

A circuit for predicting hierarchical structure in-context in Large Language Models

Tankred Saanum^{1,*†}, Can Demircan^{2,*}, Samuel J. Gershman³, Eric Schulz²

¹Recursive ²Helmholtz Munich ³Harvard University

[†]Correspondence: tankred.saanum@gmail.com ^{*}Equal contribution.

Abstract

Large Language Models (LLMs) rely on induction heads for in-context learning, yet how they manage complex hierarchical structures and contextual dependencies remains unclear. By evaluating a range of LLMs on synthetic tasks and natural language analogues, we identify adaptive induction heads in later layers that learn to attend to successor tokens based on hierarchical cues rather than simple bi-gram statistics. These heads are supported by a circuit of context matching heads that uncover latent transition relationships by routing information from distant past tokens to the current query. Controlled ablation experiments across model families confirm that this circuit is essential for accurate hierarchical predictions in context. Finally, we show that this mechanism is used in natural language to resolve ambiguities where single tokens have multiple potential successors. Our findings provide a mechanistic account of how LLMs learn to predict higher-order repetitive patterns in-context.

1 Introduction

In-context learning is one of the most pervasive features of Large Language Models (LLMs): the ability to predict future tokens more accurately given more context, without any change in model weights. Understanding its internal mechanisms has been a key focus in the community, and a natural starting point for Transformer-based LLMs is the self-attention mechanism, which routes information from past tokens to future ones. In an important study, Olsson et al. (2022) showed that a two-layer attention-only Transformer develops attention heads that make tokens attend to the successors of their past instances. These *induction heads* let the model copy previously observed patterns and token bi-grams, and have since served as a key explanatory mechanism behind in-context learning.

However, induction heads in their basic formulation cannot be the whole story of pattern induction. Many patterns in language have long-range, hierarchical dependencies: if a token x has had two different successor tokens, which should it attend to? For instance, the token *the* prefaces many different nouns in a text, so predicting its successor requires integrating higher-order contextual cues beyond bi-gram statistics. Olsson et al. (2022) indeed reported an induction head with signs of context-sensitive adaptation, raising the possibility that induction heads adapt in-context and account for more than simple copying. Similarly, Akyürek et al. (2024) report induction heads copying successor tokens via 2-gram prefix matching in Transformers trained from scratch on synthetic formal languages.

Expanding on these ideas, we investigate whether induction heads in pre-trained LLMs can learn what successor tokens to attend to in-context when the token sequences have *hierarchical dependencies*. We design a comprehensive set of synthetic token sequences that incorporate repetitive patterns at various levels of hierarchy, along with natural language analogues. In order to predict these token sequences accurately using induction heads, the LLM needs to direct them to attend to specific successor tokens while ignoring others based on these tokens’ preceding context. Remarkably, across all LLMs we evaluate, we discover induction heads in later layers that learn what successor tokens to attend to in-context (see Fig. 1 for task and induction head visualization). We subsequently verify our finding on a simple natural language test, showing how this mechanism is used for sequences closer to the training distribution.

Ultimately, if LLMs learn in-context in virtue of induction heads learning in-context, how do they do it? We propose a simple mechanism explaining how induction heads learn in-context in our task. In our proposed circuit, dedicated heads make tokens attend to the context preceding them, routing infor-

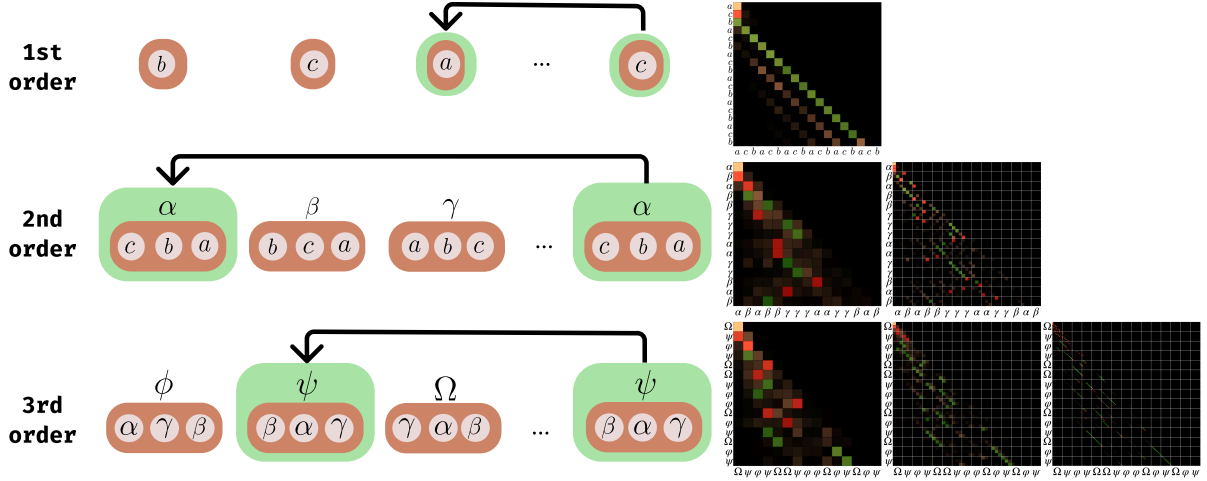


Figure 1: **Left:** Experimental design. First-order structures are learnable by a bi-gram model; second-order structures add context-dependent transitions, requiring the model to identify the current context; third-order structures merge second-order contexts into higher-order ones. **Right:** Attention from an adaptive induction head (layer 19, head 3) in Qwen2.5-1.5B. Green (red) cells mark successor tokens in the same (different) chunk as the current token; the bottom two rows show average attention from a higher-order chunk to other chunks, colored by whether the chunks match. The head moves beyond bi-gram behavior, attending to successors in the correct context.

mation from potentially distant past tokens to allow subsequent context-sensitive attention. From the output of these heads, we can decode whether the current token x_t has the same N preceding context tokens as the *previous* instance of $x_{t' < t}$ after N . These heads could support a representation of the *latent contexts* giving rise to the different successor relationships between the tokens in the sequence. With controlled ablation experiments we confirm that these heads support the in-context learning ability of induction heads. Our results are shown for LLMs in the Qwen2.5 family of models (Yang et al., 2024), and reproduced for four other models, Gemma2-2B (Team et al., 2024), Llama3.2-3B (Dubey et al., 2024), SmolLM3-3B (Bakouch et al., 2025) and Qwen3-0.6B (Yang et al., 2025a) in Appendix C. Our main contribution is to identify this hierarchical circuit, in which context matching heads route distal information so that adaptive induction heads can make context-specific predictions where simple bi-gram copying fails.

2 Related work

Analyses of in-context learning. In-context learning, one of the most pervasive features of modern language models, has been analyzed from various theoretical and empirical points of view. Brown et al. (2020) showed that Transformer language models were generally better at predicting tokens with more examples provided as context across a variety of tasks. Subsequent work focused on mod-

eling the phenomenon of in-context learning in Language models as gradient descent (Von Oswald et al., 2023), Bayesian inference (Xie et al., 2021), latent variable inference (Hendel et al., 2023), as well as in virtue of specific circuits like induction heads (Elhage et al., 2021; Singh et al., 2023) and training distribution (Chan et al., 2022). Recent work has also focused on analyzing in-context learning empirically using tools like Sparse Auto-Encoders and probing (Demircan et al., 2025; Park et al., 2024a,b; Akyürek et al., 2022), or by analyzing LLMs’ propensity to repeat patterns in their inputs (Yan et al., 2023). A related line shows that when LLMs are given sequences generated from latent structures, their internal representations come to reflect that structure (Demircan et al., 2025; Park et al., 2024a; Shai et al., 2024); we build on this by showing that LLMs recover *higher-order* transition structure and represent whether a given chunk matches the preceding one. In our paper we study the internal mechanisms that LLMs rely on in order to pick up on such repetitions when they are embedded in a hierarchy.

Induction and compression. Being able to induce repetitive structure is one of the basic building blocks of compression (Bartík et al., 2015; Sayood, 2017; Delétang et al., 2023; Saanum et al., 2023; Solomonoff, 1964). As shown, this ability is afforded in part by induction heads and context matching heads. Notably, these induction heads show an invariance to the particular token

sequences they are presented with, picking up on repetitive structure even if the sequences are composed of arbitrary tokens that never co-occur in natural language. Invariances like these are crucial for generalization and for the LLMs’ ability to serve as general-purpose compressors (Olshausen et al., 1993; Saanum et al., 2024; Quessard et al., 2020).

Zoo of attention heads for in-context learning.

Various types of attention heads have been identified as important for in-context learning. Elhage et al. (2021) and Olsson et al. (2022) identified induction heads that operate at the token level, with some implications of carrying more abstract functions. Later research (Yin and Steinhardt, 2025) argued that in-context learning is driven by function vector heads (Todd et al., 2023) rather than purely by induction heads. Several other head types support various forms of in-context learning, including concept induction heads (Feucht et al., 2025), semantic induction heads (Ren et al., 2024), n -gram heads (Akyürek et al., 2024) and symbolic induction heads (Yang et al., 2025b). Closest to our circuit are n -gram heads and function vector heads. n -gram heads predict the n -th token by literal, contiguous matching of the preceding $n-1$ tokens in models trained directly on n -grams; we instead analyze pretrained LLMs on *abstracted*, *non-contiguous* hierarchical contexts, where the cue is a latent higher-order chunk rather than a strict string match (Section 4.2), and find the work split across *two* head types rather than one. *Function vector heads* are complementary: they handle abstract mappings such as translation, whereas our circuit handles next-token retrieval in repetitive structures. We detail this two-part circuit, and how it relates to the original induction-head account, in Sections 4 to 5.1.

3 Task

3.1 Synthetic data

To investigate how LLMs learn hierarchical structure in-context, we design token sequences where repetitions appear at various levels of hierarchy. In the simplest case, let us consider a sequence $C = \langle a, b, c \rangle$ which is repeated N times to form $C' = \langle C_1, \dots, C_N \rangle$. Induction heads are perfectly fit to capture sequences like these. Each token has a unique successor token, and is repeated in a completely predictable matter without the need to consider higher-order dependencies.

Next, let us incorporate higher-order dependencies, e.g. where successor tokens can be predicted only by additionally considering the tokens that immediately precede them. Suppose we have three different token sequences like the one designed above $\alpha = \langle a, b, c \rangle$, $\beta = \langle b, c, a \rangle$, $\gamma = \langle c, b, a \rangle$. We refer to these token sequences as *2nd order chunks*. Now we can construct a new sequence where we randomly transition between our 2nd order chunks α , β and γ . Since these consist of the same tokens in different orders, simply attending to any successor token will not be a successful strategy for prediction. For instance, the c token is succeeded by a in the β chunk, and by b in the γ chunk. To this end, if induction heads are responsible for in-context learning in these cases, they have to learn what successor tokens to attend to. We refer to sequences composed of 2nd order chunks as *2nd order sequences*.

Finally, let us consider repetition at one more level of hierarchy. Like in the previous paragraph, we can treat the building blocks of our 2nd order sequences α , β , γ as primitive and compose a more complex vocabulary from them. Consider the set of *3rd order chunks* as follows: $\phi = \langle \alpha, \gamma, \beta \rangle$, $\psi = \langle \beta, \alpha, \gamma \rangle$, $\Omega = \langle \gamma, \alpha, \beta \rangle$. We can compose a new sequence with predictable, repetitive patterns by randomly transitioning between the 3rd order chunks. Here, too, simply attending uniformly to successor tokens is futile: For tokens embedded in the 2nd order chunks α , β or γ , the induction head has to attend to successor tokens from the same 2nd order structures. For tokens that transition *between* the 2nd order structures, the induction heads need to attend to successor tokens in the right position within the same 3rd order chunk ϕ , ψ and Ω . We refer to sequences composed of 3rd order chunks as *3rd order sequences*.

In this paper we evaluate models on sequences in these three levels of hierarchy, with a focus on the last two. To create them, we fix a small vocabulary, a randomly sampled subset consisting of V tokens drawn from the LLM’s original vocabulary. We then generate P unique permutations of the new vocabulary, resulting in P sequences of length V . To create what we call a 2nd order sequence, we repeat each permutation N times and shuffle their order (while keeping the 2nd order chunks intact), giving us a total sequence length of $N \times P \times V$. Finally, for the 3rd order chunks we construct P' unique permutations of the 2nd order chunks. We then compose a new sequence by randomly shuf-

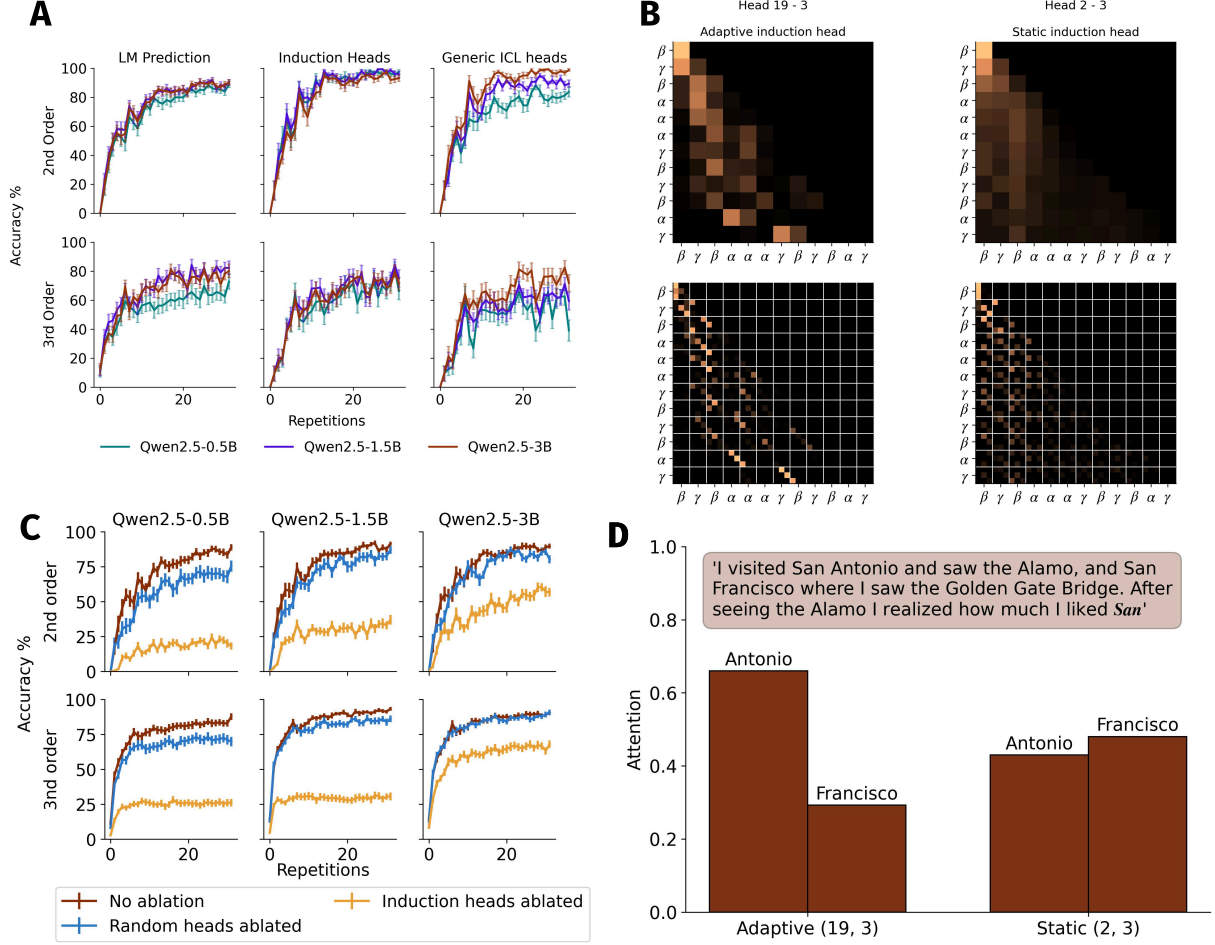


Figure 2: Some, but not all, induction heads learn to attend to successors in higher-order contexts. **(A)** Accuracy on the 2nd-order (top) and 3rd-order (bottom) tasks; learning curves average the best 5 heads. **(B)** Two strategies, each shown as a token-level map (bottom, $x_i \rightarrow x_j$) and an aggregated chunk-level map (top). The adaptive head (left) allocates attention to a token’s successor only in a matching preceding context; the static head (right) attends to all past instances regardless of hierarchy. **(C)** Ablating induction heads severely reduces accuracy and in-context learning across all LLMs and both sequence orders. **(D)** Adaptive heads also identify the correct successor in natural language, whereas earlier static heads attend more uniformly to both candidates. Accuracies in **A** and **C** are averaged over 32 sequences; error bars are SEM.

fling N repetitions of the 3rd order chunks, giving us a total sequence length of $N \times P' \times P \times V$. Example prompts for 2nd and 3rd order sequences, along with task parameters, are shown in Appendix A.7 and Table 1. We give a formal statement of this data-generating procedure in Appendix A.1.

4 Induction heads learn to attend in-context

We evaluated three LLMs from the Qwen2.5 family (Yang et al., 2024), with 0.5, 1.5 and 3 billion parameters, on the synthetically generated sequences from the three levels of hierarchy described in Section 3. We focus on the Qwen2.5 family in the main text because it spans several sizes with a consistent

architecture, letting us study how qualitative effects (e.g. the behavior of individual heads) scale with model size; our findings replicate across Gemma2, Llama3.2, SmolLM3 and Qwen3 (Appendix C). Notably, since the sequences were composed of random tokens from the vocabulary and were generated using the procedure above, they never resembled natural language text. Despite this, all models learned to predict the sequences accurately through in-context learning (see Fig. 2 A). This suggests that there are dedicated circuits for discovering and predicting structured data patterns, even if the data do not resemble natural language.

Next, we assessed if the models had induction heads that learned to attend to the *correct* successor tokens in-context. To do this we first determined

which of a model’s attention heads were induction heads. Following (Olsson et al., 2022), we classified induction heads based on the normalized dot product between each attention head’s attention matrix and an ideal induction head mask, in which each token attended to their past successor token, for a sequence with random tokens repeated twice (see Appendix A.3) (Olsson et al., 2022). This analysis left us with a pool of induction heads per model. We then assessed whether these heads attended to successor tokens from the *correct* context, which we formalize in Appendix A.2.

Remarkably, in all models we evaluated we discovered several induction heads that learned to attend to the correct successor tokens in-context, based on the above criterion for correctness. Overall, this shows that more sophisticated induction heads exist in LLMs. Induction heads that learned in-context were located in later layers, whereas induction heads that did not show signs of learning were embedded in earlier layers (we quantify this depth–learning relationship in Appendix A.8, finding that deeper induction heads learn significantly better, with $\rho = .39$ for 2nd-order and $\rho = .64$ for 3rd-order sequences). We showcase example attention maps of both adaptive, or learning, induction heads as well as their static counterparts in Fig. 2B. We also repeated this analysis for 2nd and 3rd order sequences with randomly sampled chunk lengths to control for positional periodicity as per (Wang and Sato, 2025) in Appendix A.4.

4.1 Induction head learning explains in-context learning in our task

Next we assessed whether induction heads were responsible for the LLMs learning to predict our synthetic token sequences in-context. To test this we ablated all induction heads by setting the output of these heads’ self attention mechanism to a vector of zeros. As a control condition we repeated this experiment, but ablated an equal number of randomly sampled non-induction heads. We confirmed that the induction heads were responsible for the in-context learning in the LLMs. When they were ablated, the 0.5B and 1.5B parameter models predicted tokens at chance level. The 3B parameter model was still better than chance after the ablation, but its performance was severely reduced. The control ablation on the other hand produced small or negligible reductions in accuracy (see Fig. 2C).

4.2 Natural language example

Finally, to verify that the adaptive mechanism operates in natural language, we evaluate Qwen2.5-1.5B on a simple sentence construct. As noted above, tokens like the or prefixes like San (San Francisco) and New (New York) can take many successors within a single text:

Prompt

I visited San Antonio and saw the Alamo, and San Francisco where I saw the Golden Gate bridge. After seeing the Alamo I realized how much I liked San [PREDICTION]

If induction heads are to aid prediction here, the model must use context (from the prompt or pretraining) to direct the San token to the correct Antonio token. We inspected two induction heads: Head 19-3¹, which learned strongly on the synthetic task, and Head 2-3, which did not. The adaptive head attends more to Antonio than Francisco, whereas Head 2-3 attends roughly equally to both (Fig. 2D). Aggregate evidence over 32 prompts is given in Appendix A.9.

5 Learning the building blocks of the hierarchy

So far we have shown that LLMs predict structured, hierarchical patterns in-context using induction heads that learn what to attend to. But explaining in-context learning through another in-context learning mechanism raises the question: how do *induction heads themselves* learn? In our 2nd-order sequences, the token transitions are determined by the structure of the P 2nd-order chunks $(\alpha, \beta, \dots)$; in 3rd-order sequences, by both the 2nd- and 3rd-order chunks. If the model represents which latent context a token belongs to, it can form keys and queries that let induction heads attend to the appropriate successors. To test whether the models recover these latent contexts, we trained linear probes to decode, from a token’s representation, whether its latent context (e.g. α) was identical to that of the previous context.

We trained probes to decode this binary variable from token representations associated with each attention head $\mathbf{z}$. Specifically, each attention head produces a representation for each token $\mathbf{z}_i$, which is the sum of all J tokens’ value vectors $\mathbf{v}_j$ multi-

¹We index attention heads as *layer-head*, both 0-indexed.

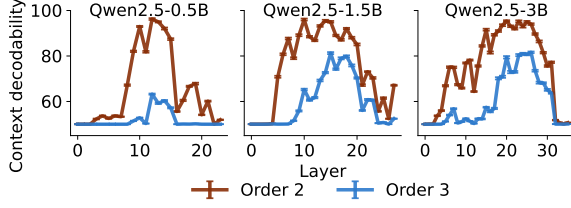


Figure 3: Linear probes decoding, from each head’s representation (Eq. (1)), whether the *previous* 2nd/3rd-order chunk matched the current one. Lines show the max test decodability per layer. 2nd-order decodability is high for all models; 3rd-order decodability improves with model size. Error bars are SEM over 10 train/test splits.

plied with how much token i attends to token j in that particular head $a_{i,j}$:

$$\mathbf{z}_i = \sum_{j=0}^J a_{i,j} \mathbf{v}_j \quad (1)$$

Probes were trained on $\mathbf{z}$ averaged within each context and evaluated on a held-out set; high accuracy indicates the representation encodes whether the previous n -order chunk matched the current one.

Many heads produced representations encoding these 2nd- and 3rd-order chunk identities (Fig. 3); we name these *context matching heads*. Several showed 2nd-order decodability above 90%. 3rd-order decodability was lower but well above chance, and always emerged in later layers than the high-2nd-order heads, consistent with the models building 2nd-order representations before 3rd-order ones.

5.1 Building blocks are learned by context matching heads

What do the attention maps of these context matching heads look like? In Fig. 4 we visualize two heads from Qwen2.5-1.5B with $> 90\%$ 2nd-order decodability, evaluated on shorter 2nd- and 3rd-order sequences. Head 13-4 makes each token attend to its predecessor. Such predecessor heads are known to pair with induction heads, theorized to *copy* previous-token representations forward to enable the induction mechanism. Our analyses suggest a more general role: by iteratively routing information from past tokens, they build up the current token’s latent context, which is why 2nd- and 3rd-order chunk identities are decodable from them. We also find a variant (Head 14-8) in which

tokens attend not only to their direct predecessor but to the N preceding tokens.

We hypothesized that such heads create the latent-context representations that let induction heads attend to the right context. To test this, we ablated head 13-4 (the off-diagonal, look-one-token-back head with $> 90\%$ decodability) and examined induction head 14-3 in the next layer. If 13-4 only copied previous-token information, ablating it should reduce 14-3’s attention to successors in general. Instead, 14-3 still attended to successors but almost entirely lost its ability to attend to the *correct* context (Fig. 6). This indicates that the one-token-back mechanism integrates higher-order contextual cues, much as chains of such heads can route distal context forward like a sliding-window mechanism (Beltagy et al., 2020).

Mechanistically, context matching heads write the token’s current context into the residual stream; adaptive induction heads then use this to update their query and attend only to keys sharing that context, resolving bi-gram ambiguity.

5.2 In-context learning suffers from ablating context matching heads

To test more broadly whether context matching heads drive the in-context learning of induction heads, we zeroed out the output of all context matching heads (latent-context decoding score $> 85\%$) and measured the effect on LLM prediction accuracy and induction-head attention accuracy, comparing against ablating an equal-sized random population of other heads (see A.5). Scores are averaged over 32 samples.

Ablating context matching heads sharply reduced prediction accuracy across all models, far more than ablating random heads, whose effect was almost negligible for Qwen2.5-0.5B (Fig. 5, left). The induction heads’ in-context learning also dropped consistently across 2nd- and 3rd-order sequences (Fig. 5, right), indicating that induction heads learn with the help of context matching heads that peer back to infer the latent token-to-token transitions.

6 Depth enables hierarchical chunk learning

Our mechanistic account makes a direct architectural prediction: because the circuit must stack context matching and induction across levels of the hierarchy, learning 3rd-order dependencies in-

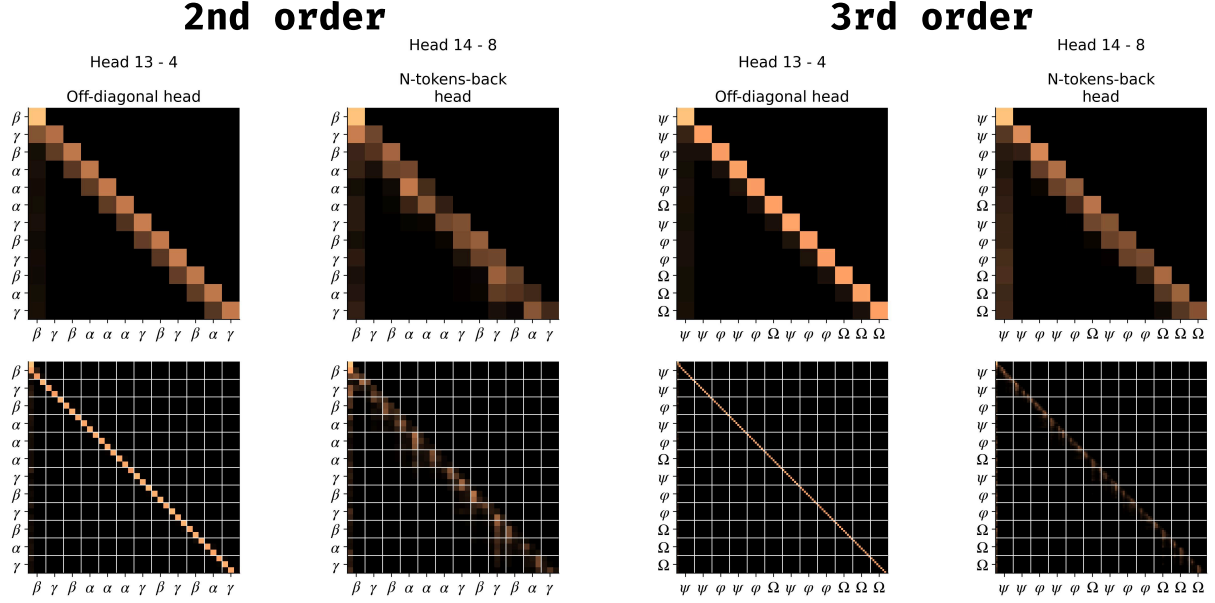


Figure 4: Two context matching heads with high latent-context decodability in Qwen2.5-1.5B, each shown with token-to-token transitions (bottom) and aggregated chunk-to-chunk relationships (top). Head 13-4 makes each token attend to its direct predecessor, propagating previous-token information forward to support the induction mechanism. Head 14-8 instead propagates information from longer chains of tokens, potentially letting induction heads match n -grams before attending to successors from the appropriate context.

context should require sufficient depth. In principle, a 2-layer circuit whose first-layer context-matching head recognizes all n -grams/chunks could already solve our tasks (Akyürek et al., 2024; Varre et al., 2025). Empirically, however, 2-layer Transformers (40M parameters, 16 heads/layer) are poorly suited to learning novel 3rd-order dependencies, whereas 4-layer Transformers (40M parameters, 8

heads/layer) are not. Training both from scratch on procedurally generated 3rd-order sequences (3 seeds, novel sequences at each step; fixed and variable chunk lengths) for 8 billion tokens, we find that the 2-layer model is substantially worse at next-token prediction (Fig. 7A) and, crucially, fails to develop induction heads that attend to successors from the correct 3rd-order context, unlike the

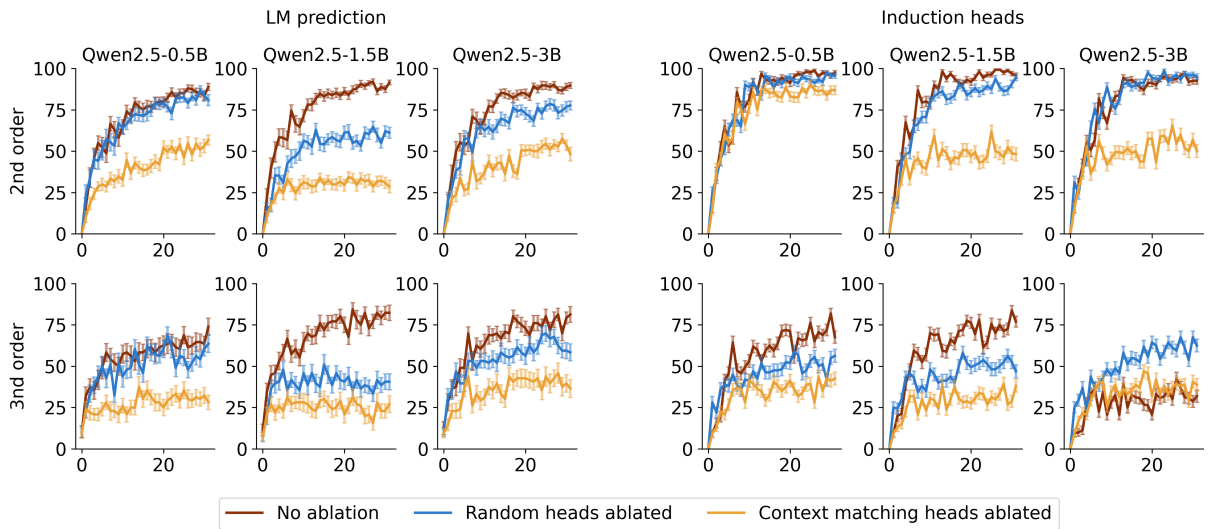


Figure 5: **(Left)** In-context predictive accuracy suffers from ablating context matching heads, far more than from ablating equally many random non-context-matching heads, across all models. **(Right)** Induction-head accuracy at attending to successors from the correct context drops similarly. Lines are means over 32 sequences; error bars are SEM.

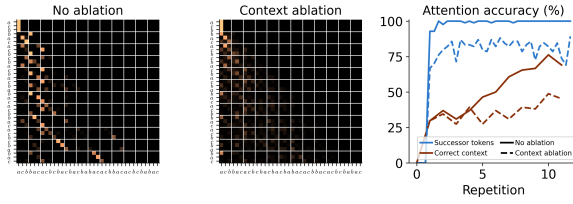


Figure 6: Ablating a single context matching head sharply reduces the subsequent induction head’s ability to attend to successors in the *correct* context, while leaving its attention to successors in general mostly intact. Heatmaps show head 14-3 on an example sequence; lines are means over 84 2nd-order sequences (chance 33%).

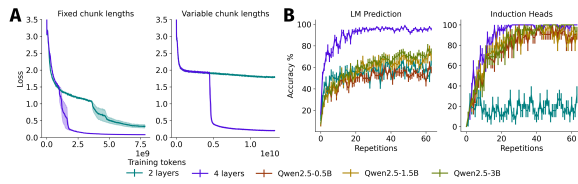


Figure 7: Depth is required for 3rd-order in-context learning. (A) 4-layer Transformers outperform 2-layer Transformers when trained from scratch to predict 3rd-order chunks in-context. (B) The 4-layer models develop adaptive induction heads like the pretrained LLMs, whereas the 2-layer models do not.

4-layer model (Fig. 7B). The adaptive induction circuit is thus not only present in large pretrained LLMs but emerges with depth, indicating that sufficient depth is necessary for hierarchical in-context learning.

7 Conclusion

We make two contributions to understanding in-context learning in LLMs. First, LLMs learn hierarchical repetitive structures using induction heads that learn what to attend to in-context. Second, these induction heads learn through an accompanying circuit of context matching heads that recover the latent contexts underlying the token transitions: they make tokens attend to preceding tokens, or to longer chains of them, routing information that lets induction heads query successors with similar preceding chains. Induction heads thus offer a unifying account of how LLMs predict in-context patterns, including natural-language cases such as the token *the*, which precedes many nouns.

Limitations

While we have argued that induction heads, with the supporting context matching heads, can give

a unifying account of LLMs’ ability to induce repetitive structures with hierarchical dependencies, there are other types of in-context learning that we have not explored. For instance, problems from the Abstraction and Reasoning Corpus (Chollet, 2019) require LLMs to learn abstract relationships between tokens in few examples. It is unclear if our proposed circuit is powerful enough to induce abstract relationships like these. Our study paves the way for future work to study induction head behavior in settings such as these.

References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.
- Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. 2024. *In-context language learning: Architectures and algorithms*. *Preprint*, arXiv:2401.12973.
- Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Noumane 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, and 4 others. 2025. SmolLM3: smol, multilingual, long-context reasoner. <https://huggingface.co/blog/smollm3>.
- Matěj Bartík, Sven Ubik, and Pavel Kubalik. 2015. Lz4 compression algorithm on fpga. In *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 179–182. IEEE.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891.
- François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.
- Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang,

- Matthew Aitchison, Laurent Orseau, and 1 others. 2023. Language modeling is compression. *arXiv preprint arXiv:2309.10668*.
- Can Demircan, Tankred Saanum, Akshay Kumar Jagadish, Marcel Binz, and Eric Schulz. 2025. [Sparse autoencoders reveal temporal difference learning in large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2021/framework/index.html>.
- Sheridan Feucht, Eric Todd, Byron Wallace, and David Bau. 2025. [The dual-route model of induction](#). Preprint, arXiv:2504.03022.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.
- Bruno A Olshausen, Charles H Anderson, and David C Van Essen. 1993. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.
- Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. 2024a. Iclr: In-context learning of representations. *arXiv preprint arXiv:2501.00070*.
- Core Francisco Park, Ekdeep Singh Lubana, Itamar Pres, and Hidenori Tanaka. 2024b. Competition dynamics shape algorithmic phases of in-context learning. *arXiv preprint arXiv:2412.01003*.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben al-lal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. [The fineweb datasets: Decanting the web for the finest text data at scale](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Robin Quessard, Thomas Barrett, and William Clements. 2020. Learning disentangled representations and group structure of dynamical environments. *Advances in Neural Information Processing Systems*, 33:19727–19737.
- Jie Ren, Qipeng Guo, Hang Yan, Dongrui Liu, Quanshi Zhang, Xipeng Qiu, and Dahua Lin. 2024. [Identifying semantic induction heads to understand in-context learning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6916–6932, Bangkok, Thailand. Association for Computational Linguistics.
- Tankred Saanum, Peter Dayan, and Eric Schulz. 2024. Simplifying latent dynamics with softly state-invariant world models. *Advances in Neural Information Processing Systems*, 37:38355–38382.
- Tankred Saanum, Noémi Éltető, Peter Dayan, Marcel Binz, and Eric Schulz. 2023. Reinforcement learning with simple sequence priors. *Advances in Neural Information Processing Systems*, 36:61985–62005.
- Khalid Sayood. 2017. *Introduction to data compression*. Morgan Kaufmann.
- Adam Shai, Lucas Teixeira, Alexander Oldenziel, Sarah Marzen, and Paul Riechers. 2024. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034.
- Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. 2023. The transient nature of emergent in-context learning in transformers. *Advances in neural information processing systems*, 36:27801–27819.
- Ray J Solomonoff. 1964. A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2023. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.
- Aditya Varre, Gizem Yüce, and Nicolas Flammarion. 2025. Learning in-context n-grams with transformers: Sub-n-grams are near-stationary points. *arXiv preprint arXiv:2508.12837*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Shuo Wang and Issei Sato. 2025. Rethinking associative memory mechanism in induction head. In *Second Conference on Language Modeling*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Jianhao Yan, Jin Xu, Chiyu Song, Chenming Wu, Yafu Li, and Yue Zhang. 2023. Understanding in-context learning from repetitions. *arXiv preprint arXiv:2310.00297*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yukang Yang, Declan Campbell, Kaixuan Huang, Mengdi Wang, Jonathan Cohen, and Taylor Webb. 2025b. Emergent symbolic mechanisms support abstract reasoning in large language models. *Preprint*, arXiv:2502.20332.

Kayo Yin and Jacob Steinhardt. 2025. Which attention heads matter for in-context learning? *Preprint*, arXiv:2502.14010.

A Appendix: Implementation details

A.1 Synthetic data generation

The parameters that were used to generate the synthetic data are shown in Table 1. The sequence parameters were shared across learning, ablations, and decoding experiments. The P' parameter was only used to generate the 3rd order sequences. In the 3rd order sequences, the length of the 2nd order chunks (V) was halved to avoid prohibitively long sequences. We used a larger batch size in order to train the linear probes for the decoding analysis. For the qualitative demonstrations (i.e., Fig. 1, 2B, 4, 6, 9) we used shorter sequences. Example prompts for both 2nd and 3rd order sequences are shown in Fig. 9.

Formal task definition. Let $\mathcal{V}$ be a set of V tokens randomly sampled from the model’s vocabulary, let $\Pi(\mathcal{X}, m)$ denote a set of m unique permutations of a sequence $\mathcal{X}$, and let $\oplus$ denote concatenation. For *2nd-order sequences*, we define P unique 2nd-order chunks $\mathcal{C}^{(2)} = \{c_1, \dots, c_P\} \subset \Pi(\mathcal{V}, P)$ and randomly shuffle N repetitions of each chunk; if $z_t \in \{1, \dots, P\}$ is the chunk index at step t , the resulting sequence is $S^{(2)} = \bigoplus_{t=1}^{N \times P} c_{z_t}$. For *3rd-order sequences*, we define P' 3rd-order chunks as unique permutations of the 2nd-order chunks, $\mathcal{C}^{(3)} = \{C_1, \dots, C_{P'}\} \subset \Pi(\mathcal{C}^{(2)}, P')$, and analogously concatenate N shuffled repetitions of these higher-order chunks.

A.2 Learning head classification

We assessed whether attention heads attended to their successor tokens from the correct contexts. This was done in a cross-validated approach. In each of the 32 folds, we identified the top 5 performing heads from 31 runs and evaluated those heads in the held-out run. For 2nd order sequences, we calculated how often a token a_t attended to a successor of $a_{t' < t}$ where $a_{t' < t}$ was embedded in the same type of 2nd order chunk. For 3rd order sequences, we designed a stricter criterion for learning. We calculated whether tokens that marked predictable transitions between 2nd order chunks attended to successor tokens from previous instances of the same 3rd order chunk.

We refer to these heads as *adaptive induction heads*, and formalize the matching criterion at the level of chunks. For an attention matrix A , where $a_{u,v}$ denotes the attention from token u to token v , we compute a chunk-level matrix B whose entry $B_{i,j}$ is the mean attention from the tokens of the current chunk i to the tokens of a past chunk j :

$$B_{i,j} = \frac{1}{V^2} \sum_{u \in \text{chunk } i} \sum_{v \in \text{chunk } j} a_{u,v}. \quad (2)$$

An induction head correctly matches the latent context at chunk i if the most attended past chunk $j^* = \arg \max_{j < i} B_{i,j}$ shares the current chunk’s latent identity, i.e. $\text{Accuracy}_i = \mathbb{1}[z_i = z_{j^*}]$. Equivalently, a head is correct when it attends most to a past token $x_{t'+1}$ whose N preceding tokens match those preceding x_t . An induction head is *adaptive* if this accuracy improves as more chunk repetitions accumulate in-context, and *static* otherwise.

Experiment	Batch Size	N	P	P'	V
Learning & Ablation	32	8	4	4	8 (4 for 3rd order sequences)
Decoding	64	8	4	4	8 (4 for 3rd order sequences)

Table 1: The parameters used to generate the synthetic data.

A.3 Induction head classification

We identified the induction heads following (Ols-son et al., 2022). To calculate the score for each attention head (l, h) , we constructed an input sequence $\mathbf{S} = [\mathbf{A}, \mathbf{A}]$, where $\mathbf{A}$ is a sequence of L randomly selected tokens. The idealized attention pattern for an induction head, requires that for any token $\mathbf{A}_i$ in the second sequence block, the attention at that position focuses on the token $\mathbf{A}_{i+1}$ in the first sequence block. We construct an idealized induction pattern matrix $\mathbf{M}$.

The induction head score for head (l, h) is then calculated as the normalized dot product between the head’s actual causal attention matrix $\mathbf{A}_{l,h}$ (flattened over the non-zero indices of the causal mask) and the flattened ideal induction mask $\mathbf{M}$. The score is bound between 0 (no induction) and 1 (perfect induction). We classify heads with a score ≥ 0.4 as induction heads.

A.4 Variable Chunk Lengths

We extended our analyses of our fixed length 2nd order and 3rd order chunks and analyzed whether the Qwen models were able to predict these types of sequences when we randomly sampled the chunk lengths for each sequence we generated. Again, evaluating across 32 sequences, we see that all models are able to predict the next token in-context. Moreover, their induction heads show a similar level of accuracy in this setting as in the fixed chunk length condition (see Fig. 8).

A.5 Control ablations

To establish a controlled comparison for the ablations, we ablated an equal number of random heads, ensuring that they were layer-matched with the ablated context matching heads. If a layer did not have enough heads, sampling was done from adjacent layers. If still insufficient, the threshold was progressively relaxed. If there still was a shortfall, the remaining heads were randomly sampled from any layer.

A.6 Decoding analysis

To assess whether the models represented the latent generative contexts, we trained probes to decode from the outputs of each attention head whether a chunk of tokens were generated by the same latent context as the previous chunk of tokens. To obtain the input variables to our decoder we therefore averaged token representations within a context. Next, we optimized an $L2$ -regularized logistic regression model using the `scikit-learn` library (Pedregosa et al., 2011). We used 75% of the data to train the classifier and the remaining 25% to test it. We repeated this procedure using 10 different seeds. Due to the structure of the sequences, the previous context and the current context were more likely to be different, creating a class imbalance. Therefore, the accuracy scores reported in Fig. 3 are balanced accuracy scores.

A.7 Prompt visualization

A.8 Early vs. late induction heads

Induction heads are found throughout different layers of language models. Here, we tested whether induction heads that appear later show stronger learning. In Fig. 10 is plotted the depth of induction heads (with induction head score $\geq .4$) against learning scores in hierarchical sequences, as described in 2. We found that induction heads that are deeper in models show stronger learning both in 2nd ($\rho = .39, p = 0.017$) and 3rd ($\rho = .64, p < 0.001$) order sequences.

A.9 Natural Language

We tested whether adaptive induction heads are successful in resolving ambiguities in natural language. We carried out the same analysis presented in Section 4.2. We provided Qwen2.5-1.5B with 16 unique sentences, where the ambiguity of which token to predict next can only be solved based on the previous context. With counterbalancing the order of the examples, we had 32 different test cases, whose results are shown in Fig. 11, providing further evidence that adaptive induction heads can infer which token to attend to from the context

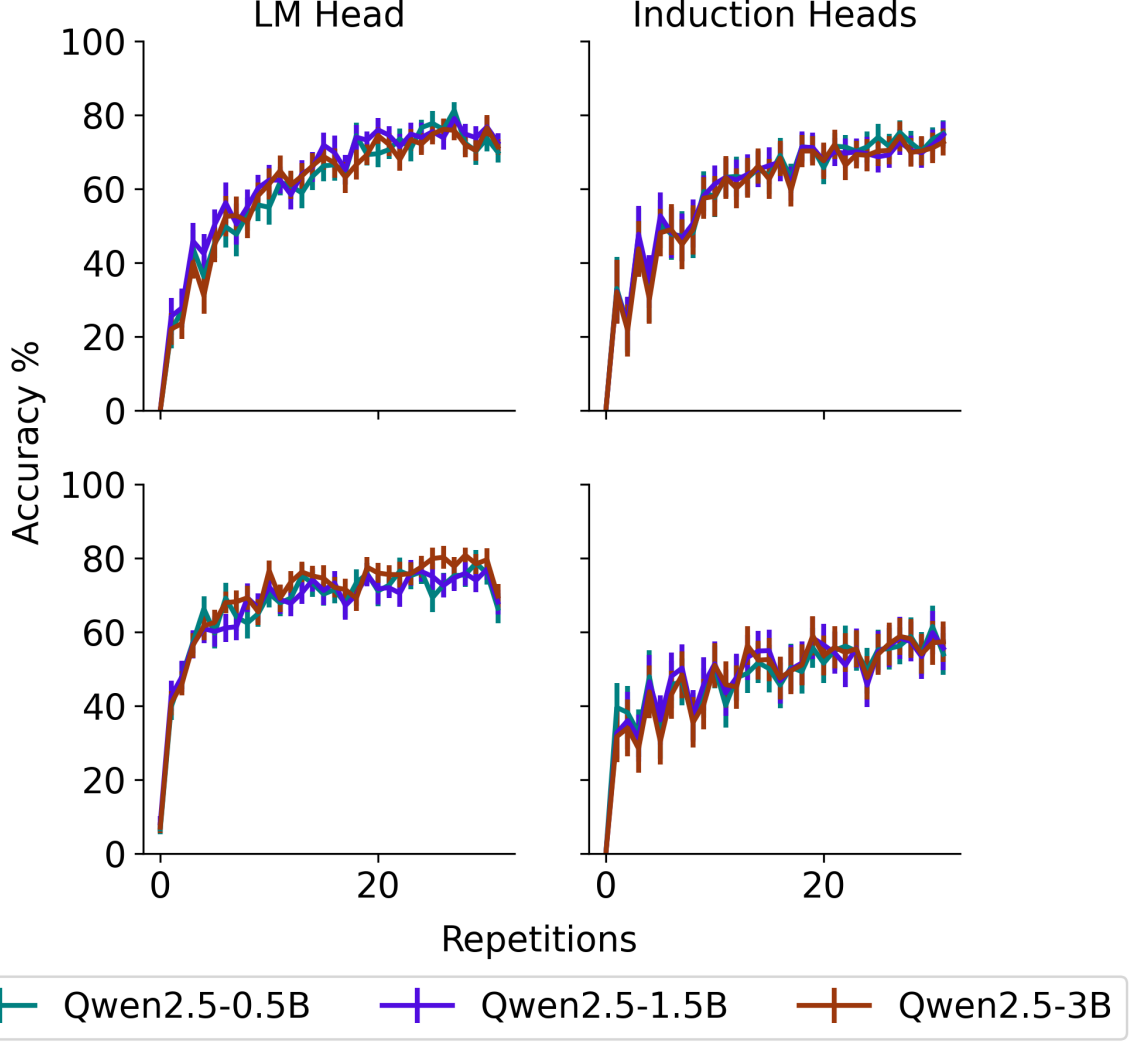


Figure 8: Language model and induction head accuracies per repetition in the variable chunk length setting. Lines are averages of 32 samples, error bars represent the standard error of the mean.

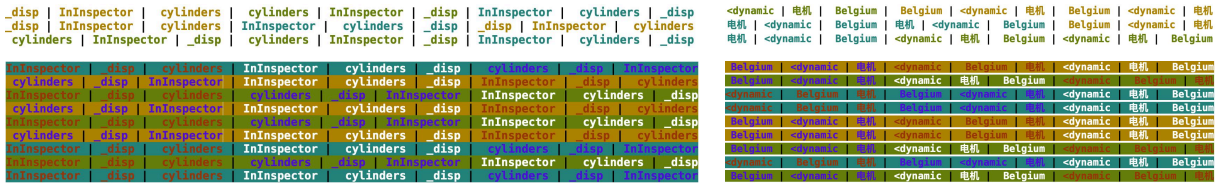


Figure 9: Example prompts. On the upper row are two second order sequences with three unique tokens, three unique chunks, and three repetitions. Tokens are separated by a vertical bar. The colors of the tokens indicate to which unique chunk they belong to. On the lower row are third order sequences, which additionally include three unique higher order chunks. Here we use different background colors to denote different unique higher order chunks. The sequences are split into new lines in the figure only for illustrative purposes.

in ambiguous cases. We provide the prompts that were used in Table 2.

B Appendix: Additional analyses

B.1 Consistency of learning heads

As described in 4, the reported learning scores in Fig. 2 A are calculated in a cross validated fashion. In Fig. 12, we show that the top 5 identified learning heads are near perfectly consistent across

#	Prompt	Correct	Wrong
1	My cousins are David Chen and David Lee. I needed to speak with the one whose last name is only three letters long, so I called David ...	Lee	Chen
2	Both the Statue of Liberty in New York and the French Quarter in New Orleans are famous tourist attractions. Because I love southern cuisine, I decided to visit New ...	Orleans	York
3	The gift box contained an Apple Watch and an Apple iPhone. The item designed to be worn on the wrist was the Apple ...	Watch	iPhone
4	The report was on the Bank of America and the Bank of Canada. Since the focus was on Canadian financial institutions, I wrote about the Bank of ...	Canada	America
5	The curriculum covered both World War 1 and World War 2. The exam question was about the earlier of the two conflicts, which was World War ...	1	2
6	The hotel room had a king-size bed and a king-size pillow. The large piece of furniture I slept on was the king-size ...	bed	pillow
7	I like reading both Stephen Hawking and Stephen King. Yesterday, I felt more like reading fiction, so I read Stephen ...	King	Hawking
8	The lecture contrasted the composers John Lennon and John Williams. I don't like the Beatles, so I focused more on John ...	Williams	Lennon
9	I visited San Antonio and saw the Alamo, and San Francisco where I saw the Golden Gate Bridge. After seeing the Alamo, I realized how much I liked San ...	Antonio	Francisco
10	They had oat bar and oat milk for breakfast. I don't like drinking anything in the mornings, so I took the oat ...	bar	milk
11	My granddad likes flowers and my grandmum likes chocolate. Therefore, I will gift this flower to my grand. ...	dad	mum
12	He asked whether I like best of 5 or best of 3 matches more. I get tired quickly, so I said best of ...	3	5
13	Between the two, planet Fulty has less sunlight than planet Julty. Because I like the sun, I like going to planet ...	Julty	Fulty
14	She could either practice her backhand or backspin. Since she already practiced her backhand yesterday, today she worked on her back. ...	spin	hand
15	I split my time between Bad Tölz and Bad Homburg. I love Bad Tölz in the winters. Since it is now July, I am in Bad ...	Homburg	Tölz
16	I like drinking caffè Americano in the morning and caffè mocha in the afternoon. It is now 3 PM, and I would like to drink caffè ...	mocha	Americano

Table 2: The natural language prompts that were used to test the induction heads. Corresponding correct and wrong answers are also provided.

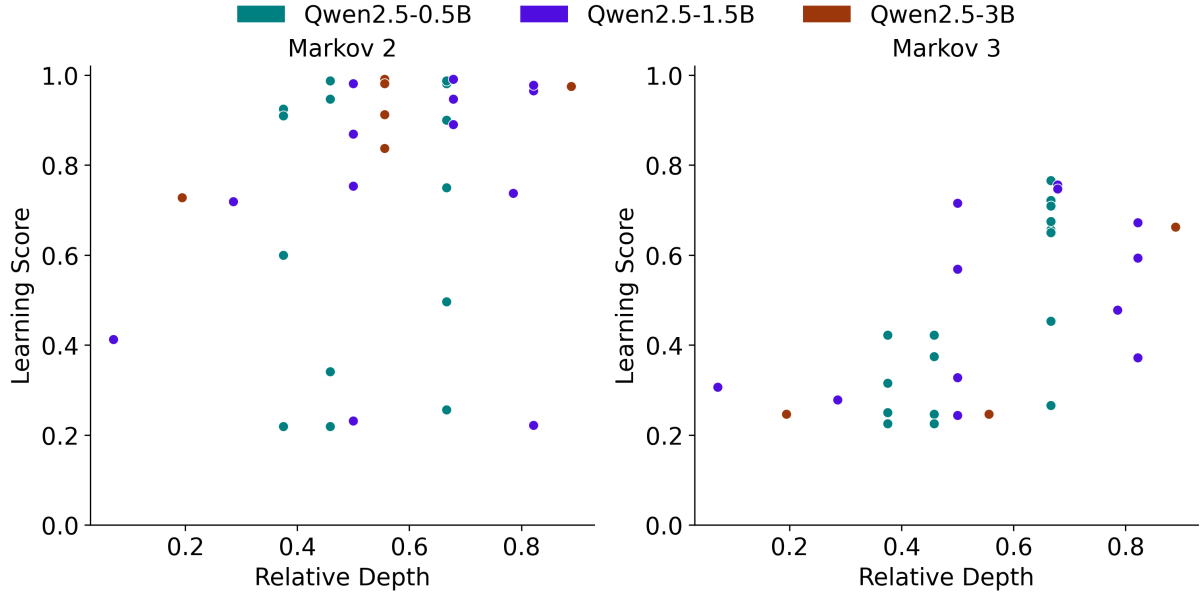


Figure 10: Induction heads that appear deeper in models are better at learning hierarchical sequences.

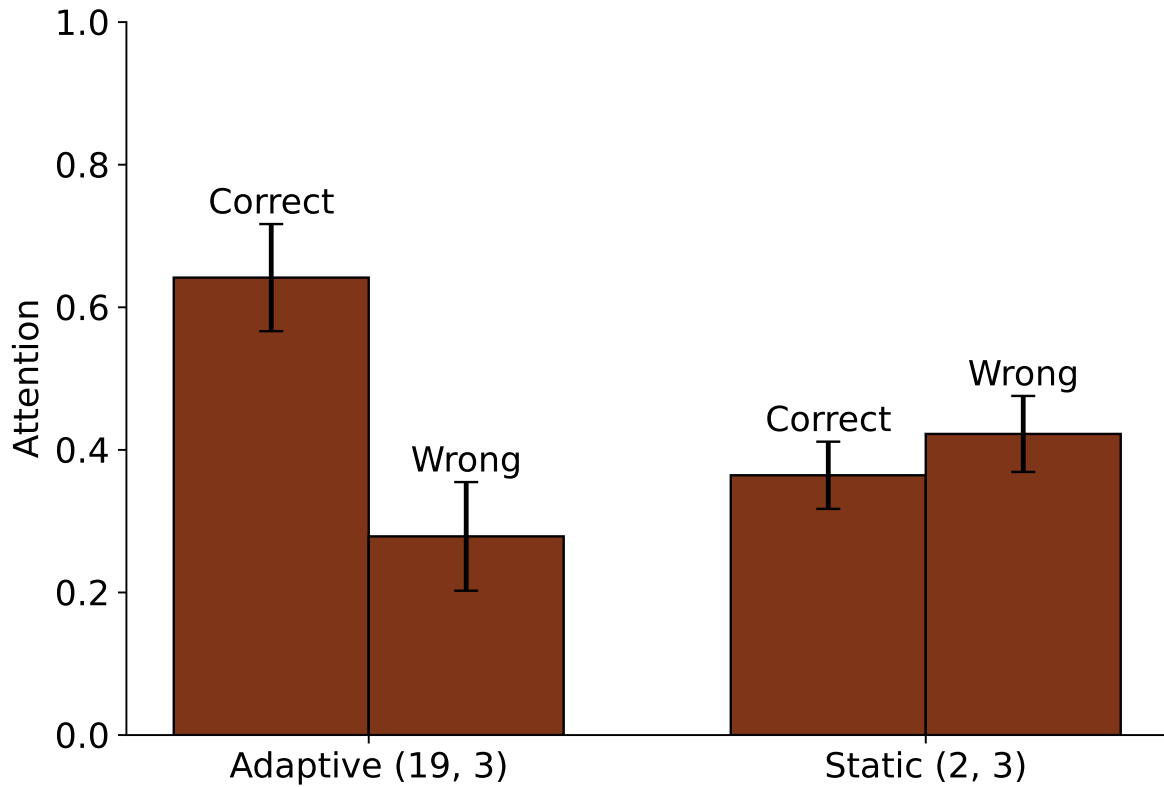


Figure 11: Adaptive induction heads attend to the correct tokens in ambiguous situations. Static induction heads do not have such a preference.

folds.

B.2 Effects of noise, token frequency, and periodicity

We ran a series of additional experiments to test the robustness of our findings. First, we introduced noise into the sequences, where for each token there

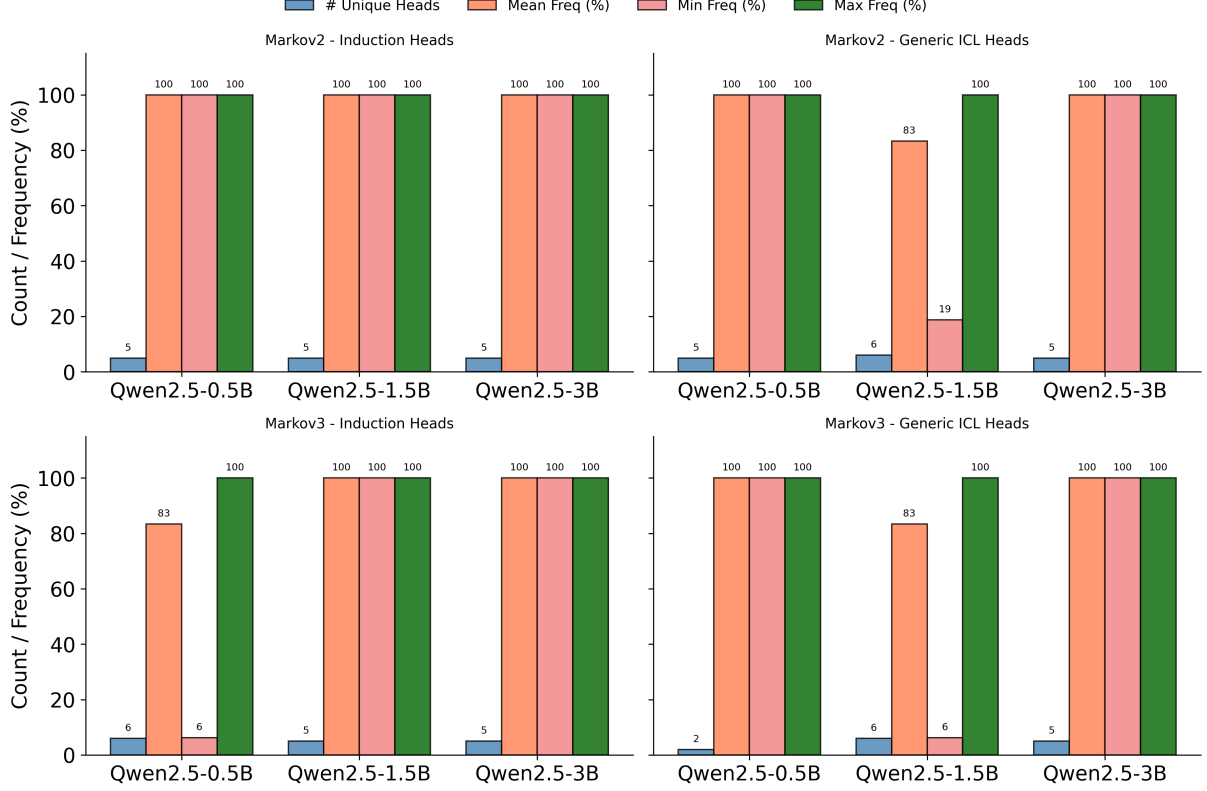


Figure 12: Both in second and third order sequences, the identified learning induction heads and the generic in-context learning heads are consistent across runs.

was a probability of it being randomly changed. As can be seen in Fig. 13, the performance of the models and the induction heads decay gradually as a function of noise.

We additionally tested whether our findings depend on using rare vs common tokens. First, we iterated through approximately 50,000,000 tokens sampled from FineWeb-Edu (Penedo et al., 2024) to identify the most common and least common tokens that appear at least once in a standard pre-training dataset. We then constructed 2nd and 3rd order sequences like described in the main text. 32 sequences were constructed by sampling from the 4162 most common tokens, and 32 from the least common 79331 tokens. As can be seen in Fig. 14, the accuracy differences between the conditions are small.

C Appendix: Replication with other Language Models

To assess the general validity of the proposed circuit, we investigated if *other* open-source LLM families showed the signs of the same algorithmic implementation of hierarchical in-context learning. To this end, we conducted a subset of our experiments in exactly the same manner on four new LLMs, Gemma2-2B (Team et al., 2024), Llama3.2-3B (Dubey et al., 2024), SmolLM3-3B (Bakouch et al., 2025) and Qwen3-0.6B (Yang et al., 2025a). Specifically, we sought to assess 1) if these models also have induction heads that learn in-context. 2) if these models have context matching heads from which one can decode the latent generative contexts in our task. And 3), if these context matching heads were causally involved in the in-context learning ability of the induction heads. To evaluate the last point, we again ablated all context matching heads whose latent context decodability was higher than 85%, and observed how this affected model accuracy and induction head accuracy. We also ablated random heads as a control condition, exactly like in Section 5.1. The results presented for the Qwen2.5 models generally reproduced with striking levels

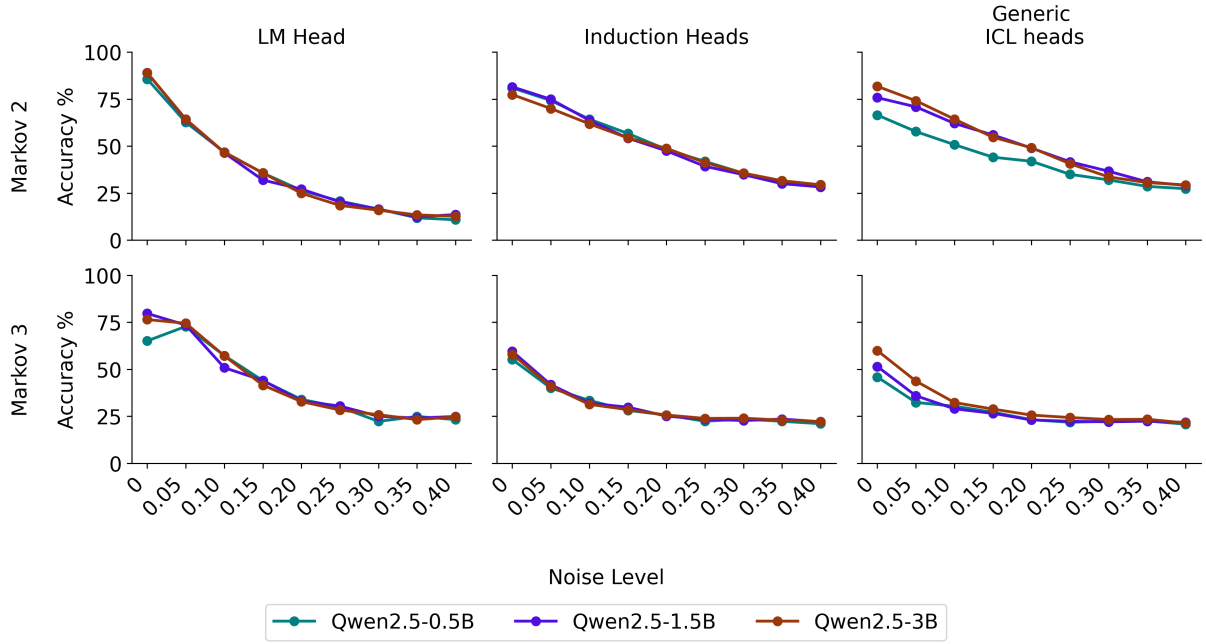


Figure 13: The performance of the models and the induction heads decay gradually as a function of noise.

of consistency. All models had induction heads that learned in-context, had heads that encoded the latent generative contexts, and that were causally linked to the in-context learning ability of the LLM and the induction heads more specifically. The results are shown individually for each model below.

C.1 Statistical testing and 3rd-order variance

We compared context-matching against random-head ablation with a paired permutation test (10,000 sign flips of the per-sequence accuracy difference across the 32 sequences, one-sided). Context matching ablation significantly outperformed the random control in 6 of the 8 conditions.

While the 2nd-order adaptive induction circuit is robustly identified across all model families, the 3rd-order results exhibit higher variance. In Llama-3.2-3B and Gemma-2-2B, the baseline 3rd-order induction accuracy is lower than in Qwen, and the ablation of context matching heads, while directionally consistent, is not always statistically significant. Three 3rd-order tests were non-significant: the Gemma induction head (context-matching ablated: 0.301 vs. random ablated: 0.296, $p = 0.72$), Llama LM prediction (0.393 vs. 0.438, $p = 0.30$), and the Llama induction head (0.219 vs. 0.244, $p = 0.17$). This suggests that while the foundational hierarchical circuit is universal, models vary in their ability to recursively stack this mechanism for deeper dependencies.

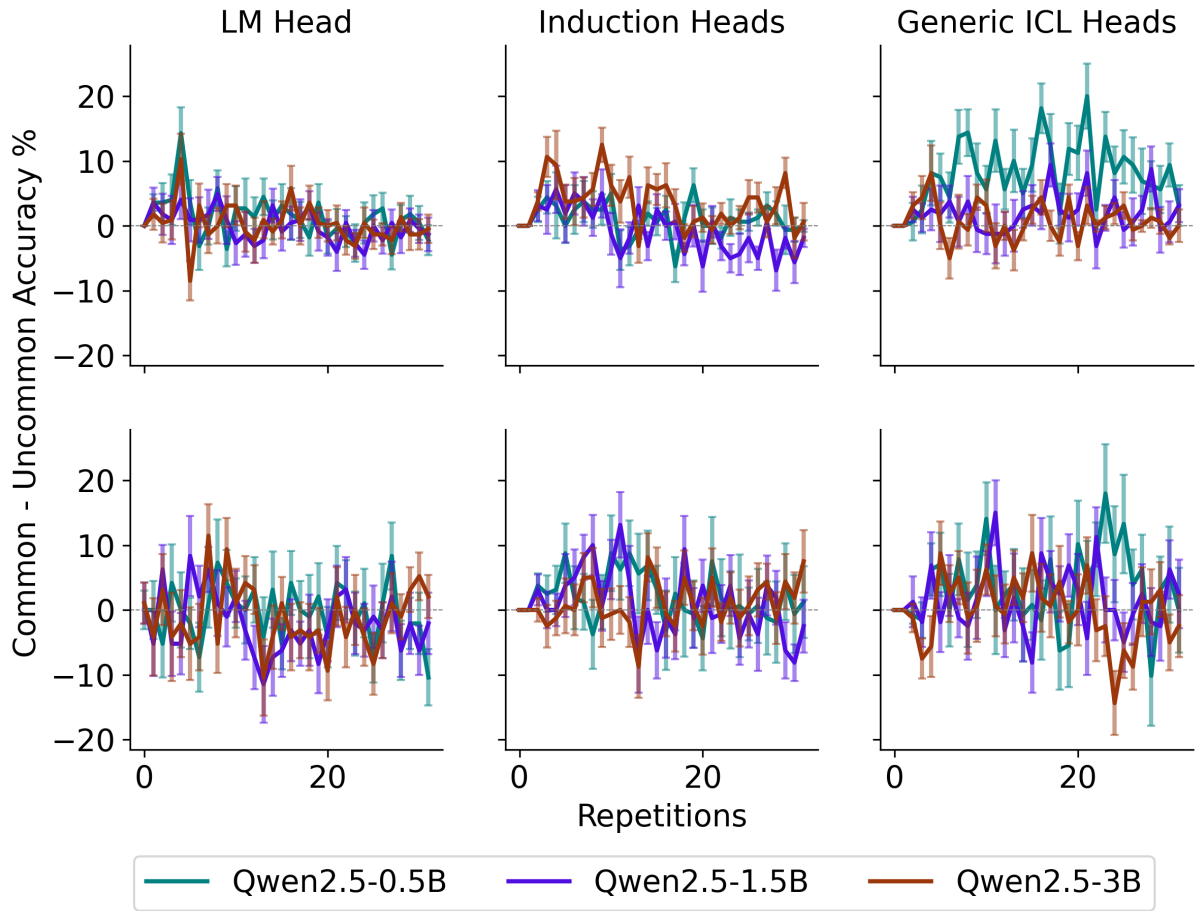


Figure 14: The accuracy differences between the high frequency and the low frequency token sequences are small.

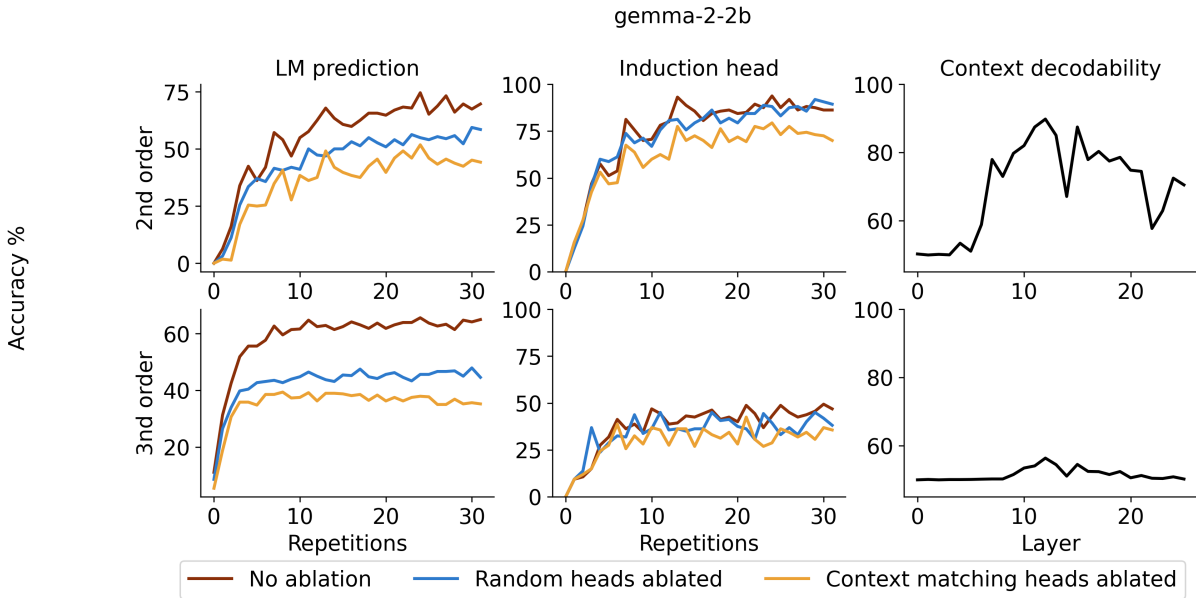


Figure 15: Experimental evaluation on Gemma2-2B.

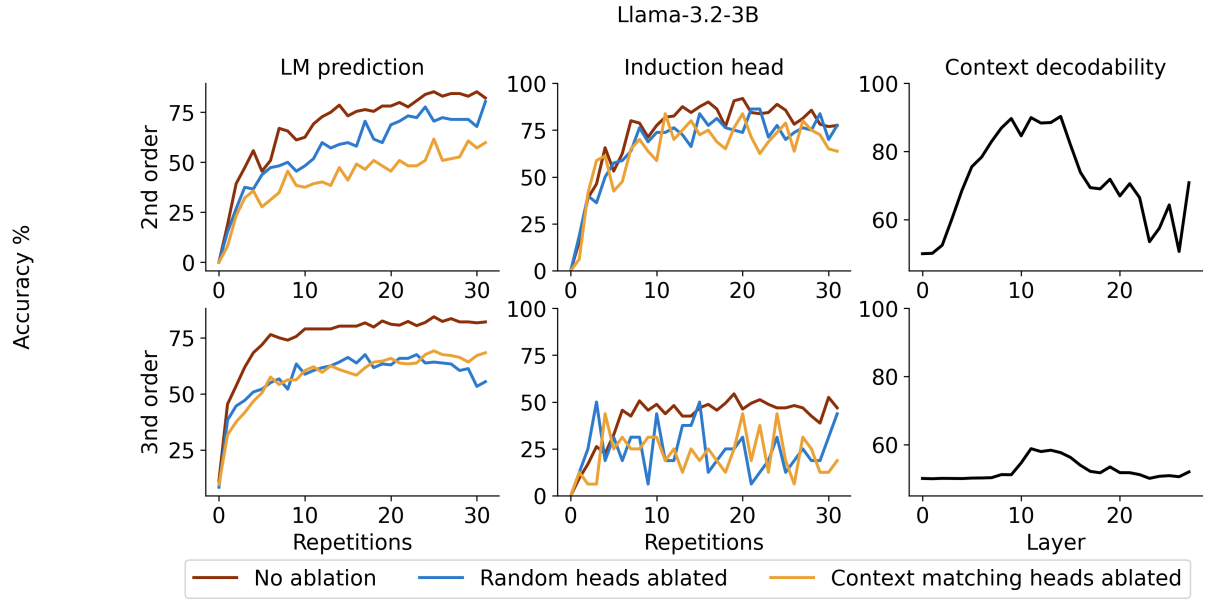


Figure 16: Experimental evaluation on Llama3.2-3B.

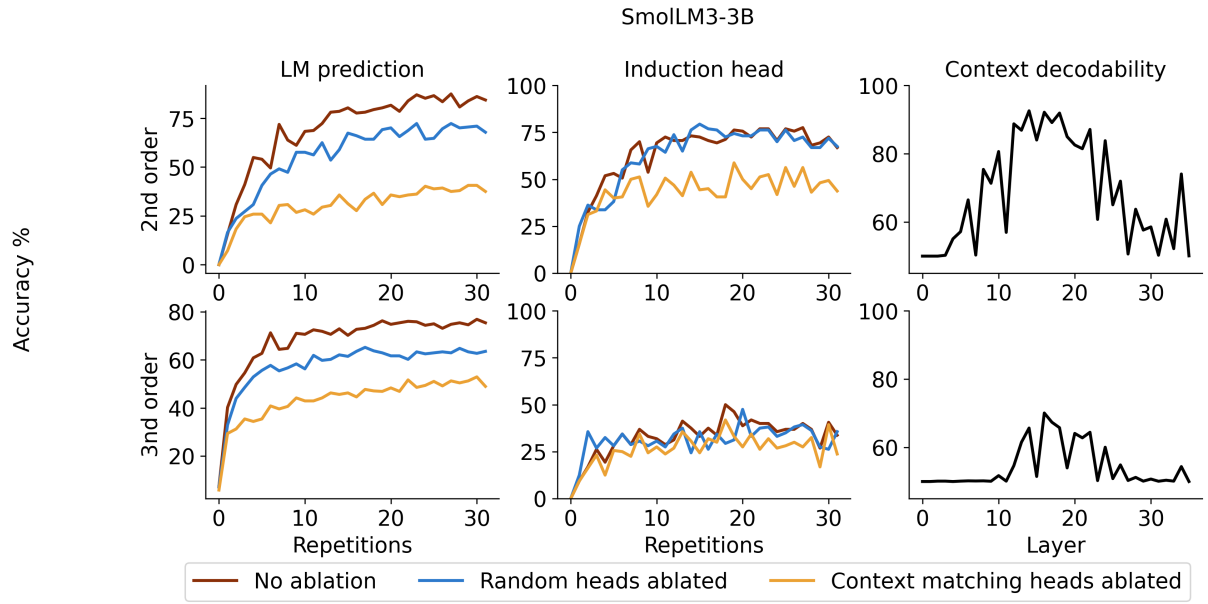


Figure 17: Experimental evaluation on SmolLM3-3B.

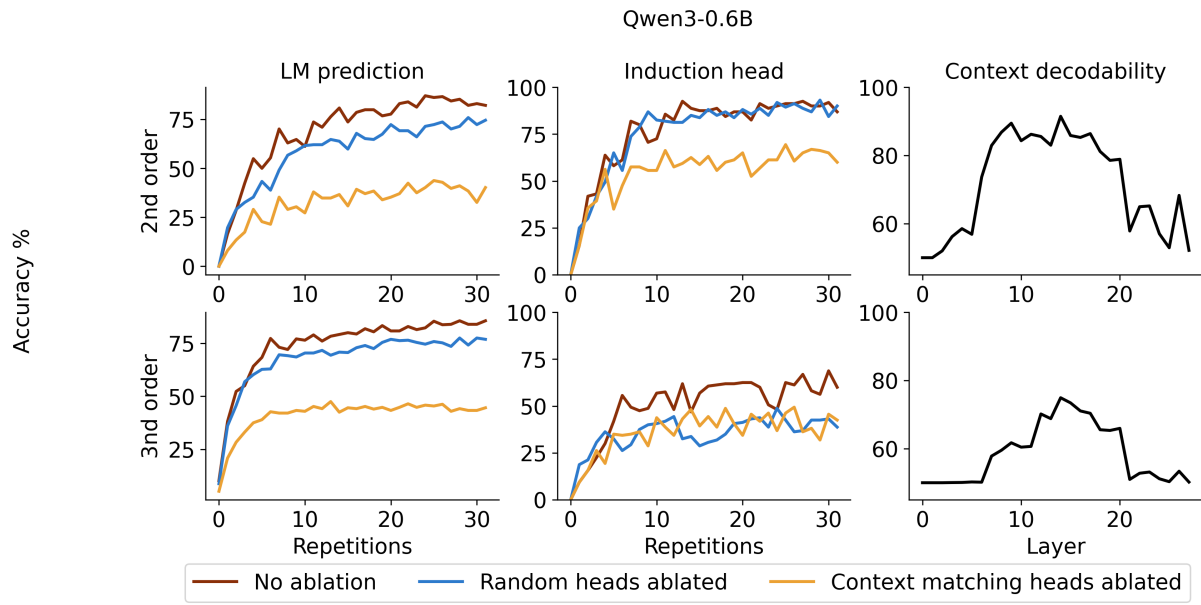


Figure 18: Experimental evaluation on Qwen3-0.6B.