SHIFTING MECHANISMS: HOW POSITIONAL ENCODING CHOICE SHAPES IN-CONTEXT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models increasingly use architectures that vary attention span and positional encoding across layers, such as applying RoPE with sliding-window attention and NoPE with global attention. However, how these choices shape in-context retrieval remains unclear. To study this question, we take a mechanistic view, tracing how positional encoding (PE) choice shapes the internal mechanisms models use for in-context retrieval. Across 22 open-weight models spanning eight families, we find that standard RoPE models rely primarily on positional retrieval, while PE hybrids shift toward semantic retrieval. On a controlled pre-training ablation, we show that confining positional encoding to local layers is sufficient to produce this semantic shift, and find that it selectively degrades representations of positional information. Finally, we show that the reported long-context gains of PE hybrids mask a retrieval trade-off: while SWA NoPE improves over RoPE on multiple-target retrieval and QA, it degrades when distinguishing semantically similar keys. These behavioral differences track the mechanism shift from positional toward semantic mechanisms rather than a uniform improvement in long-context retrieval.

1 INTRODUCTION

Language model architectures have increasingly moved away from standard RoPE (Su et al., 2021) toward hybrid designs that vary attention span and positional encoding across layers, architectural choices we jointly refer to as positional encoding (PE) choice. Prior work has shown that PE hybrids can improve long-context retrieval performance: SWA NoPE (Puvvada et al., 2025; Yang et al., 2025b), which combines local sliding-window attention (SWA) layers using RoPE with global attention layers without PE (NoPE), outperforms RoPE on several long-context benchmarks. Yet why PE choice produces these differences in long-context retrieval ability remains unclear.

Recent interpretability work provides a way to investigate this question, identifying three core mechanisms that language models use for in-context retrieval: positional, based on an entity's location in context, and lexical and reflexive, based on an entity's semantic content (Gur-Arieh et al., 2026). Using this mechanistic analysis as an intermediate lens between model architecture and retrieval behavior, we ask whether PE choice changes how models retrieve information and whether these underlying mechanism differences explain long-context retrieval performance.

We find that PE hybrids have a consistently more semantic mechanism allocation. Across 22 open-weight models spanning eight families, RoPE models rely primarily on the positional mechanism, while PE hybrids rely primarily on semantic (lexical and reflexive) mechanisms. To study why this shift occurs, we analyze a controlled set of pre-training ablations of PE choice (from RoPE to SWA RoPE to SWA NoPE), finding that SWA alone does not produce the shift, while additionally removing RoPE from global layers does. We find confining positional encoding to local layers degrades representations of positional information, consistent with the observed semantic shift.

Finally, we show how our mechanistic analysis explains the impact of PE choice on long-context retrieval behavior. We find that SWA NoPE outperforms RoPE on multiple-target needle-in-a-haystack (NIAH) and QA tasks, where semantic mechanisms excel at finding relevant content, but degrades when trying to distinguish between semantically similar keys. These results demonstrate that retrieval strategies produce different strengths and failure modes at long context, suggesting that SWA NoPE's long-context gains better reflect a shift in retrieval strategy than a uniform improvement.

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.

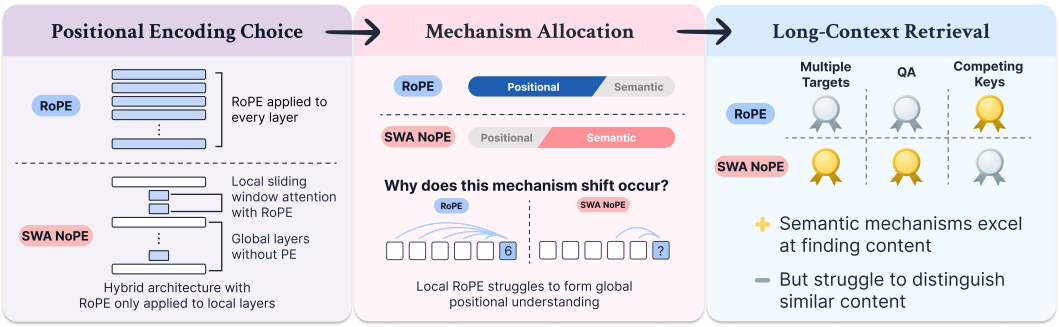

Figure 1: **Overview.** We show that positional encoding (PE) choice shapes mechanism allocation: the SWA NoPE architecture shifts toward semantic mechanisms compared to RoPE. We then show that this mechanism shift predicts long-context retrieval performance: SWA NoPE improves on multiple-target retrieval and QA tasks, but struggles to distinguish similar targets.

We summarize our contributions below:

- **Positional Encoding Choice → Mechanism Allocation.** We show that retrieval strategies are architecture-dependent and replacing uniform RoPE with SWA NoPE systematically shifts mechanism allocation from positional toward semantic retrieval.
- **Why does this shift occur?** Linear probing suggests that confining positional encoding to local layers degrades representations of positional information, providing a potential explanation for the shift away from positional retrieval.
- **Mechanism Allocation → Long-Context Retrieval.** We show that the mechanism shift predicts long-context retrieval behavior. The shift toward semantic mechanisms improves retrieval of relevant content but increases sensitivity to semantic confusability, which we validate experimentally across long-context retrieval tasks.

2 Related Work

Positional Encoding and Hybrid Architectures. RoPE (Su et al., 2021) is widely used in modern LLMs, but can struggle to extrapolate beyond its training window (Liu, 2026; Du et al., 2025; 2026), motivating substantial work on how positional encoding can scale to longer contexts. One line of work extends RoPE through interpolation and frequency scaling (bloc97, 2023a;b; emozilla, 2023; Peng et al., 2024), while another modifies or removes explicit positional encoding altogether (Gelberg et al., 2025; Barbero et al., 2024; Khan et al., 2026; Gopalakrishnan et al., 2025; Movahedi et al., 2026). Furthermore, transformers without explicit positional encodings can still achieve competitive length generalization in some settings (Haviv et al., 2022; Kazemnejad et al., 2023).

Most relevant to our work are the hybrid SWA NoPE architectures proposed by Yang et al. (2025b); Puvvada et al. (2025) that interleave local RoPE and global NoPE, improving aggregate long-context performance. Prior work has sought to explain these gains through properties of the architecture. Qiao et al. (2026), for example, argue that long-range retrieval is primarily carried by global attention layers, while local attention shapes how these retrieval capabilities emerge during training, motivating the use of NoPE in global layers. In contrast, we find that SWA NoPE does not uniformly improve long-context retrieval. Instead, its mechanism shift from positional toward semantic retrieval predicts distinct strengths and failure modes across retrieval settings.

Mechanistic Analysis of In-Context Retrieval. Early mechanistic work on entity binding identified a positional retrieval mechanism (Feng & Steinhardt, 2024; Prakash et al., 2024; 2025). More recently, Gur-Arieh et al. (2026) showed that models also use lexical and reflexive mechanisms, which we refer to collectively as semantic mechanisms. Complementary work characterizes individual attention heads as positional or symbolic, finding theoretically and empirically that symbolic mechanisms generalize more robustly to longer sequences (Urrutia et al., 2025; 2026). While this work characterizes the mechanisms models use for retrieval, we study how architectural choices in positional encoding shape which of these mechanisms models learn to rely on.

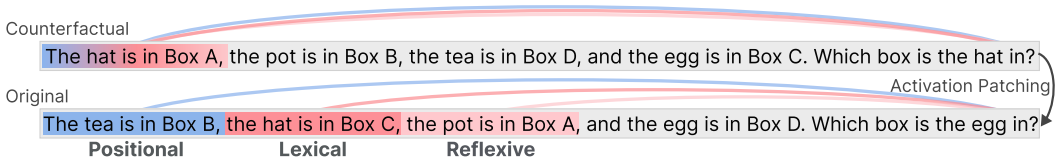

Figure 2: **Example of counterfactual activation patching.** The prompt pair is constructed so that each retrieval mechanism corresponds to a different answer after activation patching. The positional mechanism follows location and predicts 'B', the lexical mechanism follows the 'hat' key and predicts 'C', and the reflexive mechanism follows the self-referential pointer and predicts 'A'. If patching has no effect, the model retains the original prediction ('D').

3 Mechanistic Lens

In this section we explain the mechanistic lens we use throughout this work. Prior work (Gur-Arieh et al., 2026) found three core mechanisms that language models use to bind and retrieve entities in-context. We first describe the counterfactual approach used to disentangle these mechanisms, and then explain how we use it to measure mechanism allocation.

Counterfactual Patching. To understand how a model uses these mechanisms, Gur-Arieh et al. (2026) use carefully paired original and counterfactual prompts and apply activation patching on the residual stream to identify which mechanism dominates a given prediction. We illustrate their counterfactual patching framework using the Boxes task in Figure 2, where objects are assigned to boxes and the model is queried for which box contains a particular object.

The prompts are constructed so that the patched prediction uniquely identifies the dominant mechanism. We define each mechanism below:

- **Positional Mechanism (P).** The model stores the position of the queried item and later retrieves from that same position. In Figure 2, the counterfactual answer 'A' appears first, so the mechanism retrieves from the first position in the original prompt, returning 'B'.
- **Lexical Mechanism (L).** The model stores the queried item and later retrieves using that item. In Figure 2, the counterfactual answer stores the semantic key 'hat', so the mechanism retrieves the box associated with 'hat' in the original prompt, returning 'C'.
- **Reflexive Mechanism (R).** The model stores a self-referential pointer to the answer and later retrieves that answer directly. In Figure 2, the self-referential pointer directly stores 'A', so the mechanism returns 'A'.

In addition to these three mechanisms, patching can produce two other outcomes. We classify a sample as *no effect* when the model retains the original prompt's prediction ('D' in Figure 2), and as *unknown* when the prediction does not correspond to any of the three mechanisms or the original prediction. With a larger number of entities (we study 20 entities throughout this work), there are many additional predictions that can therefore fall into the unknown category.

Mechanism Allocation. A model may use different retrieval mechanisms across different samples. We therefore repeat the counterfactual patching procedure over many synthetic samples from a task and estimate the rate of each outcome as the fraction of samples classified as positional (P), lexical (L), reflexive (R), unknown, or no effect. We refer to these estimated rates as the model's mechanism allocation. We refer to the mechanism with the highest rate among P, L, and R as the model's dominant mechanism.

Given this mechanism allocation, our primary comparison is between positional and semantic retrieval. We group the lexical and reflexive mechanisms as semantic mechanisms, since both retrieve based on entity information, and compare them to the positional mechanism which retrieves based on location. We define the positional ratio as $r_{\text{pos}} = \frac{P}{L+R}$. Higher r_{pos} indicates a more positional mechanism allocation, while lower r_{pos} indicates a more semantic mechanism allocation. We use the mechanism allocation and positional ratio throughout this work to understand the internal mechanisms guiding in-context retrieval.

Table 1: **Model families and descriptions.** We compare RoPE architectures that apply RoPE consistently throughout the model with PE hybrid architectures that vary the application of RoPE across layers. We report the PE configuration, as well as window size and local:global ratio where applicable. † Gemma-4 2B/4B have a window size of 512 and Gemma-4 2B uses a 4:1 local:global ratio.

Model	PE Configuration	Window Size	Local:Global
RoPE Architectures			
Gemma-2 (Riviere et al., 2024)	RoPE (10K)	4096	1:1
Llama-3.1 (Grattafiori et al., 2024)	RoPE (500K)	—	—
Qwen-2.5 (Yang et al., 2024)	RoPE (1M)	—	—
Qwen-3 (Yang et al., 2025a)	RoPE (1M)	—	—
PE Hybrid Architectures			
Gemma-3 (Kamath et al., 2025)	RoPE (10K) / RoPE (1M)	1024	5:1
Gemma-4 (Gemma, 2026)†	RoPE (10K) / p-RoPE (1M)	512|1024	4:1|5:1
SmolLM3 (Bakouch et al., 2025)	RoPE (5M) / NoPE	—	3:1
Command R7B (Yang et al., 2025b)	RoPE (50K) / NoPE	4096	3:1

4 Positional Encoding Choice Shapes Mechanism Allocation

We use this mechanistic framework to understand how positional encoding choice shapes the retrieval mechanisms a model learns to use. We first show across 22 open-weight instruction-tuned models that PE hybrids exhibit a semantic shift compared to RoPE (Section 4.1). We then study a controlled progression of pre-training ablations on PE choice (from RoPE to SWA RoPE to SWA NoPE), finding that SWA alone does not produce the shift, while additionally removing RoPE from global layers does (Section 4.2). Finally, we find that restricting RoPE to local layers reduces the quality of positional information represented by the model, providing a potential explanation for the semantic shift (Section 4.3).

4.1 Model Battery

In this section, we analyze mechanism allocation across eight model families to examine how PE choice shapes the learned mechanisms for in-context retrieval. We divide these models into the following two groups based on their PE configuration (Table 1):

- ***RoPE Architectures.*** Models that use RoPE (Su et al., 2021) at every attention layer to encode positional information. Prior analyses of in-context retrieval mechanisms (Prakash et al., 2025; Gur-Arieh et al., 2026) studied only models of this type.
- ***PE Hybrid Architectures.*** We adopt a broad definition of PE hybrid in this work, considering any model that varies how positional encodings are applied across layers. These designs often combine this variation in PE with local and global attention layers. For example, Gemma-3 varies the RoPE base across local and global layers, Gemma-4 varies the RoPE base and uses p-RoPE (Barbero et al., 2024) in global layers, and Command R7B and SmolLM3 both use NoPE layers. Table 1 summarizes both the PE configuration and local/global attention structure of each architecture. Window size indicates the sliding window size (Beltagy et al., 2020) where applicable.

These PE hybrids reflect a broader architectural trend away from applying RoPE uniformly throughout the model and toward differentiating PE across local and global layers. Very recent architectures, including Llama 4 (Meta, 2025) and MAI-Thinking-1 (MAI, 2026), continue this trend. Our model battery therefore lets us ask what impact this shift in PE design has on retrieval mechanisms.

Experimental Details. We evaluate 22 instruction-tuned models across eight families: thirteen RoPE models (Gemma-2 {2B, 9B, 27B}, Llama-3.1 {8B, 70B}, Qwen-2.5 {3B, 7B, 32B, 72B}, Qwen-3 {1.7B, 4B, 8B, 14B}), and nine PE hybrid models (Gemma-3 {4B, 12B, 27B}, Gemma-4 {2B, 4B, 12B, 31B}, Command R7B, SmolLM3-3B). We evaluate each model on all 29 entity binding tasks from Gur-Arieh et al. (2026). For each model, we perform the mechanism analysis from Section 3 with 1000 samples per task at its selected retrieval layer (Table 6, Appendix B). We classify each prediction as positional, lexical, reflexive, or unknown, pool predictions across tasks, and compute $r_{\text{pos}} = \frac{P}{L+R}$ from the pooled counts. See Appendix A.2 for additional details.

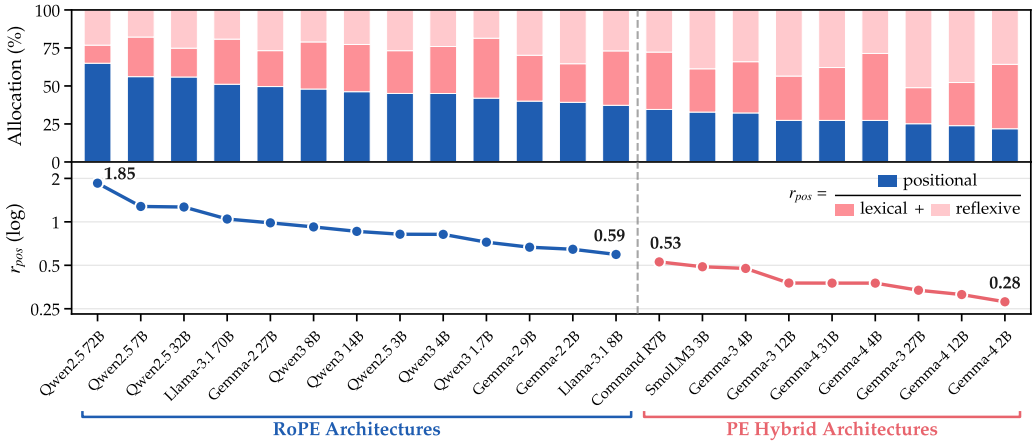

Figure 3: Mechanism allocation across the model battery pooled over 29 tasks and sorted by r_{pos}. (**Top**) Relative allocation to positional, lexical, and reflexive retrieval mechanisms. (**Bottom**) Positional ratio for each model. The architecture classes separate completely: every RoPE model has a higher r_{pos} (≥ 0.59) than every PE hybrid (≤ 0.53).

PE Hybrids Shift Mechanisms. We report r_{pos} and mechanism fractions normalized over P, L, and R in Figure 3, with the full outcome rates in Appendix C. We find that PE hybrids exhibit a systematic shift toward semantic retrieval. While every RoPE model has $r_{\text{pos}} \geq 0.59$ (Llama3.1 8B), every PE hybrid has $r_{\text{pos}} \leq 0.53$ (Command R7B). At the extremes, the positional mechanism accounts for 65% of predictions in Qwen2.5 72B ($r_{\text{pos}} = 1.85$), compared to only 22% in Gemma-4 2B ($r_{\text{pos}} = 0.28$), with semantic mechanisms correspondingly accounting for 35% and 78%. Despite substantial variation in model family, parameter count, context length, and training data, we find a clear semantic shift going from RoPE to PE hybrid architectures.

Prior mechanistic analyses of in-context retrieval studied RoPE models (Prakash et al., 2025; Gur-Arieh et al., 2026), and our extension to PE hybrids reveal a different regime. The positional mechanism accounts for the largest share among P, L, and R for every RoPE model we test, while for every PE hybrid, either the lexical or reflexive mechanism has the largest share (Table 7). In other words, the dominant mechanism consistently changes from positional to semantic across the two architecture classes. Although PE hybrids use the same retrieval mechanisms identified by prior work, they exhibit a qualitatively different mechanism allocation.

Our model battery reveals a clear difference in mechanism allocation between RoPE and PE hybrid models despite substantial variation in model family, scale, and architecture. However, this variation also makes it difficult to determine which architectural differences drive the shift toward semantic retrieval. We therefore turn to a controlled set of pre-training ablations to isolate the effect of PE choice.

4.2 Ablating Positional Encoding Choice

To isolate whether positional encoding choice drives the shift in mechanism allocation, we utilize the controlled pre-training checkpoints from Qiao et al. (2026):

- **RoPE.** Global attention for all layers, with RoPE (Su et al., 2021) applied in every layer.
- **SWA RoPE.** Interleaving global attention and local sliding-window attention (SWA) (Beltagy et al., 2020), where tokens only attend to a fixed window size. RoPE still applied in every layer.
- **SWA NoPE.** The same alternating attention pattern as SWA RoPE, but RoPE is applied only in local layers, while global layers use NoPE (Puvvada et al., 2025; Yang et al., 2025b).

Each architecture is first trained at 16K context length for 100B tokens, and then extended to 32K context length for an additional 5B tokens, totaling six checkpoints. These models each have 665M parameters and share the same training hyperparameters (see Qiao et al., 2026). Both SWA variants

Table 2: Mechanism allocation of three model architectures: RoPE, SWA RoPE, and SWA NoPE at two training checkpoints (16K and 32K training length) on the Boxes task. We report the percentage (%) of samples classified as positional, lexical, reflexive, and unknown. We additionally report the ratio of positional to semantic (lexical, reflexive) mechanisms. At both training lengths, introducing SWA RoPE alone does not produce the semantic shift, but SWA with NoPE global layers substantially shifts the mechanism allocation, reducing r_{pos} by a factor of two.

	Architecture	Positional	Lexical	Reflexive	Unknown	r_{pos}
16K	RoPE	17.6	56.6	1.6	24.1	0.30
	SWA RoPE	19.4	46.6	2.1	31.9	0.40
	SWA NoPE	10.0	68.6	1.4	20.0	**0.14**
32K	RoPE	15.9	60.6	1.4	22.1	0.26
	SWA RoPE	16.2	57.5	1.8	24.6	0.27
	SWA NoPE	9.5	71.9	1.3	17.3	**0.13**

use a 1:1 ratio of local to global layers with a sliding window size of 128 tokens. We utilize the framework from Section 3 to classify each sample as positional (P), lexical (L), reflexive (R), or unknown, based on which token the model predicts under activation patching. We measure at layer 15 using 10K samples per model. Finally, since these are small base models, we evaluate on the Boxes task from Figure 2 to ensure reliable task performance. For further details see Appendix A.3.

We report our results in Table 2. At both training lengths, SWA NoPE learns a substantially more semantic mechanism allocation than RoPE: r_{pos} falls from 0.30 to 0.14 for the 16K checkpoints and 0.26 to 0.13 for the 32K checkpoints. At both checkpoints, SWA NoPE reduces the positional mechanism and unknown rate, and increases the lexical mechanism.

In contrast, SWA RoPE does not exhibit this semantic shift and even increases r_{pos} at the 16K training checkpoint from 0.30 to 0.40, and nearly matches RoPE at 32K (0.26 to 0.27). The positional ratios are relatively low across all three architectures, consistent with the strong lexical bias of the Boxes task (see Figure 8 and Section C.2 for more details). These controlled experiments show that SWA alone does not produce the semantic shift, while additionally removing RoPE from global layers does.

4.3 Confining positional encoding degrades Ordering ID creation

Having established that the SWA NoPE architecture drives the shift toward semantic retrieval, we next investigate why this shift occurs. We first use our understanding of the three mechanisms guiding in-context retrieval to identify where PE choice could impact mechanism allocation. This analysis suggests that PE choice may affect how models form ordering IDs. We therefore examine whether SWA NoPE weakens the positional representations used to form ordering IDs.

Where can PE shape mechanisms? To begin, we look at the three in-context retrieval mechanisms. The lexical and reflexive mechanisms are both semantic by design, storing either a semantic key into the target entity or a self-referential pointer to the target entity. Neither mechanism explicitly relies on positional information for retrieval, making the positional mechanism the natural candidate through which PE choice could directly affect mechanism allocation. We therefore dig deeper into how the positional mechanism operates.

Recent work studying in-context retrieval found that the positional mechanism occurs in two distinct stages via a lookback mechanism (Prakash et al., 2025). During the first stage, the model associates each answer with an ordering ID (OI) that represents its ordinal position in the list. For example, the answer token appearing third in the list stores a representation of its ordinal position ("third").

During retrieval, the model identifies the position specified by the query and matches it to the corresponding OI to recover the answer. Once created, the OI acts as a positional key for retrieval, analogous to the semantic key used by the lexical mechanism. This decomposition suggests that PE choice may affect positional retrieval by altering how these positional keys are formed. We therefore investigate whether OI creation differs across the same controlled pre-training ablations used above.

SWA NoPE Confines PE. Our controlled ablation showed that SWA NoPE shifts toward semantic retrieval, while SWA RoPE does not. Since ordering IDs encode position, their formation may depend on how positional information is provided by these architectures. SWA RoPE applies RoPE in both local and global layers, while SWA NoPE confines RoPE to local layers and uses NoPE in global layers. Next, we test whether confining PE to local layers makes OI creation more difficult.

Experimental Details. To test whether ordering ID creation is impacted by PE choice, we train linear probes on the residual stream of the model. We use the same Boxes task from Section 4, extracting the residual stream at each of the 20 answer tokens, and train a linear probe to predict the ordering ID (the entity's position in the list $1-20$). As a control, we train a linear probe to predict the semantic key (the object paired with the box), which checks that semantic information is still represented. We train probes at layer 15 since we found in Section 4.2 that this is where retrieval occurs, and extract the 1280-dimensional activations at this layer.

Table 3: Linear probe accuracy (%) on the residual stream at layer 15 across items in the list, predicting either the index in the list or the attached semantic key, across the six model checkpoints. Standard deviation reported in parentheses over eight runs. SWA NoPE has the worst ordering ID probe accuracy, while semantic key probe accuracy remains high.

Context	Model	Ordering ID	Semantic Key
16K	RoPE	94.8 (0.2)	99.0 (0.2)
	SWA RoPE	91.6 (0.1) −3.2	98.8 (0.3) −0.2
	SWA NoPE	82.7 (0.3) −12.1	99.4 (0.1) +0.4
32K	RoPE	94.9 (0.2)	99.1 (0.2)
	SWA RoPE	92.7 (0.2) −2.2	99.0 (0.3) −0.1
	SWA NoPE	83.9 (0.3) −11.0	99.5 (0.1) +0.4

We fit an L_2-regularized logistic regression for each linear probe. We train on 1000 prompts and test on a disjoint set of 1000 prompts, forming training and test sets of $20 \times 1000 = 20{,}000$ samples each. We report the mean and standard deviation across eight independently sampled runs. We analyze the six checkpoints: RoPE, SWA RoPE, SWA NoPE, each trained at 16K context length and then extended to 32K. Additional details about the experimental setup can be found in Appendix A.4.

SWA NoPE degrades OI creation. Our results are shown in Table 3. Ordering IDs are substantially less linearly decodable under SWA NoPE, with probe accuracy decreasing by 12.1 points at 16K and 11.0 points at 32K compared to RoPE. SWA RoPE exhibits a much smaller decrease of 3.2 and 2.2 points, respectively. In contrast, semantic key accuracy remains nearly unchanged across all six checkpoints. As a result, the gap between semantic key and ordering ID accuracy grows from roughly 4 points under RoPE to 16 points under SWA NoPE. Overall, SWA NoPE selectively degrades the representation of OIs relative to semantic keys, consistent with its semantic shift.

Mechanistic Explanation. We hypothesize that confining PE to local layers impairs OI formation during training, shifting the model toward semantic retrieval. Under SWA NoPE, layers with explicit positional information have only a local view of the sequence, while layers with global context must infer position without explicit positional encoding, making globally consistent OIs more difficult to learn. Weaker OI formation may make positional retrieval less reliable, leading the model to rely more on semantic mechanisms whose underlying semantic representations remain intact.

This explanation predicts that as the sliding window shrinks relative to context length, OI formation should become more difficult and the mechanism shift should strengthen. Conversely, as the window approaches the full context length, the receptive field of the RoPE layers approaches the full context and this constraint should weaken. We leave testing this prediction with controlled checkpoints ablating window size for future work.

5 Mechanism Allocation Shapes Long-Context Retrieval

Having established in Section 4 that PE choice shifts the mechanisms models learn to use for retrieval, we next ask whether these differences predict long-context retrieval behavior. Prior work reports that SWA NoPE improves performance over RoPE (Yang et al., 2025b; Puvvada et al., 2025; Qiao et al., 2026) on long-context benchmarks such as RULER (Hsieh et al., 2024). Given that SWA NoPE also shifts retrieval toward semantic mechanisms, these gains raise a natural question: do semantic retrieval mechanisms simply scale better to long contexts? We investigate this question by examining how performance varies across the RULER subtasks using the same pre-training checkpoints studied in Section 4.

Table 4: RULER scores across the three model architectures: RoPE, SWA RoPE, and SWA NoPE. The 16K checkpoints are trained for 100B tokens, and the 32K checkpoints extend them for a further 5B tokens, and each is evaluated at their training context length. SWA NoPE substantially improves the overall RULER score, with gains concentrated in most NIAH and QA tasks. However, it does not improve over RoPE on the first NIAH multi-key task or consistently improve on variable tracking and aggregation.

Context	Model	Overall (13)	NIAH mk1 (1)	VT + Aggr. (3)	NIAH rest (7)	QA (2)
16K	RoPE	47.1	**91.2**	16.9	56.1	38.7
	SWA RoPE	47.2	84.8	**18.5**	57.2	36.7
	SWA NoPE	**58.3**	78.0	17.5	**77.2**	**43.4**
32K	RoPE	43.3	**92.4**	18.1	50.8	30.7
	SWA RoPE	45.7	81.6	**23.0**	55.1	28.8
	SWA NoPE	**53.9**	82.2	15.6	**69.6**	**42.0**

Table 5: We test each checkpoint on our confusable key task and compare to the RULER multi-value and multi-query subtasks. SWA NoPE struggles to select amongst competing keys, especially as they become more semantically similar, flipping the model ranking compared to multi-item NIAH where all keys are returned.

		Confusable keys				Multi-item	
Context	Model	$0/4$	$1/4$	$2/4$	$3/4$	MULTIVALUE	MULTIQUERY
16K	RoPE	**97.0**	**90.8**	**84.6**	**70.8**	35.0	35.0
	SWA RoPE	92.6	80.4	75.4	60.2	44.0	38.4
	SWA NoPE	88.2	73.0	57.4	52.6	**80.8**	**85.7**
32K	RoPE	**96.8**	**91.8**	**82.6**	**67.6**	25.2	30.5
	SWA RoPE	82.6	75.6	57.4	46.8	26.4	37.9
	SWA NoPE	76.0	67.6	46.8	38.6	**53.8**	**65.7**

We re-evaluate the released checkpoints from Qiao et al. (2026) on RULER (Hsieh et al., 2024). We evaluate both the 16K and 32K checkpoints at their respective training context lengths, using 500 samples per task. We evaluate on all thirteen tasks: eight needle-in-a-haystack (NIAH) tasks (three single key, three multi key, one multi value, and one multi query task), two QA tasks, a variable tracking task, and two aggregation tasks (common word extraction, frequent word extraction). We group the subtasks by retrieval setting to examine which tasks drive the aggregate RULER improvement. Additional experimental details can be found in Appendix A.5.

SWA NoPE is not uniformly better. We report the RULER results in Table 4. Consistent with prior work (Qiao et al., 2026), SWA NoPE substantially improves the overall RULER score over RoPE and SWA RoPE. However, this aggregate gain is not uniform across subtasks. SWA NoPE substantially improves most needle-in-a-haystack (NIAH) and QA tasks, but does not improve over RoPE on the first NIAH multi-key task or the variable tracking and aggregation tasks. We report the full task breakdown in Appendix D. These differences suggest that SWA NoPE changes which retrieval settings the model handles well rather than uniformly improving long-context retrieval.

Mechanism Shift Predicts a Retrieval Trade-off. The semantic shift identified in Section 4 suggests that SWA NoPE may excel when retrieving relevant content without discrimination among competing keys, but struggle when such discrimination is required. This distinction is reflected in the RULER results: SWA NoPE performs particularly well on multi-value and multi-query NIAH, where the requested values should all be returned rather than selecting one target while rejecting competing keys. To test the other side of this trade-off directly, we construct a retrieval task in which each context contains four needles, one queried key and three distractor keys, with each key consisting of four hyphenated words (e.g., *daughter-others-prop-sudo*). Importantly, even when the keys share no words ($0/4$), retrieval requires distinguishing the queried key from the three distractors. We increase semantic confusability by varying the number of words shared across keys from zero to three.

We report the results in Table 5. Requiring the model to distinguish among competing keys reverses the model ordering even when the keys share no words ($0/4$): RoPE outperforms SWA NoPE, whereas SWA NoPE substantially outperforms RoPE on multi-value and multi-query retrieval. Increasing the semantic similarity among these competing keys further amplifies this difference, as predicted by the shift toward semantic retrieval. At 16K, the gap between RoPE and SWA NoPE grows from 8.8 points at $0/4$ to 27.2 points at $2/4$, with the same pattern at 32K, before narrowing at $3/4$ as performance degrades across all architectures. Together, these results show that SWA NoPE's long-context gains reflect a shift in retrieval strategy with distinct strengths and failure modes rather than a uniform improvement in retrieval ability.

6 Conclusion

In this work we analyzed how positional encoding (PE) choice shapes the in-context retrieval mechanisms a language model learns during training. Across controlled pre-training checkpoints and a battery of open-weight instruction-tuned models, we find that PE choice substantially shifts whether models rely on positional or semantic retrieval. Our controlled analysis further links this shift to degraded ordering ID representations when positional encoding is confined to local layers. These differences in mechanism allocation translate into different long-context retrieval profiles, rather than a uniform improvement or degradation in retrieval ability. As recent open-weight models increasingly adopt architectures that vary PE across layers, understanding these mechanistic consequences becomes increasingly important.

Intentional Design. Through our mechanistic analysis we showed where and why SWA NoPE improves or degrades long-context retrieval, highlighting a broader role for interpretability in architectural design (Orgad et al., 2026). By allowing model developers to better understand how architectural choices shape the internal mechanisms learned by the model, mechanistic analysis can help developers shape for their desired downstream behavior. More broadly, our results suggest that mechanistic interpretability can serve not only to explain existing models, but also as a tool for intentional design.

Avoiding Long-Range RoPE. Recent work has highlighted theoretical limitations of RoPE at long context, leading to either token or positional confusion (Liu, 2026; Du et al., 2026). Long-context extensions based on RoPE scaling, such as YaRN (Peng et al., 2024), additionally modify the positional encoding to extrapolate beyond the context lengths seen during training. Instead, the SWA NoPE architecture restricts RoPE to fixed-size sliding-window layers, and so the maximum positional distance remains fixed as the global context length increases. At the same time, our results show that confining positional encoding in this way substantially changes the model's learned retrieval mechanisms. We leave further work exploring the benefits and drawbacks of these architectural choices to future work.

Limitations and Future Work

Scaling. One limitation of this work is the cost associated with training models from scratch. While we are able to test small-scale checkpoints ablating the positional encoding choice (Qiao et al., 2026), we are unable to train larger models, exhaustively ablate combinations of design choices, or analyze how the mechanisms develop during training. We leave these directions for future work.

Data. Mechanism allocation depends on which retrieval mechanisms are reinforced by the training objective, and therefore on the training data distribution. It would be interesting to study how different data distributions shape the in-context retrieval mechanisms learned by the model, or whether particular mechanisms could be targeted directly through the choice of long-context training data.

Linear hybrids. Many recent open-weight models have also adopted linear hybrids instead of PE hybrids (Kimi et al., 2025; Blakeman et al., 2025; Qwen, 2026). We provide a preliminary analysis of these models in Appendix E, but leave a more comprehensive study for future work. In particular, it would be interesting to understand whether alternative efficient attention strategies shift the allocation of the same retrieval mechanisms studied here, or lead models to learn fundamentally different retrieval mechanisms.

AI STATEMENT

In this work, we used generative AI tools for implementing methods in code and feedback on research methodology. Generating data sets, developing theoretical models, formulating/proving mathematical claims are not applicable to this work. Additionally, we used generative AI tools for help with plot and figure creation, as well as polishing the writing of the paper. We have reviewed all AI-assisted work, and stand by the research contributions made in this work. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*, 2024.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nvidia nemotron 3: Efficient and open intelligence. *arXiv preprint arXiv:2512.20856*, 2025.

bloc97. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023a. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

bloc97. Add NTK-Aware interpolation "by parts" correction, 2023b. URL https://github.com/jquesnelle/scaled-rope/pull/1.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. Context length alone hurts llm performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*, 2025.

Yufeng Du, Phillip Harris, Minyang Tian, Eliu A Huerta, Srikanth Ronanki, Subendhu Rongali, Aram Galstyan, and Hao Peng. Rope distinguishes neither positions nor tokens in long contexts, provably. *arXiv preprint arXiv:2605.15514*, 2026.

emozilla. Dynamically Scaled RoPE further increases performance of long context LLaMA with zero fine-tuning, 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *International Conference on Learning Representations*, volume 2024, pp. 36391–36413, 2024.

Yoav Gelberg, Koshi Eguchi, Takuya Akiba, and Edoardo Cetin. Extending the context of pretrained llms by dropping their positional embeddings. *arXiv preprint arXiv:2512.12167*, 2025.

Gemma. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Anand Gopalakrishnan, Robert Csordás, Jürgen Schmidhuber, and Michael C Mozer. Decoupling the" what" and" where" with polar coordinate positional embeddings. *arXiv preprint arXiv:2509.10534*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. In *International Conference on Learning Representations*, volume 2026, pp. 27757–27786, 2026.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1382–1390, 2022.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 4, 2025.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

Mohammad Aflah Khan, Krishna P Gummadi, Manish Gupta, and Abhilasha Ravichander. Fractional rotation, full potential? investigating performance and convergence of partial rope. *arXiv preprint arXiv:2603.11611*, 2026.

Kimi, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, et al. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025.

Feilong Liu. Rotary positional embeddings as phase modulation: Theoretical bounds on the rope base for long-context transformers. *arXiv preprint arXiv:2602.10959*, 2026.

MAI. Mai-thinking-1: Building a hill-climbing machine. Technical report, Microsoft AI, 2026. URL https://microsoft.ai/pdf/mai-thinking-1.pdf.

Meta. The llama 4 herd, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Sajad Movahedi, Timur Carstensen, Arshia Afzal, Frank Hutter, Antonio Orvieto, and Volkan Cevher. Selective rotary position embedding. In *International Conference on Learning Representations*, volume 2026, pp. 10219–10247, 2026.

Hadas Orgad, Fazl Barez, Tal Haklay, Isabelle Lee, Marius Mosbach, Anja Reusch, Naomi Saphra, Byron Wallace, Sarah Wiegreffe, Eric Wong, et al. Interpretability can be actionable. *arXiv preprint arXiv:2605.11161*, 2026.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 31932–31951, 2024.

Nikhil Prakash, Tamar Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*, volume 2024, pp. 8057–8082, 2024.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs. *arXiv preprint arXiv:2505.14685*, 2025.

Krishna C Puvvada, Faisal Ladhak, Santiago Akle Serrano, Cheng-Ping Hsieh, Shantanu Acharya, Somshubra Majumdar, Fei Jia, Samuel Kriman, Simeng Sun, Dima Rekesh, et al. Swan-gpt: An efficient and scalable approach for long-context language modeling. *arXiv preprint arXiv:2504.08719*, 2025.

Ziqing Qiao, Yinuo Xu, Chaojun Xiao, Zhou Su, Zihan Zhou, Yingfa Chen, Xiaoyue Xu, Xu Han, and Zhiyuan Liu. Rethinking the role of efficient attention in hybrid architectures. *arXiv preprint arXiv:2606.15378*, 2026.

Qwen. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864, 2021. URL https://arxiv.org/abs/2104.09864.

Felipe Urrutia, Jorge Salas, Alexander Kozachinskiy, Cristian Buc Calderon, Hector Pasten, and Cristobal Rojas. Decoupling positional and symbolic attention behavior in transformers. *arXiv preprint arXiv:2511.11579*, 2025.

Felipe Urrutia, Juan José Alegría, Cinthia Sanchez Macias, Jorge Salas, Cristian B Calderon, and Cristobal Rojas. Positional versus symbolic attention heads: Learning dynamics, rope geometry, and length generalization. *arXiv preprint arXiv:2605.31558*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Bowen Yang, Bharat Venkitesh, Dwaraknath Gnaneshwar Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. Rope to nope and back again: A new hybrid attention strategy. *Advances in Neural Information Processing Systems*, 38:64133–64157, 2025b.

A Experimental Details

In this section, we describe the experimental details for our main experiments. We first describe the mechanistic lens used throughout this work (Section A.1), then the model battery (Section A.2), the pre-training ablation (Section A.3), linear probing (Section A.4), and long-context behavior (Section A.5). All experiments are run in bf16 on a single NVIDIA RTX 6000 Ada (48GB), except models larger than 25B parameters, which exceed its memory and are run on an NVIDIA B200.

A.1 Mechanistic Lens

In this section we describe general details about the framework we use for our main mechanistic analysis. We utilize the datasets and framework from Gur-Arieh et al. (2026) that disentangles three mechanisms that language models use to perform in-context retrieval: positional, lexical, and reflexive. In total they propose 29 datasets: ten synthetic groupings (Filling Liquids, People and Objects, Programming Dictionary, Music, Biology Experiment, Chemistry Experiment, Transportation, Sports Events, Space Observations, Boxes). While the Boxes task only has two entities (Object, Box), each other task queries up to three different entities (first, second, third), totaling 29 task/query pairs.

The given LM is run on two prompts: the counterfactual prompt and original prompt. Then, at the given layer ℓ, the activations from the LM run on the counterfactual prompt are patched to the LM run on the original prompt. Through this patching procedure, each of the three mechanisms predicts a different token in the original prompt (see Gur-Arieh et al. (2026) for additional details on how the mechanisms are disentangled). Across 1000 generated pairs, the mechanism allocation can be measured by counting the number of samples that follow each of the mechanisms.

We prompt at multiple tokens for all our results, and patch at a different layer for each model. The full breakdown for our battery is shown in Table 6 highlighting our choices for each model.

	RoPE			PE Hybrid	
Model	**Layer**	**Positions**	**Model**	**Layer**	**Positions**
Gemma-2 2B	18	$-1, -4, -6, -8$	Gemma-3 4B	23	$-1, -4, -6, -8$
Gemma-2 9B	28	$-1, -4, -6, -8$	Gemma-3 12B	29	$-1, -4, -6, -8$
Gemma-2 27B	25	$-1, -4, -6, -8$	Gemma-3 27B	41	$-1, -4, -6, -8$
Qwen2.5 3B	31	$-1, -4, -6, -8$	Gemma-4 2B	24	$-1, -4, -6, -8$
Qwen2.5 7B	22	$-1, -4, -6, -8$	Gemma-4 4B	29	$-1, -4, -6, -8$
Qwen2.5 32B	51	$-1, -4, -6, -8$	Gemma-4 12B	35	$-1, -8, -10, -12$
Qwen2.5 72B	62	$-1, -4, -6, -8$	Gemma-4 31B	41	$-1, -8, -10, -12$
Llama-3.1 8B	16	$-1, -4, -6, -8$	Command R7B	19	$-1, -3, -5, -7$
Llama-3.1 70B	35	$-1, -4, -6, -8$	SmolLM3 3B	23	$-1, -4, -6, -8$
Qwen3 1.7B	19	$-1, -8, -10, -12$			
Qwen3 4B	24	$-1, -8, -10, -12$			
Qwen3 8B	24	$-1, -8, -10, -12$			
Qwen3 14B	28	$-1, -8, -10, -12$			

Table 6: Patched layer and token positions per model. Positions are matched by token role: the '?' and ':' of the query, plus the final two chat-scaffold positions. Offsets differ with template length.

A.2 Model Battery

In this section we describe details about the larger model battery. We evaluate 22 different post-trained models. RoPE: Qwen2.5 (3B, 7B, 32B, 72B), Qwen3 (1.7B, 4B, 8B, 14B), Llama-3.1 (8B, 70B), and Gemma-2 (2B, 9B, 27B). PE hybrid: Gemma-3 (4B, 12B, 27B), Gemma-4 (2B, 4B, 12B, 31B), Command R7B, and SmolLM3 3B. We also average across the full task battery using 20 entities, and evaluate with 1000 samples for each task. Two important decisions for each model are the choice of positions to patch, as well as the layer selection. Since these are all instruction-tuned models with their own chat templates, we match the patched tokens by role (final token, newline, ':', '?') and perform layer selection in Section B to select these for each model. Our results are shown in Table 6. Gemma-2, Qwen2.5, and Llama3.1 follow prior work, while we compute the selections for the new models ourselves.

A.3 PRE-TRAINING ABLATION

In this section we describe details about the model checkpoints from the pre-training ablation (Qiao et al., 2026), and our mechanistic analysis on these models.

We evaluate these models on Boxes because this task is relatively simple and they are 665M parameter base models that struggle on most tasks. Additionally, we use a cloze format (appending 'The object is in Box' to the end of the prompt) to ensure these models can retain high accuracy on this task. We run 10K samples per model to get a robust measure of the mechanism allocation, use only the last token position for patching, and patch at layer 15. The 16K checkpoints were trained for 100B tokens at 16K context length, and the 32K checkpoints were trained for an additional 5B tokens at 32K context length. Qiao et al. (2026) trains three different architectures: RoPE, SWA RoPE, and SWA NoPE. Both SWA architectures use a sliding window size of 128 tokens, with a ratio of 1:1 local:global layers. Further, SWA NoPE removes RoPE from the global layers but retains RoPE in the sliding window layers. These models also use an attention sink.

To validate our choice of layer 15, we present layer scans for these models. Our results are shown in Figure 4. We show that layer 15 is the last layer before the answer is retrieved for each model in our pre-training checkpoints. The messiness 4 setting differentiates the reflexive self-referential pointer from the counterfactual prompt's answer by ensuring the object does not exist in the original prompt, so it can only be reflexive in messiness 4 if the answer has already been retrieved. See Gur-Arieh et al. (2026) for additional details on the layer-selection procedure.

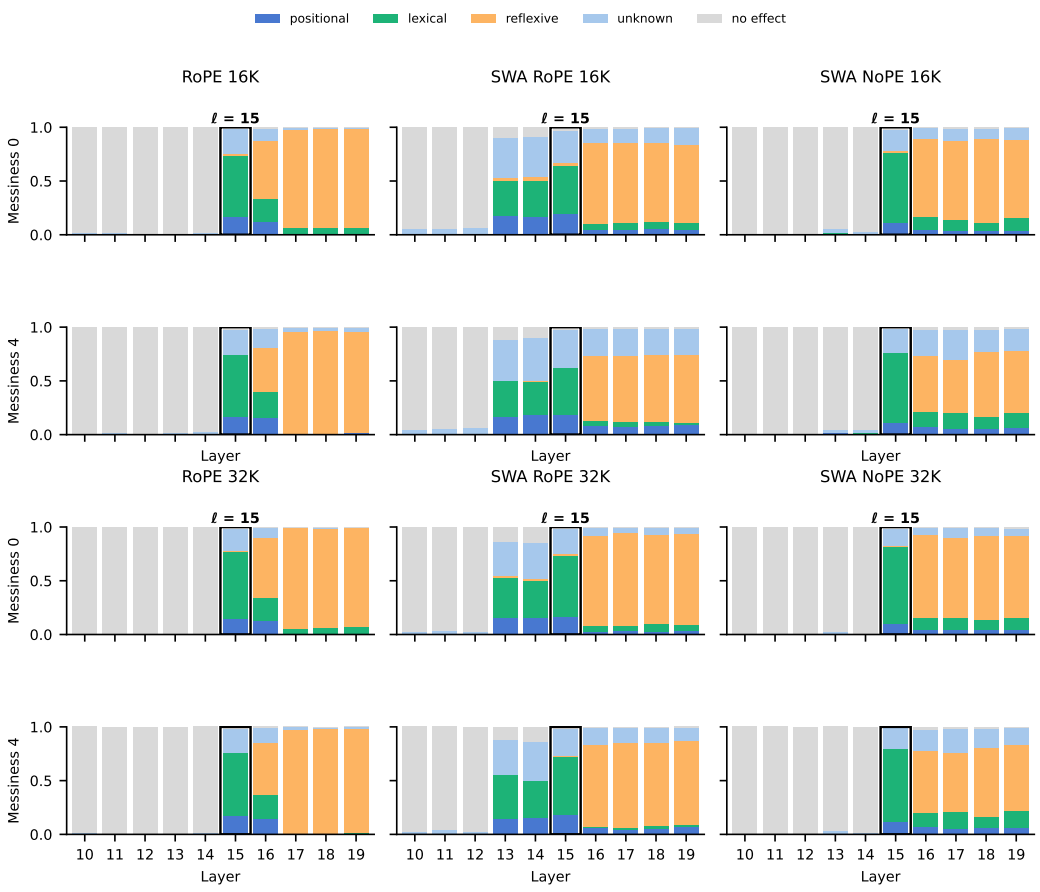

Figure 4: For every checkpoint in our pre-training ablation, layer 15 is the last layer before the answer is retrieved.

A.4 LINEAR PROBING

In this section we provide additional details for the linear probing experiments.

We probe the three pre-training checkpoints (RoPE, SWA RoPE, SWA NoPE), each with 30 layers and $d_{\text{model}} = 1280$. We evaluate both the 16K checkpoints trained for 100B tokens and the corresponding 32K checkpoints extended for an additional 5B tokens. For all models, we probe at layer 15, the same layer used for our mechanism analysis. We extract the residual stream entering block 15 using a forward pre-hook, with no KV cache.

We use the Boxes task with $N = 20$ entities ("the {Object} is in Box {Box}"). For each entity, we extract the residual at the Box-letter token. For the *ordering ID* probe, the label is the entity's position in the list (20 classes, chance 5%). For the *Semantic Key* control, we predict the object bound to each box (82 classes, chance $\approx 1.2\%$). Objects are randomized across positions between prompts, making the semantic-key label independent of position.

We fit multinomial logistic regression with L2 regularization ($C = 1$) on standardized features using scikit-learn (`StandardScaler` followed by `LogisticRegression`). We use all 1280 dimensions without dimensionality reduction. Each repeat uses 1000 training prompts (20,000 examples) and a separate set of 1000 held-out test prompts. We run eight repeats, redrawing both sets and refitting the probe each time, and report mean held-out accuracy $\pm$ SD across repeats.

A.5 LONG-CONTEXT BEHAVIOR

In this section we provide additional details for our RULER evaluation and confusable-key task. We evaluate our pre-training checkpoints on RULER following Hsieh et al. (2024), using 500 samples per task. We evaluate the 16K checkpoints at 16K context length and the 32K checkpoints at 32K context length.

Our confusable-key task builds on NIAH-mk1, where we replace each key with four hyphenated words and vary the number of words shared across competing keys. Each prompt contains four key-value pairs inserted into a haystack of Paul Graham essays.

For a $k/4$ example, we sample k words shared across all four keys and independently sample the remaining $4 - k$ words for each key, with no overlap between the remaining words. We then independently shuffle the four words within each key before joining them with hyphens. Thus, $0/4$ contains four disjoint keys, while at $3/4$ all four keys share three words and differ by only one. Values are uniformly sampled 7-digit numbers, and key-value pairs use the standard RULER needle template. We uniformly sample one of the four keys to query, giving a chance level of 25%.

We score a prediction as correct when the target value appears in the model generation. We generate 500 samples at each similarity level ($0/4$, $1/4$, $2/4$, $3/4$) for each checkpoint at both 16K and 32K context lengths. The $0/4$ condition uses the same haystack, needle and query structure as NIAH-mk1, but replaces its single-word keys with disjoint four-word keys. It therefore controls for the change in key format while varying only key similarity across the $0/4$–$3/4$ ladder.

B LAYER SELECTION VALIDATION

In this section we describe the protocol we use to select the patching layer for our main results. We mostly follow the setup of Gur-Arieh et al. (2026) and select the last layer at which patching does not transfer an already-retrieved answer. To test this, we use their messiness-4 counterfactual, in which the counterfactual answer is a new item that does not appear in the original prompt. If the model outputs this answer after patching, the answer can only have come from the patched activation, so retrieval has already happened by that layer. Scans patch the same positions as the main experiments and use the boxes task with query category 1. We show layer scans for three models to illustrate the process.

- Gemma-3 4B (Figure 5) has a clear shift from retrieval in layer 23 to almost entirely reflexive in layer 24 and the messiness-4 condition returns entirely the counterfactual answer, so we select $\ell = 23$.
- Gemma-4 4B (Figure 6) is also clean: after layer 29 the messiness-4 scan shows a substantial share of samples returning the counterfactual answer, so we select $\ell = 29$.
- Gemma-4 2B (Figure 7) is less clean: after layer 24 there is a small increase in the counterfactual answer. We therefore select $\ell = 24$, erring on the side of preventing an already-retrieved counterfactual answer from entering the mechanism allocation analysis.

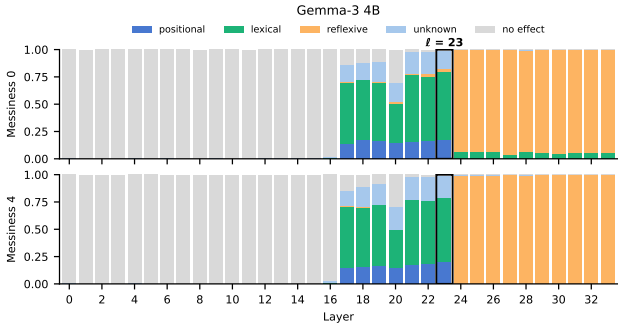

Figure 5: Layer scan for Gemma-3 4B with the messiness-0 (top) and messiness-4 (bottom) counterfactuals. Bars show the share of each mechanism when patching at each layer.

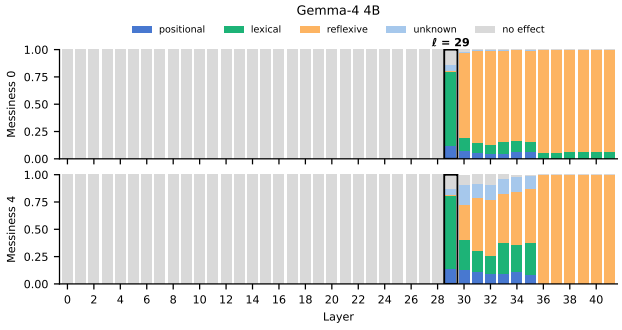

Figure 6: Layer scan for Gemma-4 4B with the messiness-0 (top) and messiness-4 (bottom) counterfactuals. Bars show the share of each mechanism when patching at each layer.

Our selected retrieval layer for all PE hybrids also ended up being the global layer when there was a mix of local and global layers. Gemma-4 2B, which has a 4:1 ratio of local to global layers (global layers 4, 9, 14, ...), illustrates this: patching first has an effect at global layer 19, the lexical share rises sharply at global layer 24, and the messiness-4 answer is fully retrieved from layer 30, immediately after global layer 29.

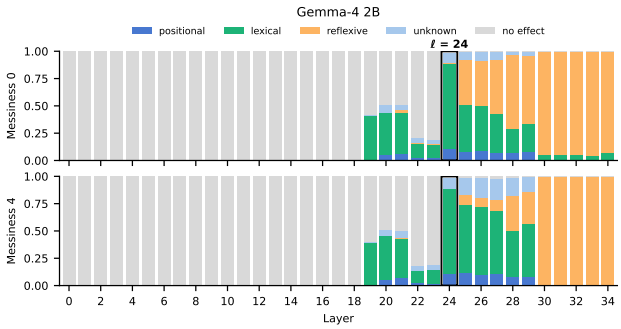

Figure 7: Layer scan for Gemma-4 2B with the messiness-0 (top) and messiness-4 (bottom) counterfactuals. Bars show the share of each mechanism when patching at each layer.

C Mechanism Allocation Extended Results

In this section, we report the full mechanism distribution in Table 7: positional, lexical, reflexive, unknown, and no effect, along with the resulting r_{pos}. While the no-effect rate is generally low, it is substantially higher for some models, particularly Gemma-4 12B, even when patching multiple tokens.

Figure 8 shows r_{pos} separately for each task. The ordering between RoPE and PE hybrids does not hold for every individual task, but emerges in aggregate. Across tasks, however, there is a strong shift in r_{pos} from the most positional models to the most semantic models.

C.1 Full Distribution Table

Model	Positional	Lexical	Reflexive	Unknown	No effect	r_{pos}
RoPE						
Qwen2.5 72B	**46.0**	8.4	16.4	21.9	7.3	1.85
Qwen2.5 7B	**36.0**	16.7	11.4	33.0	2.7	1.28
Qwen2.5 32B	**37.9**	12.8	17.0	29.1	3.2	1.27
Llama-3.1 70B	**28.8**	16.7	10.8	28.5	15.2	1.05
Gemma-2 27B	**33.0**	15.7	17.8	25.1	8.2	0.98
Qwen3 8B	**29.3**	18.9	12.9	35.4	3.5	0.92
Qwen3 14B	**30.5**	20.5	15.0	30.1	3.9	0.86
Qwen2.5 3B	**27.6**	17.2	16.4	36.4	2.3	0.82
Qwen3 4B	**26.3**	18.1	14.1	38.0	3.5	0.82
Qwen3 1.7B	**22.5**	21.2	10.0	43.2	3.2	0.72
Gemma-2 9B	**29.1**	21.9	21.7	25.4	1.9	0.67
Gemma-2 2B	**26.2**	16.9	23.6	31.1	2.2	0.65
Llama-3.1 8B	**24.1**	23.1	17.4	27.4	8.0	0.59
PE hybrid						
Command R7B	21.8	**23.8**	17.5	30.8	6.1	0.53
SmolLM3 3B	18.6	16.1	**21.9**	40.6	2.7	0.49
Gemma-3 4B	22.4	23.5	**23.7**	28.5	1.9	0.48
Gemma-3 12B	21.9	23.3	**34.8**	17.4	2.6	0.38
Gemma-4 31B	21.2	26.9	**29.3**	15.0	7.6	0.38
Gemma-4 4B	17.6	**28.4**	18.5	13.1	22.4	0.38
Gemma-3 27B	20.9	19.7	**42.5**	14.5	2.4	0.34
Gemma-4 12B	12.7	15.1	**25.4**	6.5	40.3	0.31
Gemma-4 2B	17.1	**33.1**	28.0	18.8	3.0	0.28

Table 7: Mechanism distribution per model, as a percentage of all samples (29 task–query pairs, 29,000 samples each), sorted by $r_{pos} = P/(L+R)$. **Bold**/underline: highest/second-highest among Positional, Lexical, and Reflexive.

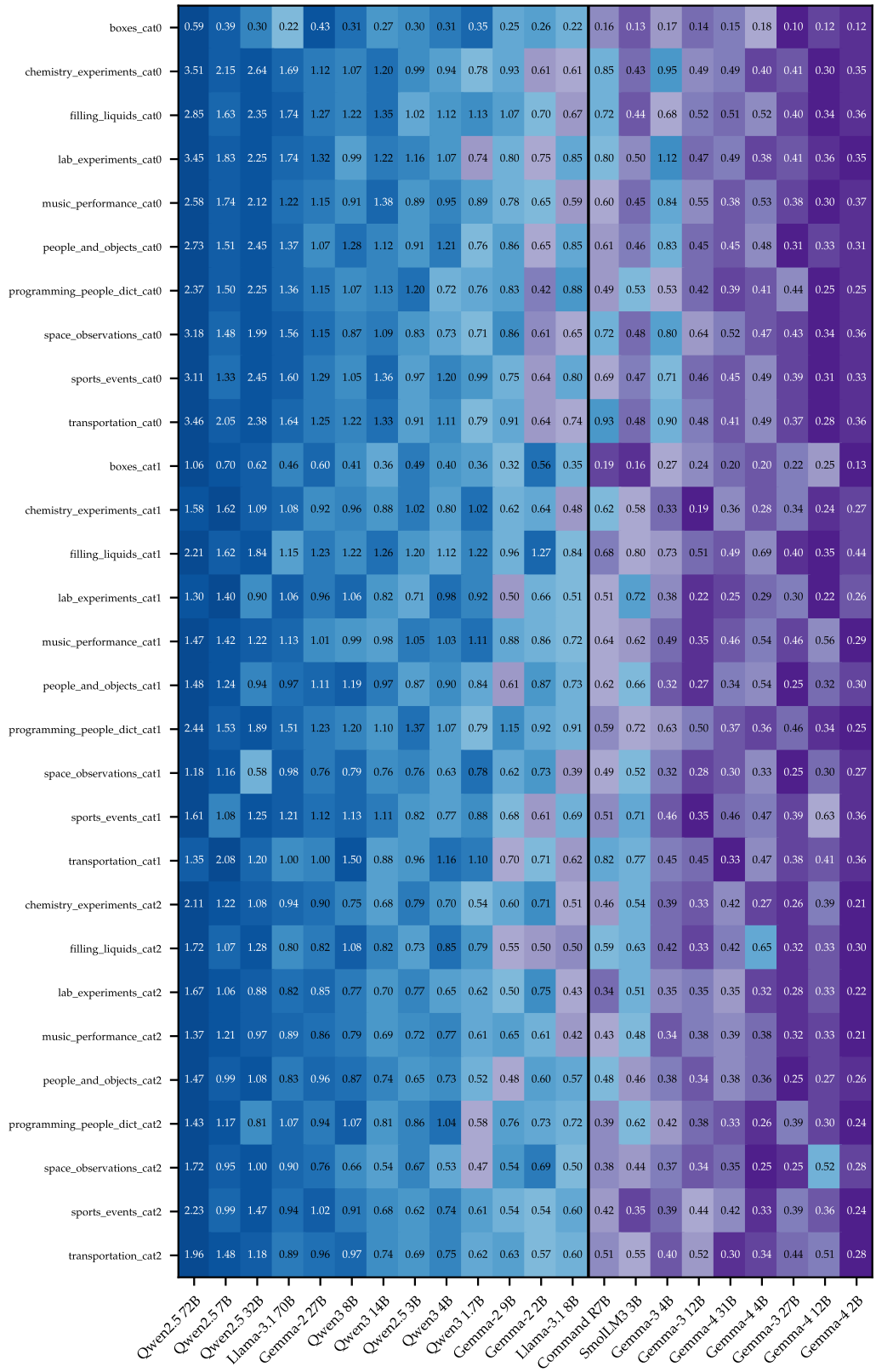

Figure 8: Full per-task breakdown of r_{pos} across the model battery (29 tasks). Separation does not hold for every individual task, but holds when averaged across tasks.

C.2 Mechanism Allocation by Target Entity Position

Figures 9 to 11 show mechanism allocation as a function of target entity position t_{entity} for each model. As shown by Gur-Arieh et al. (2026), t_{entity} controls the tradeoff between lexical and reflexive mechanisms: when the target precedes the query entity in the group, models shift toward reflexive retrieval. This breakdown confirms that our results hold consistently across all query types.

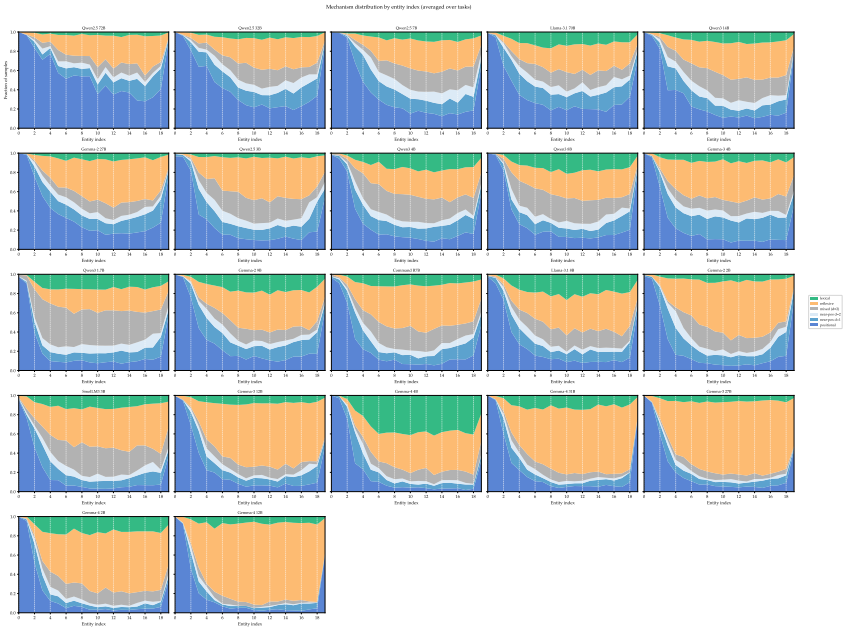

Figure 9: Mechanism allocation by entity index for $t_{entity} = 1$, averaged over tasks. Reflexive retrieval dominates lexical retrieval because the target entity always precedes the query entity.

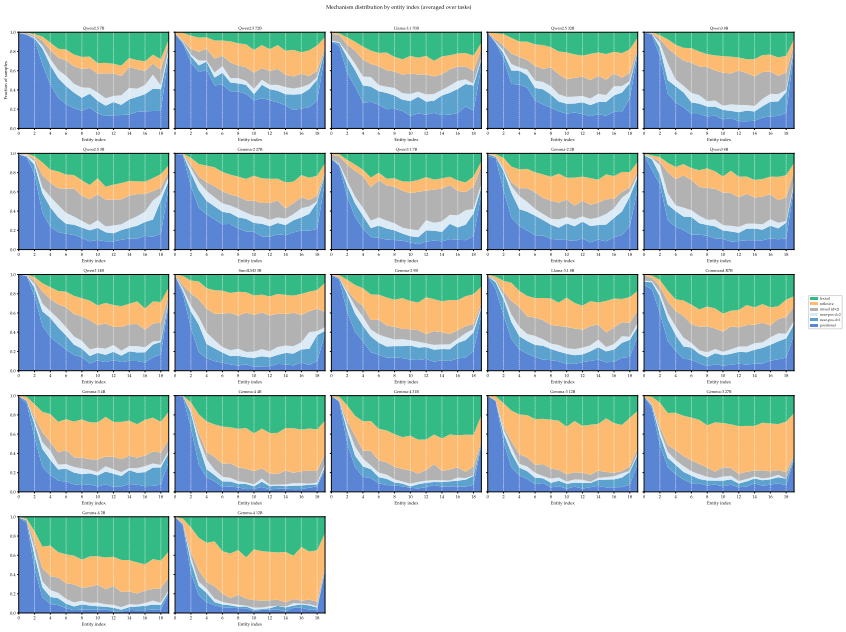

Figure 10: Mechanism allocation by entity index for $t_{entity} = 2$, averaged over tasks.

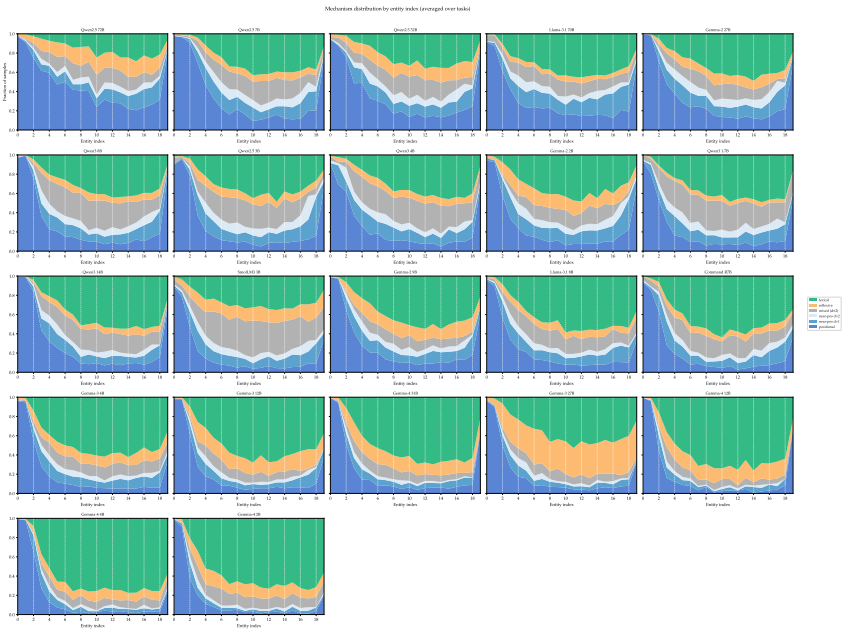

Figure 11: Mechanism allocation by entity index for $t_{entity} = 3$, averaged over tasks. Lexical retrieval dominates reflexive retrieval because the target entity follows the query entity.

D RULER Extended Results

In Table 8, we report the per-task RULER results for the pre-training ablation. The per-task results show that SWA NoPE's gains are concentrated rather than uniform. Its largest improvements occur on several NIAH tasks, particularly S3, MK3, MV, and MQ, as well as both QA tasks. However, SWA NoPE harms or does not improve performance on MK1 and the variable-tracking and aggregation tasks, motivating examining how the shift in retrieval mechanism interacts with the structure of the retrieval task.

Table 8: Per-task RULER scores at 16K and 32K context. $N = 500$ per task. Best result for each task and context length is bolded.

		NIAH								VT + Aggr.			QA	
Context	Model	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2
16K	RoPE	88.2	97.4	**50.0**	**91.2**	83.0	4.0	35.0	35.0	0.1	0.6	**50.1**	50.6	26.8
	SWA RoPE	**100.0**	94.6	36.6	84.8	75.2	11.6	44.0	38.4	**6.1**	**5.9**	43.4	45.8	27.6
	SWA NoPE	**100.0**	**100.0**	48.0	78.0	**84.2**	**41.6**	**80.8**	**85.7**	0.2	3.5	48.7	**54.8**	**32.0**
32K	RoPE	**100.0**	99.6	18.8	**92.4**	**81.2**	0.2	25.2	30.5	14.4	1.1	38.7	33.6	27.8
	SWA RoPE	**100.0**	**99.8**	27.2	81.6	74.0	20.2	26.4	37.9	**14.9**	**9.1**	**45.0**	30.8	26.8
	SWA NoPE	**100.0**	97.6	**77.6**	82.2	68.8	**24.0**	**53.8**	**65.7**	2.5	5.1	39.1	**51.8**	**32.2**

In Table 5 we showed that SWA RoPE degrades on the confusable-key task even though its mechanism allocation stays similar to RoPE. To study why this is, we categorize the incorrect predictions of each model at 32K context length. As shown in Table 9, 82-94% of SWA NoPE's errors at 32K retrieve a competing key's value. In contrast, roughly half of SWA RoPE's errors return no needle value and instead consist primarily of degenerate repetitions. Thus, SWA RoPE's reduced accuracy at 32K reflects a distinct failure mode rather than the competing-key retrieval errors that characterize SWA NoPE.

Table 9: Error types on the confusable-key task at 32K context length. Conditioned on an incorrect answer, we report the percentage of errors that return a value bound to one of the three competing keys (Competing key), versus no needle value (Neither).

		0/4		1/4		2/4		3/4	
Context	Model	Comp. key	Neither	Comp. key	Neither	Comp. key	Neither	Comp. key	Neither
32K	RoPE	94%	6%	83%	17%	86%	14%	76%	24%
	SWA RoPE	43%	57%	39%	61%	52%	48%	53%	47%
	SWA NoPE	94%	6%	86%	14%	82%	18%	83%	17%

E LINEAR HYBRIDS

Many recent open-weight models increasingly use linear attention hybrids (Kimi et al., 2025; Blakeman et al., 2025; Qwen, 2026), often motivated by long-context gains that are primarily computational. A natural question is how these architectures affect in-context retrieval mechanisms.

We briefly examine three linear hybrids. Qwen3.5 4B interleaves Gated DeltaNet layers with full p-RoPE attention at a 3:1 ratio, RecurrentGemma 2B interleaves gated linear recurrence with local p-RoPE attention (window 2048) at a 2:1 ratio, and Granite-4.0-H 1B interleaves Mamba2 layers with full NoPE attention at a 9:1 ratio. We run the full task battery on each using the same protocol as the main experiments.

Results. None of the three exhibits the mechanism allocation characteristic of the PE hybrids in our main battery. Qwen3.5 falls within the RoPE range ($r_{\text{pos}} = 0.72$). RecurrentGemma (0.59) and Granite (0.60) instead lie near the boundary: roughly matching the lowest RoPE model, Llama-3.1 8B (0.59), while remaining above the highest PE hybrid, Command R7B (0.53). Figure 12 breaks down mechanism allocation by entity index. In RecurrentGemma and Granite, reflexive retrieval accounts for under 10% of legible predictions, the lowest of any model we test. Nearly half of the predictions for these models fall outside the three mechanisms. Exploring whether this reflects a retrieval mechanism outside our taxonomy, model scale, or anything else, we leave for future work.

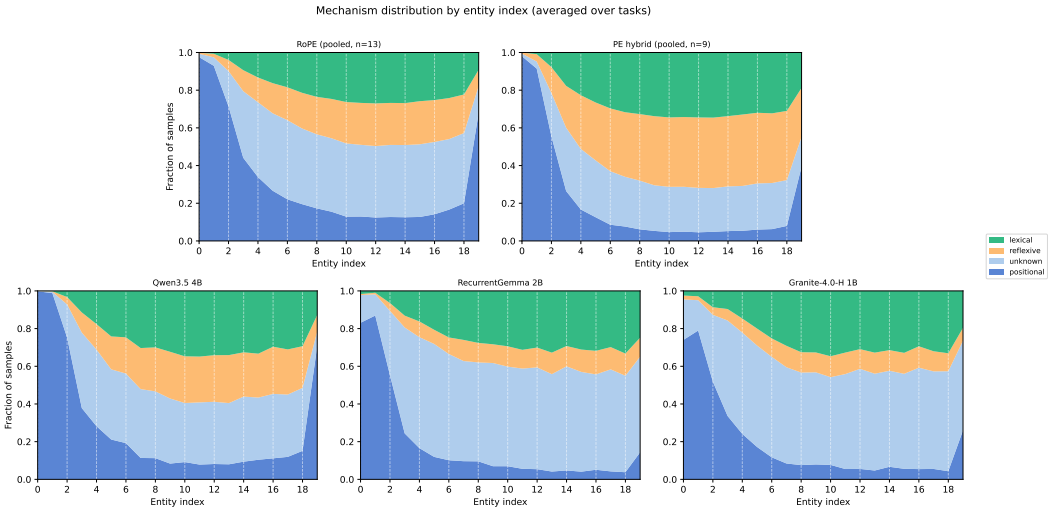

Figure 12: Mechanism allocation across linear hybrids compared to RoPE/PE hybrids.