are documented in detail in the appendix and the accompanying code to facilitate reproducibility.

AI Use Statement

We used generative AI tools to improve the clarity and grammar of the manuscript, assist with implementing our research ideas in code, and help identify and locate relevant prior work. The authors manually reviewed all AI-assisted edits, checked and tested the AI-assisted code, and verified the citations and descriptions of related work by consulting the original publications. We did not use generative AI in any other stages of this work, including research ideation, experimental design, data generation or processing, or the analysis and interpretation of results. The authors take full responsibility for the final content of the paper, including the implementation and reported results.

References

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do LLMs attend to the first token? In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=tu4dFUsW5z.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, pp. 97114–97144, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f1b04face60081b689ba740d39ea8f37-Abstract-Conference.html.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=JYs1R9IMJr.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ZeI104QZ8I.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024. URL https://arxiv.org/abs/2404.15255.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262.

Joe O'Connor and Jacob Andreas. What context features can transformer language models use? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 851–864. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.70. URL https://aclanthology.org/2021.acl-long.70/.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems*, volume 36, pp. 79155–79172. Curran Associates, Inc., 2023. doi: 10.52202/075280-3464.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, volume 38, pp. 100092–100118. Curran Associates, Inc., 2025. doi: 10.52202/085713-3345.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Alessandro Stolfo, Ben Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. Confidence regulation neurons in language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 125019–125049. Curran Associates, Inc., 2024. doi: 10.52202/079017-3970.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=F7aAhfitX6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1288–1301. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.75. URL https://aclanthology.org/2024.findings-acl.75/.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the emergence of position bias in transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67756–67781. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25ad.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, pp. 21875–21895, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, pp. 1651–1678, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

A Measurement and Selection Protocols

This appendix supplements Section 3.2 with the implementation of calibration and selection. Table 2 identifies the experiments that depart from the default protocol; the following appendices give their data, interventions, and aggregation rules in main-text order.

Calibration and event counting. We measure the MLP intermediate activation $m_{t,c}^{(\ell)}$ defined in Section 3, immediately before the output projection. In the main pretrained-model analysis, thresholds are estimated from 4,096 sample–token positions drawn without replacement from the 256 length-2,048 calibration sequences. For each neuron, we take the 98th percentile of its absolute activation at these positions. A second pass over the same input IDs counts events satisfying $|m_{t,c}^{(\ell)}| \geq \theta_c^{(\ell)}$, including ties. Calibration, selection, and the descriptive position profiles therefore use the same corpus; the ablation evaluation uses separate text.

The activation rates $p_c^{(\ell)}(r)$ and $b_c^{(\ell)}$ follow Section 3.2: they pool active-token counts and total token counts across sequences, within region r and across all positions, respectively. Their ratio is the relative activation rate $F_c^{(\ell)}(r)$. We use the measured $b_c^{(\ell)}$ to account for threshold ties and changes of evaluation data. A single-position profile uses $r = \{t\}$; a window profile pools counts within that window before computing $F_c^{(\ell)}(r)$.

Selection and aggregation. For the main pretrained-model profiles, eligible neurons have $b_c^{(\ell)} > 10^{-4}$ and at least 30 events in the first 32 tokens. We rank them within each layer by the first-32-token relative activation rate, breaking ties by neuron index. We retain 1% of the MLP width, rounding the count up to an integer. The positional profiles use F from Eq. 1, averaged arithmetically over the selected neurons within each layer. Reported model summaries take the median of these layer means. Geometric summaries and alternative reference regions are stated where used below.

Table 2: Selection settings and what remains fixed. Q98 and Q95 denote neuron-specific 98th- and 95th-percentile thresholds, respectively. Each row is expanded in the indicated appendix.

Analysis	Selection	Fixed across the comparison
Pretrained profiles (B)	Q98; first 32 tokens; 1% per layer	Identities and thresholds across positions
Ablation (C)	Qwen3-0.6B main set; 31 per layer	Selected neurons before held-out evaluation
Boundary token (D)	Main set for the primary comparison; reselection checked separately	Primary identities and thresholds
Document restart (D)	Independent Q98 calibration and discovery; 31 per layer	Identities and thresholds across the paired inputs
Attention gating (D)	Q98; first 32 tokens; 1% per layer in each checkpoint	Same text inputs; separate model-specific populations
Paired prefix (E)	Independent Q98 beginning selection; 1% per layer	Identities, thresholds, and target tokens
Corpus prefix (E)	Q98; 1% per layer at each condition's target start	Prefix length and target position in the two prefixed conditions
k-lag copy (F)	Q98; up to ten layer-0 neurons at each position	Reselected by position, condition, and seed
Novel-KV (G)	Q95 on natural text; 31 per layer	Identities and amplitude calibration across evidence conditions
Retrieval patch (H)	Top-5% events; eight layer-0 neurons ranked at the first pair	Selected coordinates across clean and patched evaluations

B Pretrained Models and Selection Robustness

This appendix supplements Sections 4.1–4.2 and Figure 2(a–d). It specifies the model-dependent selection budget and tests whether the corpus-level pattern is also visible in individual sequences.

Models and text. Each checkpoint is evaluated on 256 length-2,048 RefinedWeb sequences using its own tokenizer. Source files are shuffled before scanning documents; documents shorter than 2,048 tokens are skipped, and the first 2,048 tokens of each accepted document are retained. Tokenization adds no special tokens. Table 3 gives the layer counts and selection budgets. We use the released base checkpoints; model cards identify the specific releases.[1] All checkpoints use the pre-down-projection MLP coordinates and the calibration and ranking procedure in Appendix A.

Table 3: Models in Figure 2(a–d). The last column is the median across layers of the mean first-32-token relative activation rate.

Checkpoint	Layers	MLP width	Selected/layer	Median F
Qwen3-0.6B-Base	28	3,072	31	7.67
Qwen3-1.7B-Base	28	6,144	62	8.62
Qwen3-4B-Base	36	9,728	98	7.00
Qwen3-8B-Base	36	12,288	123	7.13
Llama-3.1-8B	32	14,336	144	4.74
Mistral-7B-v0.3	32	14,336	144	5.50

Individual-sequence evaluation. For each of the original 256 Qwen3-0.6B sequences, we measure the first-32-token rate of the fixed main-experiment neurons and divide by each neuron's original corpus-wide reference rate, retaining the original thresholds. We average within layers and take the median over 28 layers. Every sequence is above $1\times$, and every sequence exceeds all 20 layer- and count-matched random neuron sets drawn from nonsink neurons.

Leave-one-out selection. We remove the evaluated sequence from the counts used for ranking and from the reference rate, select neurons on the other 255 sequences, and evaluate the omitted sequence. The original calibration thresholds remain fixed. Table 4 summarizes both checks; intervals resample sequences 2,000 times. The sequence medians differ from the main-text $7.67\times$ because averaging sequences after taking layer medians differs from pooling corpus counts first.

Table 4: Single-sequence checks on Qwen3-0.6B. Intervals are 95% sequence-bootstrap intervals for the median.

Selection	Median [95% interval]	Sequences above $1\times$
Fixed main-experiment neurons	9.105 $[8.497, 9.706]$	256/256
Leave-one-out ranking	9.101 $[8.506, 9.725]$	256/256

Selection in other windows. We reselect neurons in each of a sequence's 64 nonoverlapping 32-token windows. The initial window exceeds 95% of the other 63 reselected-window controls in 227/256 sequences with corpus-calibrated thresholds, and 202/256 with thresholds estimated on the individual sequence. This descriptive comparison retains all 256 sequences.

C Activation Ablation

This appendix supplements Section 4.3 and Figure 2(e) by specifying the intervention, evaluation split, and target-position aggregation.

Intervention and controls. We fix the 868 Qwen3-0.6B coordinates selected in the main experiment (31 in each of 28 layers). At every input position, all selected pre-down-projection activations are set to zero in all 28 layers during the same forward pass. We apply the identical intervention to 20 random neuron sets, each containing the same number of neurons per layer drawn from outside the sink set.

[1] Official model cards: Qwen3-0.6B-Base, Qwen3-1.7B-Base, Qwen3-4B-Base, Qwen3-8B-Base, Llama-3.1-8B, and Mistral-7B-v0.3.

Evaluation data. We use 2,048 held-out RefinedWeb documents of 2,048 tokens, excluding localization documents by their source-file and line identifiers. Clean and ablated runs receive the same inputs. Forward computation uses BF16, with cross-entropy computed in FP32 and loss sums accumulated in FP64; evaluation batch size is eight.

Loss aggregation. Input position t predicts target token $t + 1$. Consequently, the first 32 predictions correspond to target positions $[1, 33)$, and the late region uses $[129, 2048)$. Within each region we pool the loss increase over its target tokens before averaging. Table 5 reports these aggregates and variability across the 20 random neuron sets.

Table 5: Token-weighted increase in NLL after ablation. Random values are the mean $\pm$ one standard deviation over 20 random neuron sets.

Target positions	Sink neurons	Random neurons
Early: $[1, 33)$	0.581	0.0777 ± 0.0775
Late: $[129, 2048)$	0.0274	0.0381 ± 0.0203

Every random neuron set has a smaller early ablation effect than the sink neurons. The late sink-neuron effect lies within the spread of the random controls.

D SEQUENCE-BEGINNING CONTROLS

This appendix supplements Section 5.1 and Figure 3. The three controls separately change the explicit boundary marker, the source location of the opening text, and the attention mechanism.

Boundary-token insertion with fixed neurons. For each original Qwen3-0.6B sequence x, the inputs are $x[0 : 2048]$ and $[151643] + x[0 : 2047]$. The inserted token is `<|endoftext|>` (ID 151643). The main-experiment neurons and Q98 thresholds are fixed. The first 32 text tokens occupy $[0, 32)$ without insertion and $[1, 33)$ with insertion. Their median relative activation rates are $7.67\times$ and $7.447\times$, with all 28 layer means above $1\times$ in both conditions.

Boundary-token effects on selection. We also rank neurons separately with and without the marker, aligning the same 2,047 text tokens and excluding the marker from the reference region. The two full-corpus sets share 699 of 868 coordinates (80.53%). To check whether either selection requires the marker to respond, we rank on the first 128 paired examples and evaluate both fixed sets on the other 128. The original no-boundary Q98 calibration is retained. Table 6 shows elevated responses in all four combinations.

Table 6: Boundary-token selection robustness. Entries are medians of layer-mean relative activation rates on the evaluation half; all four entries have 28/28 layers above $1\times$.

Input used for selection	Evaluate without marker	Evaluate with marker
Without marker	7.1758	7.0096
With marker	6.9888	7.2403

Document restart. This experiment selects 31 neurons per layer using 256 natural-text calibration sequences and a separate 256-sequence discovery set. This independently selected population is fixed before the paired evaluation on 256 long documents. One input takes the document's first 2,048 tokens. The other starts from a source-token offset sampled between 4,096 and 12,288, aligned to a nearby sentence or paragraph boundary, and takes the next 2,048 tokens. Its model positions restart at zero.

For each condition, we divide the first-32-token activation rate by the sequence-wide rate across that condition's inputs. We average the ratios over the 31 neurons within each layer and take the median over 28 layers. The main text and Figure 3(b) report $6.74\times$ and $4.24\times$ for these conditions. All 28 restart layer means exceed $1\times$.

Attention-gating comparison. We use the two released checkpoints of Qiu et al. (2025): standard attention (`1B_baseline`) and elementwise SDPA output gating (`1B_gate_elementwise`). Both receive the same 256 length-2,048 RefinedWeb inputs and have 28 layers with residual width 2,048; their MLP widths are 6,144 and 5,504. Using Q98 thresholds, we select the highest-ranked 1% of coordinates separately in each layer and model (62 and 56 coordinates per layer) by the first-32-token rate relative to the sequence-wide rate.

We average the log relative activation rates over selected neurons and layers, and exponentiate. This gives $7.19\times$ for the baseline and $5.12\times$ for the gated checkpoint. The schematic attention maps in Figure 3(c) illustrate the source study's reported attention patterns.

E Matched-Prefix Experiment

This appendix supplements Section 5.2 by separating the fixed-neuron, identical-target test from the corpus comparison in Figure 4. Only the paired test holds both neuron identities and target tokens fixed.

Paired construction. Both conditions use identical target tokens $B[1024 : 2048]$. The related condition prepends $B[0 : 1024]$, whereas the unrelated condition substitutes an equally long prefix $A[0 : 1024]$ from another document. Target token identities and positions are held fixed across conditions. We evaluate 131 paired examples for Qwen3-0.6B and 148 for Llama-3.1-8B.

Fixed neurons and target-start activation rates. An independent sequence-beginning analysis supplies Q98 thresholds and 31 neurons per layer in Qwen3-0.6B or 144 per layer in Llama-3.1-8B. We fix these throughout the paired evaluation. For each target, we compute each selected neuron's activation rate $p_c^{(\ell)}(r)$ over its first 32 tokens, at absolute positions $r = [1024, 1056)$, and average over selected neurons and targets. Table 7 reports this mean activation rate for each condition and the unrelated mean divided by the related mean. Intervals resample complete target pairs 20,000 times with seed 20260727.

Table 7: Paired prefix comparison with fixed neurons and identical target tokens. Condition columns are mean target-start activation rates (percent); the final column divides the unrelated rate by the related rate and gives its 95% paired-bootstrap interval. Ratios use unrounded rates.

Model	Pairs	Related (%)	Unrelated (%)	Ratio [95% interval]
Qwen3-0.6B	131	1.55456	2.35584	1.51544 [1.41522, 1.62572]
Llama-3.1-8B	148	1.89065	3.19066	1.68760 [1.60810, 1.77254]

The Qwen3-0.6B rate therefore becomes 1.51544 times its original value: a relative increase of 51.54%, or an absolute increase of 0.80128 percentage points. For Llama-3.1-8B, the corresponding increases are 68.76% and 1.30001 percentage points. The ratios compare prefix conditions, rather than comparing either condition with the sequence-wide baseline in F.

Target-start versus target-body activity. We divide each target's start rate by its rate over target offsets $[128, 512)$, then average these ratios across targets. This start/body ratio increases from 0.98 to 1.41 in Qwen3-0.6B and from 1.04 to 1.63 in Llama-3.1-8B. The paired increases are 0.43 (95% interval $[0.34, 0.53]$) and 0.59 ($[0.52, 0.67]$).

Corpus comparison in the main figure. The no-prefix condition measures the first 32 tokens of a true document beginning. The related-prefix condition measures a same-document continuation at position 1,024. The unrelated-prefix condition measures a new document beginning at position 1,024 after a different document's 1,024-token prefix. The two prefixed conditions share prefix length and target position, but use different target passages. Approximately 1% of neurons per layer are selected separately at each condition's target start. This asks whether the population pattern reappears; the paired analysis above establishes the same-neuron result.

Table 8: Corpus comparison with condition-specific selection. Values are geometric relative activation rates with 95% layer-bootstrap intervals.

Model	No prefix	Related prefix	Unrelated prefix
Qwen3-0.6B	7.51 [6.75, 8.38]	1.28 [1.27, 1.29]	5.25 [4.75, 5.78]
Llama-3.1-8B	3.93 [3.59, 4.31]	1.31 [1.29, 1.32]	3.84 [3.49, 4.24]
DCLM-7B	5.26 [4.80, 5.79]	1.38 [1.36, 1.40]	4.37 [4.09, 4.67]

Corpus aggregation and intervals. For Table 8, we average log regional/sequence-wide rate ratios over selected neurons within each layer, average across layers, and exponentiate. The 95% intervals use 2,000 layer-bootstrap draws with seed 20260601.

Boundary separators. With raw concatenation, the mean log ratio at the second document's start divided by the corresponding mean log ratio at the true sequence beginning is 0.82, 0.98, and 0.89 for Qwen3-0.6B, Llama-3.1-8B, and DCLM-7B. Same-position continuations give 0.12, 0.19, and 0.19. These ratios use mean log enrichment before exponentiation. The response also persists in the evaluated newline, EOS, and EOS+BOS separator variants.

F Controlled k-Lag Copy Task

This appendix supplements Section 5.3 and Figure 5 with the training setup, position convention, and calculation of the plotted sink span.

Data and position convention. Tokens are indexed from zero. The first k tokens, $x_0, \ldots, x_{k-1}$, are independent uniform draws from $V = 64$ symbols; every later token obeys $x_t = x_{t-k}$. Hidden position τ predicts $x_{\tau+1}$, so the first predictable target is x_k at $\tau = k - 1$. The conditional entropy is $\log V$ for $0 \le \tau \le k - 2$ and zero thereafter. For $k = 4$, three next-token targets are unpredictable before the first copied target. The IID control samples every token independently, leaving conditional entropy $\log V$ throughout.

Training and evaluation. For each $k \in \{2, 4, 8, 16, 32, 64\}$ and each seed in $\{0, 1, 2, 3\}$, we train a separate two-layer decoder-only Transformer with 512 prediction positions. The model has residual width 96, four attention heads, MLP width 192, pre-block RMSNorm, SwiGLU, RoPE with base 10,000, and no dropout. AdamW training uses 2,000 steps, learning rate 10^{-3}, weight decay 0.01, batch size 64, and gradient-norm clipping at 1.0. Sequences are sampled dynamically. Evaluation uses four batches of 32 newly sampled sequences per model. The IID control has its own four models and is not duplicated for different nominal lags.

Selection at each position. To estimate the sink span without prespecifying an initial selection window, we apply the same selection rule at every position. We use the layer-0 activation $m^{(0)}_{\tau,c}$ from Section 3. Each neuron's Q98 threshold is estimated over all sequence–position pairs in that model's evaluation set. We compute $F^{(0)}_c(\tau)$ using Eq. 1, with the baseline pooling all 512 positions, including τ. The selected set $\mathcal{S}_\tau$ contains the ten neurons with the largest $F^{(0)}_c(\tau) > 1$, or all such neurons if fewer than ten exist, breaking ties by neuron index. Ten is the 5% budget rounded up from 192 neurons. Neurons are reselected at each position, allowing the participating population to vary.

Relative activation-rate profile. For each seed, we average $F^{(0)}_c(\tau)$ arithmetically over $\mathcal{S}_\tau$ to obtain $\overline{F}(\tau)$. Every evaluated position has ten selected neurons. Lines and bands in Figure 5(b) give the mean and population standard deviation of these profiles across four seeds. The $1\times$ line is the sequence-wide baseline for an individual neuron; selecting the most elevated neurons separately at every position can keep the population mean above one even at late positions.

Consecutive span from relative activation rates. Following the main-text definition, let $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ be the mean and population standard deviation of $\overline{F}$ over $\mathcal{B} = \{256, \ldots, 511\}$. For each seed, the sink span is

$$w_{\text{sink}} = \max\{w \in \{0, \ldots, 512\} : \ \overline{F}(\tau) \geq \mu_{\mathcal{B}} + 2\sigma_{\mathcal{B}} \text{ for all } 0 \leq \tau < w\}. \tag{5}$$

Counting stops at the first sub-threshold position; later isolated spikes do not extend the span. The late reference accounts for the elevation induced by selecting neurons at every position. Calibration, selection, the late mean and standard deviation, and span computation are performed separately for every model, before averaging spans across seeds. After Q98 thresholding, activation magnitude does not enter the calculation. The late region and multiplier determine the operational width; this threshold is not a significance test.

Table 9: Consecutive initial sink spans in the copy task. Standard deviations use the four-seed population convention, as do the bands and error bars in Figure 5.

Condition	Seed 0	Seed 1	Seed 2	Seed 3	Mean $\pm$ SD
$k = 2$	1	1	6	1	2.25 ± 2.17
$k = 4$	9	3	3	9	6.00 ± 3.00
$k = 8$	5	11	8	7	7.75 ± 2.17
$k = 16$	12	9	15	7	10.75 ± 3.03
$k = 32$	19	25	20	28	23.00 ± 3.67
$k = 64$	37	39	35	44	38.75 ± 3.34
IID	2	2	3	1	2.00 ± 0.71

The span does not locate the exact predictability transition. The brief IID span describes its initial elevation relative to its own late region.

G Novel-KV Evidence Intervention

This appendix supplements Section 5.4 and Figure 6 by specifying how evidence is blocked and how the fixed population's activation magnitude is measured.

Prompts and evidence access. We evaluate Qwen3-0.6B-Base in BF16 on 2,000 paired base prompts. Each length-2,048 prompt contains 408 random single-token key–value mappings and a final query, with the queried mapping near 10%, 50%, or 90% of the table. The replacement condition changes only its key to an unused key. The masking condition keeps every token unchanged and prevents all later positions from attending to the entire queried-mapping span. Applying the mask throughout the causal computation blocks both direct access and relay through intervening tokens; masking only the final query would not provide this control. The first 2,047 positions are processed as the prefill and the final query is evaluated using its resulting cache. Answer accuracy is restricted to the 408 candidate values, giving chance $1/408 \approx 0.245\%$.

Fixed population and calibrated magnitude. Independent natural text supplies Q95 event thresholds ($q = 0.05$) for beginning-based selection, retaining 31 coordinates in each of 28 layers. The query readout uses their absolute activation magnitudes, rather than thresholding them again. A separate set of 256 length-2,048 natural-text sequences supplies, for each coordinate, the mean $\mu_c^{(\ell)}$ and population standard deviation $\sigma_c^{(\ell)}$ of $|m_{t,c}^{(\ell)}|$ over positions $[512, 2048)$. These calibration quantities and the 868 selected coordinates remain fixed across evidence conditions. At the final query position t, the readout is

$$\frac{1}{868} \sum_{\text{selected } (\ell,c)} \frac{|m_{t,c}^{(\ell)}| - \mu_c^{(\ell)}}{\max\{\sigma_c^{(\ell)}, 10^{-6}\}}. \tag{6}$$

Each coordinate's absolute activation is centered by its calibration mean and expressed in calibration-standard-deviation units before averaging.

Matched nonsink population. For each selected coordinate, a greedy match chooses an unused nonsink coordinate from the same layer with nearby log calibration mean and log calibration standard deviation of absolute activation. The candidate pool excludes both the event-selected sink set and an auxiliary beginning-selected set based on activation magnitude. This gives the same layer counts without replacement. We apply Eq. 6 using the matched coordinates' own calibration means and standard deviations.

Paired results. For each prompt we subtract its visible-condition readout from its unavailable-condition readout, then average over prompts. The 95% intervals use 5,000 paired-bootstrap resamples. Table 10 retains the numerical results underlying the main figure. Restricted accuracy is 85.95% with visible evidence, 0.25% after replacement, and 0.15% after masking.

Table 10: Unavailable-minus-visible changes in standardized activation magnitude. The last column subtracts the matched nonsink change from the sink change. Values are computed before rounding.

Intervention	Sink change [95% interval]	Nonsink change	Difference
Replacement	0.06646 [0.06513, 0.06779]	0.01309	0.05338
Masking	0.06656 [0.06521, 0.06794]	0.01309	0.05348

The sink readout increases on 98.05% of replaced prompts and 98.00% of masked prompts. Using this same scalar readout to distinguish unavailable from visible evidence gives AUROC 0.920–0.921.

H Retrieval Intervention

This appendix supplements Section 6.1 and Table 1. It specifies the controlled model, the paired first–middle construction, and the individual accuracies that produce the reported gap.

Model and task. We use the content–label task of Wu et al. (2025). The selected checkpoint is a two-layer causal Transformer with residual and MLP width 64, ReLU MLPs, RoPE, and no LayerNorm. It is trained for 100,000 updates with batch size 128 and training seed 0. Training uses 2,048 content classes, 32 possible labels, and content noise scale 0.75. Each context contains eight content–label pairs, with two sampled classes occurring four times each, followed by a query: 17 input positions in total. The intervention evaluation uses novel content classes and 2,000 examples per condition.

Paired first–middle construction. For each sample index, both conditions reset the random generators to the same seed, producing the same underlying class means, class-label assignments, context-class ordering, and noise draws. Let a and b be the original first and fifth content vectors, with labels y_1 and y_5. In `first_vs_middle`, the query is generated from the original class of a, then the fifth content vector is replaced by a. In `middle_vs_first`, the query is generated from the original class of b, then the first content vector is replaced by b. Both labels and the other six content–label pairs are preserved. Table 11 summarizes the construction.

Labels are sampled independently for distinct original content classes; repeated occurrences of the same class share its label. No constraint enforces $y_1 \neq y_5$. The query uses a separate noise draw from the context tokens. Its noise is paired across conditions, but its class mean can differ, so the conditions can have different inputs and predictions. Their accuracies are evaluated against y_1 and y_5, respectively.

Table 11: Paired retrieval construction from original pairs (a, y_1) and (b, y_5). The query is a noisy realization of the indicated original class; y_1 and y_5 may coincide.

Condition	First pair	Fifth pair	Query class	Target
`first_vs_middle`	(a, y_1)	(a, y_5)	Class of a	y_1
`middle_vs_first`	(b, y_1)	(b, y_5)	Class of b	y_5

Selecting the eight coordinates. Selection uses 4,096 training-distribution examples and layer-0 post-ReLU, pre-output-projection activations. The evaluator takes these examples from the training-data generator; exclusion from prior training is not assumed. For each neuron, it marks the top 5% of its activation observations as events and ranks neurons by their event frequency at positions 0 and 1, the first content–label pair. This finite-sample top-event rule and the eight-of-64 budget are the settings of this controlled model, rather than the pretrained-model Q98/1% defaults. The selected zero-based coordinates are $\{1, 3, 13, 17, 25, 45, 52, 61\}$, fixed for all evaluations. Five random controls each select eight nonsink coordinates from the same layer and undergo the same activation-copying operation.

Intervention and aggregation. Within each forward pass we simultaneously copy the selected activations from positions 0 and 1 to positions 8 and 9, respectively, immediately before the layer-0 MLP output projection. Other coordinates at this intervention site are left as computed; downstream activations can change. Clean and patched runs use the same inputs and weights. Accuracy follows the evaluation protocol's mean of batch accuracies in each condition, and the bias is the first-condition accuracy minus the middle-condition accuracy. Table 12 reports the selected top-eight setting and its five individual controls. These controls describe variability across coordinate sets at a single trained checkpoint, not uncertainty across training seeds.

Table 12: Accuracies and first–middle gaps for the selected retrieval intervention. Accuracy columns are evaluation averages in percent; gaps and changes are percentage points. Random rows use the five saved random neuron sets, and Random mean averages these sets.

Intervention	First	Middle	Gap	Gap change
Clean	84.28	66.87	17.41	0.00
Sink patch	83.82	68.41	15.41	−2.00
Random 0	83.54	66.64	16.90	−0.51
Random 1	85.25	65.38	19.87	+2.46
Random 2	85.49	64.96	20.53	+3.12
Random 3	85.18	65.66	19.51	+2.10
Random 4	85.12	66.14	18.97	+1.56
Random mean	84.92	65.76	19.16	+1.75

Paired accuracy changes. The sink patch decreases first-target accuracy by 0.46 points and increases middle-target accuracy by 1.54 points, jointly reducing the gap by 2.00 points (11.5% of the clean gap). One random control also decreases the gap, but its decrease is smaller; the five-control average increases it.