

In-Context Learning Beyond Copying: A Training-Time Analysis of Abstractive ICL

Kerem Şahin^{1,*}, Sheridan Feucht¹, Adam Belfki¹, Jannik Brinkmann²,
Aaron Mueller³, David Bau¹, Chris Wendler¹

¹Northeastern University, ²University of Mannheim, ³Boston University

Abstract

The sudden emergence of induction circuits and the ability to perform in-context copying in LLMs coincides with a sharp drop in training loss over a short span of the training process. This co-occurrence has been cited as evidence that it marks the point at which a wide range of in-context learning (ICL) capabilities begin to emerge. To analyze the training dynamics of few-shot abstractive ICL capabilities (i.e., tasks where the answer is not contained in the input context), we propose HAPAX, a training regime that omits the loss contribution of tokens predictable by the induction circuit to control training pressure for copying. Using this intervention, we find that abstractive ICL capabilities can still emerge, despite a substantial reduction in inductive copying and the absence of the characteristic phase changes observed during Vanilla training. Analysis of attention heads shows that HAPAX-trained models develop fewer and weaker induction heads. Together, these results suggest that the developmental link between emergent phenomena associated with induction circuits and the emergence of abstractive ICL is weaker than previously hypothesized. More broadly, HAPAX provides a controlled framework for analyzing how copying-related training signals shape the development of in-context learning.

1 Introduction

Language modeling is fundamentally repetitive: in a coherent document, many sequences appear more than once, like “the Dursleys” in Harry Potter, or “public static void” in Java code. Words that *do* appear only once in a given text are so rare that corpus linguists give them a special name, *hapax legomena*, transliterated from Ancient Greek for “said once”. Prior work has identified *induction heads* (Elhage et al., 2021), attention heads

*Correspondence to sahin.ke@northeastern.edu. Code, model checkpoints, and evaluation data can be found at <https://hapax.baulab.info>

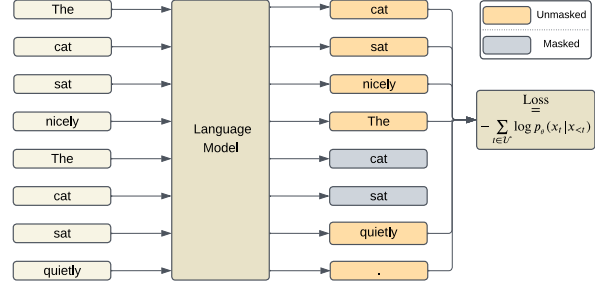


Figure 1: An overview of the HAPAX training regime. To suppress inductive copying, we introduce HAPAX where positions predictable by induction heads within a context window do not contribute to the loss (gray positions).

that exploit this repetition by matching patterns from earlier context and copying their continuations. Olsson et al. (2022) observed that the emergence of induction circuits, where many attention heads rapidly attain high prefix-matching scores and inductive copying capabilities within a narrow training window, coincides with an increase in ICL performance. They hypothesized that this phase change marks the emergence of a wide range of ICL capabilities, driven by the formation of induction circuits.

Subsequent work has found other attention heads that are causally more important for various ICL capabilities (Todd et al., 2024; Feucht et al., 2025; Yin and Steinhardt, 2025). Yin and Steinhardt (2025) find that function vector heads are causally more important than induction heads for ICL task performance, but provide correlational evidence that the attention heads with high prefix-matching scores later transform into other ICL-related heads during training. In order to understand the implications of the phenomena observed with the emergence of induction circuits, a causal intervention on the training process is necessary.

A natural thought experiment to analyze the training dynamics is to ask what would happen

to ICL capabilities if every n -gram within a Large Language Model’s (LLM) context window were a previously unseen hapax legomenon, removing the very repetition that induction heads exploit.

Our contributions are as follows. (1) We introduce HAPAX, a novel training regime which provides a causal methodology for analyzing the emergent phenomena associated with induction circuits. (2) Using HAPAX, we provide an alternative view of ICL training dynamics in LLMs: despite the absence of the characteristic phase change and a substantial drop in inductive copying capacity, abstractive ICL capabilities are retained. (3) We show that the token-loss difference metric, previously used to track ICL emergence, primarily reflects gains from inductive copying, and propose computing it over non-extractive token positions to track abstractive capabilities. (4) Together, our results show that the few-shot abstractive ICL capabilities can develop without the canonical induction-circuit phase transition observed under standard training.

2 Preliminaries

Following Todd et al. (2024), we distinguish *extractive* ICL tasks, where the model must extract the answer from the input, (e.g., “foil car purple : foil, pen cloud window :” $\rightarrow$ “pen”), from *abstractive* tasks, where the model needs to generate new answers rather than copying (e.g., “Greece : Athens, China : Beijing, Egypt” $\rightarrow$ “Cairo”). Abstractive tasks cannot be solved by copying from context, making them a natural testbed for our interest in the emergence of non-copying based ICL tasks.

Induction circuits provide a mechanistic account of copying-based ICL behavior. Induction circuits consist of three steps: (1) *previous token heads* that allow each token to store which token came before it, (2a) *induction heads* that attend to the previous token information in earlier contexts, resulting in a “prefix-matching” attention pattern, and (2b) increasing the probability of the attended token in the output, which we refer to as inductive copying (Figure 2). Formally, given input tokens $(x_1, \dots, x_j)$, induction circuits operate by searching for tokens that hold information of the current token x_j (i.e., searching x_{i+1} where $x_i = x_j, i < j$) (Elhage et al., 2021). If there is a matching x_{i+1} , the induction head increases its logit for the next prediction.

To quantify prefix matching, we use the score from Nanda and Bloom (2022). On a sequence $x = (r_1 r_2 \dots r_s r_1 r_2 \dots r_{s-1})$ of randomly sam-

pled tokens repeated twice, the prefix-matching score for head (l, h) is

$$\text{PrefixMatching}(l, h) = \frac{1}{s-1} \sum_{i=1}^{s-1} A_{s+i, i+1}^{(l, h)} \quad (1)$$

where $A^{(l, h)}$ is the attention map of the attention head at layer l and index h .

3 HAPAX

3.1 Training Protocol

HAPAX masks the loss contribution of induction-predictable positions, i.e. positions where the correct token can be retrieved by copying the continuation of an earlier matching bigram, reducing the training pressure for inductive copying. Single-token repetitions are not masked because they cannot be predicted by induction.

Formally, let $\mathbf{x} = (x_t)_{t \in S}$ be an input sequence of tokens from vocabulary $\mathcal{V}$, where $S = \{1, 2, \dots, T\}$ denotes the set of all token positions and $x_t \in \mathcal{V}$ for all $t \in S$. The probability assigned by the model to the correct prediction at a position t is defined as:

$$p_\theta(x_t | x_{<t}) = \text{softmax}(f_\theta(x_{<t}))_{x_t} \quad (2)$$

where f_θ denotes the model with parameters θ . We define the set of masked positions as:

$$\mathcal{M} = \{j \in \mathcal{S} : \exists 1 < i < j \text{ such that } (x_{i-1}, x_i) = (x_{j-1}, x_j)\}. \quad (3)$$

The set of unmasked positions is then $\mathcal{U} = \mathcal{S} \setminus \mathcal{M}$. We compute the HAPAX loss, a negative log-likelihood, only over the unmasked positions:

$$\mathcal{L}_{\text{Hapax}} = -\frac{1}{|\mathcal{U}|} \sum_{t \in \mathcal{U}} \log p_\theta(x_t | x_{<t}) \quad (4)$$

We train Vanilla and HAPAX 1B models from scratch using the transformers library (Wolf et al., 2020) implementation of GPT-NeoX (Andonian et al., 2023). We use the same hyperparameter and training configuration as the Pythia models (Biderman et al., 2023). We use the Pile dataset to train our models (Gao et al., 2020). We train our models for 19900 steps. Using the estimates of Hoffmann et al. (2022), a 1B-parameter model reaches the

¹Results reported in the text include only the subset of tasks with statistically significant differences ($p < 0.05$, McNemar’s test) unless stated otherwise.

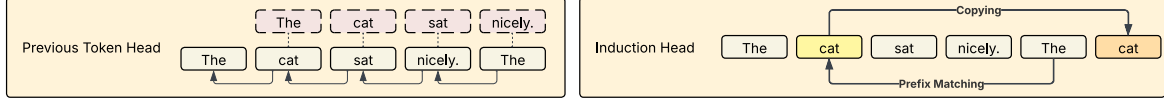


Figure 2: Demonstration of the induction circuit. Previous token heads allow each token to store which token came previously. Induction heads do a match-and-copy operation to reproduce the subsequence that appeared earlier in the context.

compute-optimal regime after training on approximately 20.2B tokens. Our main 1B Vanilla and HAPAX models are trained beyond this threshold.

Checkpoints are saved every 100 steps, and evaluations are computed from step 100 through step 19900. The training data consists of 41.7B tokens, of which 13.2B (31.7%) tokens are masked for the HAPAX model due to loss masking. Our main results are based on a single random seed for each training condition, following the Pythia training configuration and its default seed (1234). We also train two supplemental runs (seeds 1337 and 2807) under a different precision and GPU ecosystem, whose details can be found in Appendix G.

Reported confidence intervals are computed using a normal approximation over evaluation examples. Further training variants that we explored but found ineffective are provided in Appendix E to inform future work on training dynamics of LLMs.

3.2 Similarity-Thresholded HAPAX

Certain token pairs can have a high cosine similarity in the input embedding space. As an example, the tokens “National” and “National” (with a leading space) have a cosine similarity of 0.84 in the input embedding space of the vanilla model. Although they are not exactly the same token, the model can still acquire a high copying signal through these similar tokens. Referring back to Equation 3, we modify the exact matching with a thresholded matching logic to create a stricter suppression. We mask a position if the corresponding tokens have a cosine similarity higher than τ :

$$\mathcal{M} = \{j \in \mathcal{S} : \exists i < j \text{ s.t. } S_{\cos}(e_{x_{i-1}}, e_{x_{j-1}}) > \tau \text{ and } S_{\cos}(e_{x_i}, e_{x_j}) > \tau\}. \quad (5)$$

where e_{x_i} refers to the embedding for the token x_i . We choose $\tau = 0.3$. Consequently, 21.8B (52.5%) out of 41.7B tokens are masked for the Thresholded-HAPAX model. This variant allows us to test whether the copying-related training signals

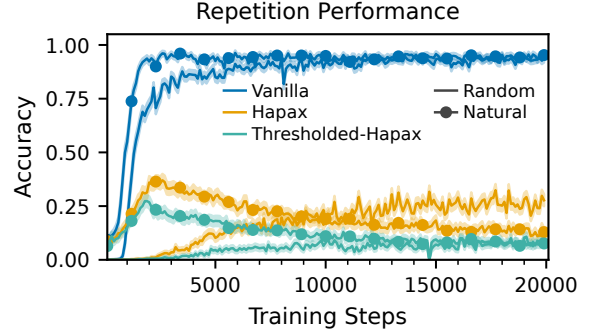


Figure 3: Repetition performance in both random token and natural text settings. HAPAX models struggle with repetition, which is solvable with induction circuits. This shows the HAPAX training can control copying behavior. Accuracy is measured over 1000 randomly generated samples with $s = 25$.

are effectively reduced in HAPAX. However, because this criterion masks substantially more token positions than exact-match HAPAX, it also risks removing useful non-copying training signal; we therefore treat it as a more aggressive stress test. See Appendix B for details.

4 Effects of HAPAX on ICL

In the following sections we verify that HAPAX suppresses inductive copying but abstractive ICL capabilities emerge despite the substantial reduction in inductive copying. We further analyze loss values throughout training and find that abstractive capabilities emerge even without the characteristic phase shift in loss observed for the vanilla model. Comparisons reported in the text include the tasks with statistically significant differences ($p < 0.05$, McNemar’s test; see Appendix F for our test choice).

4.1 Suppression of Inductive Copying

We evaluate inductive copying on a random repetition task where the model is given 1000 sequences of random repeated tokens $r_1 r_2 \dots r_s r_1 r_2 \dots r_{s-1}$ and is expected to predict r_s . This synthetic task

does not occur in natural language but is solvable through the induction circuit. Figure 3 demonstrates that HAPAX causes a major drop in random repetition performance. The HAPAX model experiences a 66% drop and Thresholded-HAPAX experiences an 89% drop in accuracy relative to the vanilla model at the end of training. We also evaluate performance on natural text repetition in Figure 3, where the models are given 1000 sequences of the form $r_1 r_2 \dots r_s r_1 r_2 \dots r_{s-1}$ with $r_1 r_2 \dots r_s$ taken from whole sequences in the Wiki-Text dataset (Merity et al., 2017). Natural text repetition accuracy decreases over time, suggesting the model’s increased incentive to not repeat natural text. The initial increase of natural text repetition is due to the model’s increased language modeling capacity, as a non-repeated sequence $r_1 r_2 \dots r_{s-1}$ has the same initial accuracy for the HAPAX model (Figure A11) as the maximum attained natural text repetition accuracy. For extractive tasks, HAPAX and Thresholded-HAPAX show reduced performance compared to vanilla on all but one task with significant differences (Tables A3, A5). We also observe this reduction in copying behavior in model generations. See Figure C25 for examples and Appendix C for fluency evaluations.

Lastly, to test whether the reduction in copying stems from the specific positions we select or simply from the number of tokens masked, we add a Random-Masked baseline that masks the same number of positions per context but chooses them at random, using the same configuration and seed, for the first 5000 steps. Random repetition accuracy at step 5000 is 93.7% for Vanilla and 93.5% for the Random-Masked baseline, but 13.3% for HAPAX. The reduction in copying therefore cannot be explained by the number of tokens masked alone, and the positions selected by HAPAX have direct influence on the copying signal.

These results provide evidence that the HAPAX training regime can control inductive copying.

4.2 Abstractive ICL Capabilities

Abstractive tasks require generating information not contained in the context. We evaluate our models on 34 abstractive tasks, where 26 are from Todd et al. (2024) and 8 are word-level translation tasks from the Open Multilingual Wordnet, accessed via NLTK (Bird and Loper, 2004) (Appendix D.2). We organize tasks into categories following the categorization of Hendel et al. (2023) (Table A8).

We evaluate each model under two settings. In

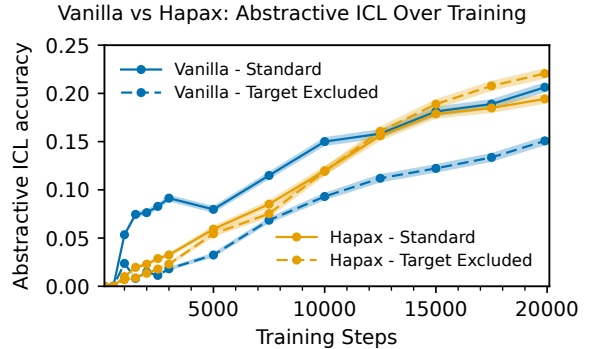


Figure 4: The mean absolute gap between the Standard and Target Excluded evaluation is 4.7% for Vanilla versus 0.8% for HAPAX over training, demonstrating that HAPAX is less affected by induction-driven label copying. The target-excluded curves of both models emerge at similar times, indicating that HAPAX does not delay the onset of abstractive ICL learning despite the reduction of copying.

the *standard* setting, we use the original 5-shot prompts. Some tasks from Todd et al. (2024) have small label spaces, so the target prediction frequently appears in the few-shot demonstrations as well, and a model can succeed by matching the context’s label distribution rather than by abstractive generalization. To control for this, we also introduce a *target-excluded* setting that samples few-shot examples from data points that do not have the same target prediction (see Appendix D.3).

Table 1 summarizes abstractive ICL across 34 tasks. Macro-averaged accuracy is 19.5% for HAPAX and 20.6% for Vanilla under standard evaluation, and 22.1% and 15.1% respectively under target exclusion. At the task level, 22 tasks show a statistically significant difference under standard evaluation (13 favoring HAPAX, 9 favoring Vanilla) and 28 under target exclusion (26 favoring HAPAX, 2 favoring Vanilla). This indicates that part of Vanilla’s standard-evaluation accuracy comes from copying target predictions present in the context, and that abstractive ICL capabilities can emerge under HAPAX despite the suppression of inductive copying. Consistent with prior work (Todd et al., 2024; Feucht et al., 2025; Yin and Steinhardt, 2025), we verify in Appendix A.5 that function-vector and concept heads still develop under Hapax and function-vector heads remain causally important for abstractive ICL.

For Thresholded-HAPAX, 24 tasks show a statistically significant difference under standard evaluation (7 favoring Thresholded-HAPAX, 17 favoring

Table 1: Abstractive task performance summary at step 19900 over 34 tasks. **Sig.** gives the number of tasks with significant accuracy differences under McNemar tests with joint Benjamini–Hochberg correction (FDR level 0.05). The **> Vanilla** column gives the number and fraction of those significant tasks where the corresponding variant has higher accuracy than vanilla. *Standard* does not exclude the target label in few-shot demonstrations; *Target excl.* excludes the target label. Individual task names, accuracies and task category explanations are given in Appendix A.1.

Task Category	Vanilla vs. HAPAX			
	Vanilla	HAPAX	Sig.	> Vanilla
<i>Algorithmic</i> ($n = 8$)				
Standard	28.6 ± 0.9	27.3 ± 0.8	5	1/5 (20%)
Target excl.	25.7 ± 0.8	31.7 ± 0.9	8	7/8 (88%)
<i>Linguistic</i> ($n = 4$)				
Standard	30.3 ± 2.2	39.8 ± 1.9	4	4/4 (100%)
Target excl.	30.3 ± 2.2	39.9 ± 1.9	4	4/4 (100%)
<i>Knowledge</i> ($n = 11$)				
Standard	18.1 ± 1.2	18.8 ± 1.2	6	4/6 (67%)
Target excl.	12.4 ± 1.1	21.4 ± 1.3	9	9/9 (100%)
<i>Translation</i> ($n = 8$)				
Standard	4.3 ± 0.4	6.1 ± 0.5	4	4/4 (100%)
Target excl.	4.3 ± 0.4	6.1 ± 0.5	4	4/4 (100%)
<i>Sentence-level</i> ($n = 3$)				
Standard	39.2 ± 1.1	9.4 ± 0.8	3	0/3 (0%)
Target excl.	4.7 ± 0.4	18.1 ± 1.0	3	2/3 (67%)
All ($n = 34$)				
Standard	20.6 ± 0.5	19.5 ± 0.5	22	13/22 (59%)
Target excl.	15.1 ± 0.5	22.1 ± 0.5	28	26/28 (93%)

Vanilla) and 26 under target exclusion (16 favoring Thresholded-HAPAX, 10 favoring Vanilla) (Appendix A.1). Thresholded-HAPAX applies heavier masking and produces lower standard-evaluation scores than HAPAX and Vanilla. However, under target exclusion, it achieves higher accuracy than the Vanilla model on 62% of significant tasks, which suggests that abstractive capabilities can emerge despite a stronger reduction in induction capability.

In Appendix H we report an additional 2.8B comparison between Vanilla and HAPAX with similar trends. Overall, these results allow us to examine how abstractive ICL changes when inductive copying is suppressed. Rather than observing broad degradation of ICL, we find that abstractive capabilities are retained, and Figure 4 shows that Hapax and Vanilla acquire abstractive ICL at similar points in training under target-excluded evaluation.

5 Loss Dynamics Beyond N-gram Copying

Sections 4.1 and 4.2 show that abstractive ICL capabilities can emerge under HAPAX despite a substantial reduction in inductive copying. This raises the question of whether the loss-based metrics previously used to track ICL emergence reflect this.

We use the token-loss difference metric, which is defined by the differences of the cross-entropy loss of arbitrary token positions, conventionally using the 500th and 50th token positions. Formally, let $L_t = -\log p_\theta(x_t | x_{<t})$ denote the cross-entropy loss at token position t . The token-loss difference (TLD) between the 50th and 500th tokens is then given by $\Delta TLD = L_{50} - L_{500}$. Intuitively, token-loss difference measures improvement across increasing token positions. If the loss at token 500 is lower and $TLD > 0$, it shows that the model’s predictions improved with increasing context.

Yin and Steinhardt (2025) provided evidence that at the final checkpoint of a model, the metric is strongly influenced by induction heads but does not correlate well with in-context learning task performance. Figure 5 (All Samples) shows that the Vanilla model exhibits a sudden improvement in the metric, while HAPAX and Thresholded-HAPAX appear to worsen over training. As HAPAX models perform worse at exact copying tasks but maintain their capabilities on abstractive ICL tasks, our results demonstrate that the sudden increase in the metric for the Vanilla model is indicative of the emergence of inductive copying capabilities (consistent with observations from Lv et al. (2025)), but lacks indicative power for the emergence of abstractive ICL capabilities. As HAPAX masks 31.7% of tokens, a substantial fraction of positions in the data are ones where copying-based mechanisms are useful, which explains why the data distribution is dominated by repetition and why this results in a sudden loss drop as soon as induction is learned. Additionally, in Figure 9 at step 5000, the Random-Masked model reaches a validation loss of 2.70, against 2.67 for Vanilla and 3.76 for HAPAX. The gap between Vanilla and HAPAX therefore cannot be attributed solely to the reduction in supervised positions.

To investigate this hypothesis further, we propose evaluating TLD on samples where neither the 500th nor the 50th token can be predicted correctly through inductive copying, alongside a Thresholded variant that also excludes positions pre-

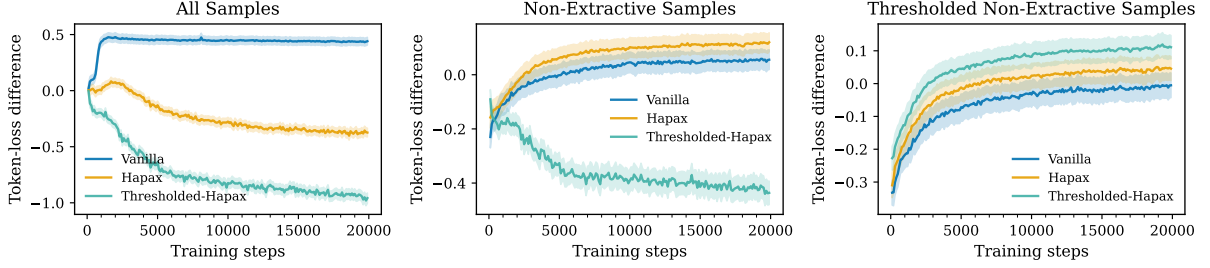


Figure 5: Comparison of token-loss metrics across training. Unlike the regular (All Samples) TLD, each HAPAX variant is able to achieve non-trivial ICL capabilities on its corresponding distribution. Since the regular distribution is dominated by induction-predictable positions, its gains primarily reflect exact copying rather than abstractive ICL. Computing the metric on targeted subsets thus reveals distinct emergence patterns across the distributions of interest. Metrics are computed on randomly sampled validation data.

dictable through highly similar tokens ($\tau = 0.3$).

Figure 5 shows that, contrary to the regular TLD metric, the Vanilla model does not exhibit a sudden phase shift on non-extractive samples and is statistically indifferent to HAPAX. The individual losses at the non-extractive positions (Figure A13) are lower and the TLD metric itself is higher in the HAPAX model compared to the Vanilla model, showing that despite the reduction in repetition and inductive copying, the TLD metric improves for non-extractive samples.

A similar pattern holds for Thresholded-HAPAX, where the TLD metric is worse on the first two distributions (since this variant also masks non-extractive but similar tokens) yet shows an increasing trend on the thresholded variant. Overall, our results show that the TLD metric, across all variants, captures the coarse trends of the distribution on which it is computed, so its implications for different types of ICL capabilities depend on the choice of distribution. The regular TLD metric is dominated by induction-predictable positions, and evaluating it on different sample distributions reveals different insights.

6 Development of Prefix-Matching Under HAPAX

Within Vanilla training, the sudden emergence of induction heads with high prefix-matching scores is a well-documented phenomenon (Olsson et al., 2022). One possibility is that this phase transition is a developmental prerequisite for abstractive ICL, where the sharp emergence of prefix-matching is a necessary transition only after which abstractive capabilities can be learned. Such developmental prerequisites have been documented elsewhere; for example, Chen et al. (2024a) analyzed the sud-

den emergence of Syntactic Attention Structure in MLMs, and its necessity for the later acquisition of grammatical capabilities.

We analyze the trajectory of prefix-matching scores in the Vanilla and HAPAX models. In Figure 6, the Vanilla model contains 12 out of 128 attention heads that attain a prefix-matching score over 80%, (See Section 2 for formula) spike early in training and then decay, while the HAPAX variants display only a single head that attains above 80% prefix-matching score. This emergence unfolds gradually over 1800 steps in HAPAX, in contrast to the sudden emergence in Vanilla.

Beyond their reduced number and strength, many remaining attention heads with high prefix-matching scores in HAPAX also depart from the canonical induction-head behavior. We apply logit lens (Nostalgebraist, 2020) to the top 10 attention heads with the highest prefix-matching score by taking the corresponding slice of the output matrix W_o for each attention head’s output, then applying the final layer norm and unembedding matrix W_u to approximate the distribution over tokens promoted by that head.

As shown in Figure 7, inclusion of the token to be copied in the top- k promoted tokens worsens from Vanilla to HAPAX, indicating that reduced copying behavior also holds at the attention-head level. This also holds under mean-ablation studies measuring each head’s contribution to predicting the target token (Appendix A.2). Among the top 10 attention heads with the highest prefix-matching score in HAPAX, 6 suppress rather than promote the correct token (compared to 2 in Vanilla), functionally behaving closer to anti-induction heads (McDougall et al., 2024; Olsson et al., 2022) than to regular induction heads (Appendix A.2).

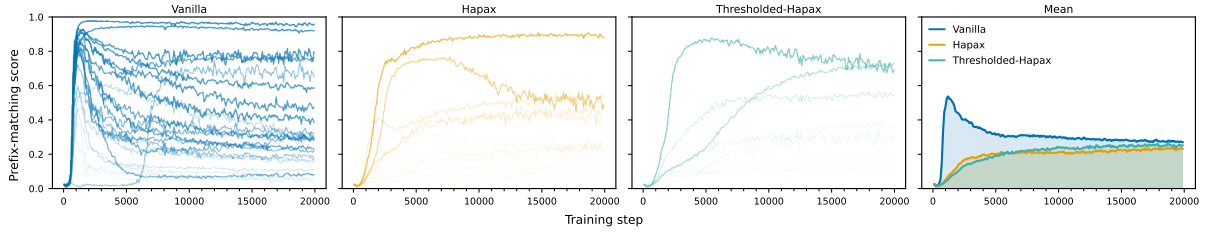


Figure 6: Prefix-matching scores of attention heads across training. Each line is one head. Opacity is proportional to peak score attained throughout training (heads with peak < 0.1 omitted). Many heads in the Vanilla model peak suddenly and then decay (Mean plot), with 12 of 128 reaching $> 80\%$ prefix-matching score. HAPAX variants do not display such a transition and only 1 head in HAPAX ever exceeds 80% .

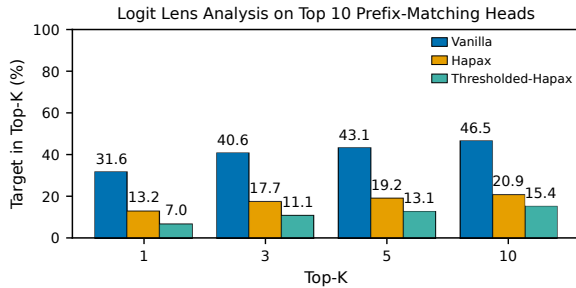


Figure 7: Logit lens on top-10 attention heads with the highest prefix-matching scores on the random repetition task (Section 4.1). For each head, we measure whether the target token r_s appears among its top- k promoted tokens. Bars show the percentage averaged over the top-10 attention heads with the highest prefix-matching scores. Although these heads attend to r_s , HAPAX variants increasingly fail to promote it at the output. See Figure A14 for per-head contributions.

Taken together, these findings suggest that abstractive ICL can emerge without the abrupt prefix-matching phase transition observed during Vanilla training.

We additionally give supplemental runs with different seeds which differ from the reference in precision (bf16 vs fp16) and GPU ecosystem (ROCm vs CUDA). We observe similar qualitative trends in the emergence of abstractive ICL, prefix-matching scores, and TLD. See Appendix G for details.

7 Cross-Checkpoint Patching Reveals Initialization-Level Prefix-Matching Structure

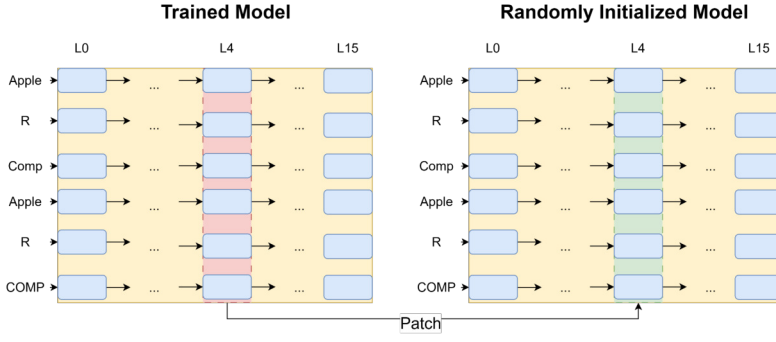
Under HAPAX, the baseline phase transitions do not appear, and prefix-matching scores are attenuated. However, prefix matching is not eliminated entirely. We explore how these patterns can persist when the training objective removes the loss contribution of induction-predictable positions. One possibility involves previous-token heads, which can

be learned for reasons beyond in-context copying, such as detokenization (Lad et al., 2025; Gurnee et al., 2023; Feucht et al., 2024; Kaplan et al., 2025). If previous-token information is useful for reasons other than in-context copying, then later heads may still exploit this information and produce prefix-matching patterns even without explicit optimization for in-context copying. To test this, we compare the Vanilla model at step 5000 with the same model at random initialization. At step 5000, layer L4 contains previous-token heads, but no strong prefix-matching heads appear in L4 or earlier layers (Figure A20). We patch the layer-L4 output from step 5000 into the randomly initialized model and measure whether later randomly initialized heads begin to exhibit prefix matching. Figure 8 shows that adding previous-token information increases the maximum prefix-matching score from 0.076 to 0.122¹. The highlighted heads also rank among the top five prefix-matching heads at the final Vanilla checkpoint (Figure A18), suggesting that a degree of prefix-matching tendency may already be present at initialization. We observe a similar pattern in HAPAX: two of its top three final prefix-matching heads also increase their prefix-matching scores in this patching experiment. Since Vanilla and HAPAX share the same initialization seed, this overlap suggests that the remaining prefix matching scores can partly reflect initialization-level attention tendency.

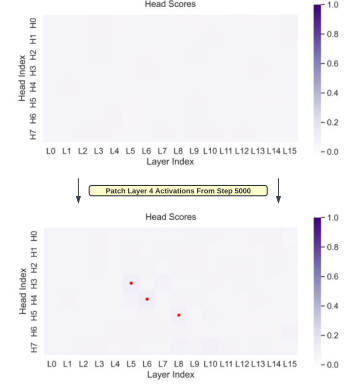
8 Related Work

ICL and Attention Heads Despite being trained to do simple next-token prediction, LLMs exhibit the capability to do in-context learning (ICL) (Brown

¹Highlighted heads are statistically significant under a paired t -test comparing each head’s prefix-matching score to the mean score of all other heads across the same 1000 samples.



(a) Visualization of the patching setting.



(b) Prefix-Matching scores before and after the patching.

Figure 8: (a) Demonstration of the patching setup. (b) Induction scores for the randomly initialized Vanilla model before and after patching layer L4 from the step-5000 Vanilla model. At this checkpoint, L4 is the earliest layer with a strong previous-token head (Figure A20). Patching L4 increases the maximum prefix-matching score from 0.076 to 0.122. Red dots mark the top three prefix-matching heads after patching, which also rank highly at the final checkpoint.

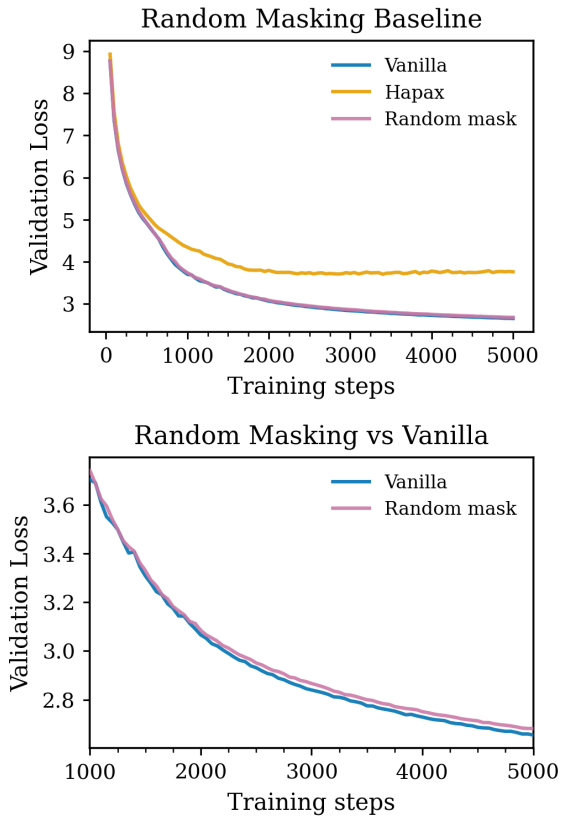


Figure 9: Validation loss over the first 5000 steps for the Vanilla, HAPAX, and Random-Masked models. Random masking removes the same number of positions per context as HAPAX (31.7% of positions), but selects them at random. At step 5000 the Random-Masked model reaches a validation loss of 2.70, against 2.67 for Vanilla and 3.76 for HAPAX. The gap between Vanilla and HAPAX therefore cannot be attributed solely to the reduction in supervised positions.

et al., 2020; Chowdhery et al., 2023), where they can perform tasks demonstrated in-context “on-the-fly” without additional training. Early work identified induction circuits responsible for inductive copying (Elhage et al., 2021) and analyzed their influence on ICL (Olsson et al., 2022; Crosbie and Shutova, 2025), hypothesizing they are the basis of in-context learning. Subsequent studies found various attention heads important for ICL, including function vector heads that trigger tasks (Todd et al., 2024), concept induction heads that copy lexical units (Feucht et al., 2025), semantic induction heads extracting relations (Ren et al., 2024), symbolic induction heads inducing over abstract variables (Yang et al., 2025), and n-gram generalizations of induction heads (Akyürek et al., 2024). Other work has studied how data structure shapes these phenomena, showing that parallel structure in pretraining data can support ICL (Chen et al., 2024c). Our work complements prior studies by examining the training dynamics underlying abstractive ICL when the training pressure for inductive copying is reduced, with the goal of better isolating the role of emergent induction-circuit phenomena.

Training Dynamics and Induction Heads.

In concurrent work, Wang and Murfet (2026) introduce patterning, a susceptibility-based framework that derives data interventions to steer training toward a target internal structure. They show in a 3M-parameter language model that targeted re-weighting of tokens can accelerate, delay, or prevent induction formation. They study how to

control circuit formation. We instead use HAPAX to suppress copying in order to study abstractive ICL’s emergence relative to the induction-circuit phase transition at 1B and 2.8B scale.

Theoretic work on synthetic setups has provided insights into induction head development and training dynamics. Several studies analyze phase transitions where [Singh et al. \(2024\)](#) use clamping to study subcomponents of induction heads and their effect on phase changes. [Chan et al. \(2022\)](#) shows how skewed Zipfian distributions lead to the emergence of ICL. [Bietti et al. \(2023\)](#) demonstrates the evolution from global bigram statistics to induction head solutions over training, and [Chen et al. \(2024b\)](#); [Edelman et al. \(2024\)](#) further analyze convergence to induction-like solutions in Markov chain data with [Edelman et al. \(2024\)](#) demonstrating the different phases where the model learns more complex strategies with more training. [Minegishi et al. \(2025\)](#) studies a meta-learning setting to investigate mechanisms in non-exact copying scenarios. [Singh et al. \(2025\)](#) show that induction heads can be reused across competing strategies over training. In our work, we instead focus on a natural language setting and create a data distribution where induction cannot predict the correct next token to suppress inductive copying.

Loss Masking. Prior work incorporated loss masking strategies to prevent memorization for information of interest ([Hans et al., 2024](#); [Kosireddy and Lucas, 2025](#)). We incorporate a loss masking strategy to create a distribution which allows us to suppress induction behavior throughout training.

Repetition. Prior work studied repetition from the perspective of data distribution and model outputs. [Zucchet et al. \(2025\)](#) analyzes how training data repetition speeds up emergent behavior in language models. [Aoyama et al. \(2026\)](#) characterizes statistical conditions under which induction heads emerge. Several works analyze the “repetition curse” observed within models ([Hiraoka and Inui, 2025](#); [Yao et al., 2025](#)). [Welleck et al. \(2020\)](#) presents unlikelihood loss training to mitigate repetitions in the model. As repetition is closely tied to induction heads, HAPAX discourages repeated sequences, aiming to suppress inductive copying and, as a result, repetition.

9 Conclusion

Our results suggest that the phase transition observed in standard training is closely tied to the

learning of repeated, induction-predictable structure in the data. When this copying signal is suppressed through Hapax, abstractive ICL capabilities still develop and are retained, despite the absence of the canonical phase transition. More broadly, our approach illustrates how targeted loss masking can disentangle training dynamics that emerge together under standard training, letting us test whether a phenomenon is tied to the capability being studied or to other predictable structure in the data.

Limitations

HAPAX tests whether the abstractive ICL capabilities studied here can emerge when copying-related training signal is reduced. The results characterize this specific question rather than final behavior of the model across all model sizes, training budgets, and masking variants. Our main analysis is limited at the 1B setting with the official Pythia configuration, covering Vanilla, HAPAX, and Thresholded-HAPAX across training, with the 2.8B run as a scale check. We also report two seeds in Appendix G, though they differ from the main run in precision and GPU ecosystem. Scaling across larger models, longer training, more seeds, and alternative masking variants is an important future direction.

Models reach non-trivial few-shot accuracy on several tasks and remain low on others (e.g., Polish-English) (Table A2). The comparisons measure whether HAPAX retains performance comparable to Vanilla, rather than absolute capability on every task.

Results are scoped to the few-shot abstractive settings used here. These settings test whether certain forms of abstractive ICL can emerge despite the absence of the canonical induction-circuit phase transition and reduced copying behavior. Other forms of abstractive ICL which require larger models or more complex reasoning may exhibit different dynamics. Similarly, our experiments show that function-vector and concept heads emerge and remain causally important under HAPAX. Additional mechanisms may also be at play, which have not been analyzed within the scope of this work.

The Random-Masked baseline that separates the effect of masking induction predictable positions from that of reducing supervision is trained at the 1B setting for 5000 steps. This covers the window in which the emergence of inductive copying is observable, but the comparison is not carried through to the end of training due to computational limits.

Ethics Considerations

This paper aims to advance the foundational understanding of in-context learning mechanisms. While such research may influence future model development and deployment, we cannot meaningfully anticipate these downstream impacts within the scope of this work.

Acknowledgments

We thank Andy Ardit, Amir Zur, Arnab Sen Sharma, Constanza Fierro, and Nikhil Prakash for valuable feedback and discussions. AB and DB are supported by the NSF National Deep Inference Fabric Grant #2408455, and CW and DB are supported by NSF Research Grant #2403304. SF and DB and KŞ are funded by a grant from Coefficient Giving. KŞ is supported by the Fulbright Program, which is sponsored by the U.S. Department of State and Turkish Fulbright Commission. The contents of this work are solely the responsibility of the authors.

The authors acknowledge the MGHPCC AI Computing Resource (AICR), with support from the Commonwealth of Massachusetts and the AICR participating institutions, for providing AI and high-performance computing resources used in the results reported in this publication. The authors thank the Northeastern University Research Computing team for their assistance with using the AICR cluster.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

- Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. 2024. [In-context language learning: Architectures and algorithms](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 787–812. PMLR.
- Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, and 2 others. 2023. [GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch](#).
- Tatsuya Aoyama, Ethan Gotlieb Wilcox, and Nathan Schneider. 2026. [Predicting the emergence of induction heads in language model pretraining](#). *Preprint*, arXiv:2511.16893.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. 2023. [Birth of a transformer: A memory viewpoint](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Steven Bird and Edward Loper. 2004. [NLTK: The natural language toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 18878–18891. Curran Associates, Inc.
- Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2024a. [Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs](#). In *The Twelfth International Conference on Learning Representations*.
- Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. 2024b. [Unveiling induction heads: Provable training dynamics and feature learning in transformers](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2024c. [Parallel structures in pre-training data yield in-context learning](#). In *Proceedings of the 62nd Annual Meeting of the Association*

- for *Computational Linguistics (Volume 1: Long Papers)*, pages 8582–8592, Bangkok, Thailand. Association for Computational Linguistics.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. [Palm: Scaling language modeling with pathways](#). *Journal of Machine Learning Research*, 24(240):1–113.
- Joy Crosbie and Ekaterina Shutova. 2025. [Induction heads as an essential mechanism for pattern matching in in-context learning](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5049–5111, Albuquerque, New Mexico. Association for Computational Linguistics.
- Thomas G. Dietterich. 1998. [Approximate statistical tests for comparing supervised classification learning algorithms](#). *Neural Computation*, 10(7):1895–1923.
- Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. 2024. [The evolution of statistical induction heads: In-context learning markov chains](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- 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, David Atkinson, Byron C Wallace, and David Bau. 2024. [Token erasure as a footprint of implicit vocabulary items in LLMs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9727–9739, Miami, Florida, USA. Association for Computational Linguistics.
- Sheridan Feucht, Eric Todd, Byron C Wallace, and David Bau. 2025. [The dual-route model of induction](#). In *Second Conference on Language Modeling*.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. [The pile: An 800gb dataset of diverse text for language modeling](#). *Preprint*, arXiv:2101.00027.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. [Finding neurons in a haystack: Case studies with sparse probing](#). *Preprint*, arXiv:2305.01610.
- Abhimanyu Hans, John Kirchenbauer, Yuxin Wen, Neel Jain, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and Tom Goldstein. 2024. [Be like a goldfish, don’t memorize! mitigating memorization in generative LLMs](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. [In-context learning creates task vectors](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, Singapore. Association for Computational Linguistics.
- Tatsuya Hiraoka and Kentaro Inui. 2025. [Repetition neurons: How do language models produce repetitions?](#) In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 483–495, Albuquerque, New Mexico. Association for Computational Linguistics.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, and 3 others. 2022. [An empirical analysis of compute-optimal large language model training](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.
- Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2025. [From tokens to words: On the inner lexicon of llms](#). *Preprint*, arXiv:2410.05864.
- Tagore Rao Kosireddy and Evan Lucas. 2025. [Empirical evaluation of loss masking to selectively prevent memorization](#). In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pages 142–149, Vienna, Austria. Association for Computational Linguistics.
- Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. 2025. [The remarkable robustness of llms: Stages of inference?](#) *Preprint*, arXiv:2406.19384.
- Ang Lv, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Rui Yan. 2025. [Language models “grok” to copy](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 735–741, Albuquerque, New Mexico. Association for Computational Linguistics.
- Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. [Copy suppression: Comprehensively understanding a motif in language model attention heads](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363,

- Miami, Florida, US. Association for Computational Linguistics.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Gouki Minegishi, Hiroki Furuta, Shohei Taniguchi, Yusuke Iwasawa, and Yutaka Matsuo. 2025. [Beyond induction heads: In-context meta learning induces multi-phase circuit emergence](#). In *Forty-second International Conference on Machine Learning*.
- Neel Nanda and Joseph Bloom. 2022. Transformerlens. <https://github.com/TransformerLensOrg/TransformerLens>.
- Nostalgebraist. 2020. [Interpreting gpt: The logit lens](#). LessWrong.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html>.
- 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.
- Aaditya K Singh, Ted Moskovitz, Sara Dragutinović, Felix Hill, Stephanie C.Y. Chan, and Andrew M Saxe. 2025. [Strategy cooption explains the emergence and transience of in-context learning](#). In *Forty-second International Conference on Machine Learning*.
- Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie C.Y. Chan, and Andrew M Saxe. 2024. [What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 45637–45662. PMLR.
- Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. [Function vectors in large language models](#). In *The Twelfth International Conference on Learning Representations*.
- George Wang and Daniel Murfet. 2026. [Patterning: The dual of interpretability](#). Preprint, arXiv:2601.13548.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations*.
- Maurice Weber, Daniel Y Fu, Quentin Gregory Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Re, Irina Rish, and Ce Zhang. 2024. [Redpajama: an open dataset for training large language models](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *International Conference on Learning Representations*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.
- Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. [Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders](#). In *Forty-second International Conference on Machine Learning*.
- Yukang Yang, Declan Iain Campbell, Kaixuan Huang, Mengdi Wang, Jonathan D. Cohen, and Taylor Whittington Webb. 2025. [Emergent symbolic mechanisms support abstract reasoning in large language models](#). In *Forty-second International Conference on Machine Learning*.
- Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. 2025. [Understanding the repeat curse in large language models from a feature perspective](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7787–7815, Vienna, Austria. Association for Computational Linguistics.
- Kayo Yin and Jacob Steinhardt. 2025. [Which attention heads matter for in-context learning?](#) In *Forty-second International Conference on Machine Learning*.
- Nicolas Zucchet, Francesco D’Angelo, Andrew Kyle Lampinen, and Stephanie C.Y. Chan. 2025. [The emergence of sparse attention: impact of data distribution and benefits of repetition](#). In *The Thirty-ninth*

A Additional Model Results

A.1 ICL Task Results

Within this section, we provide the full table summaries of the tasks.

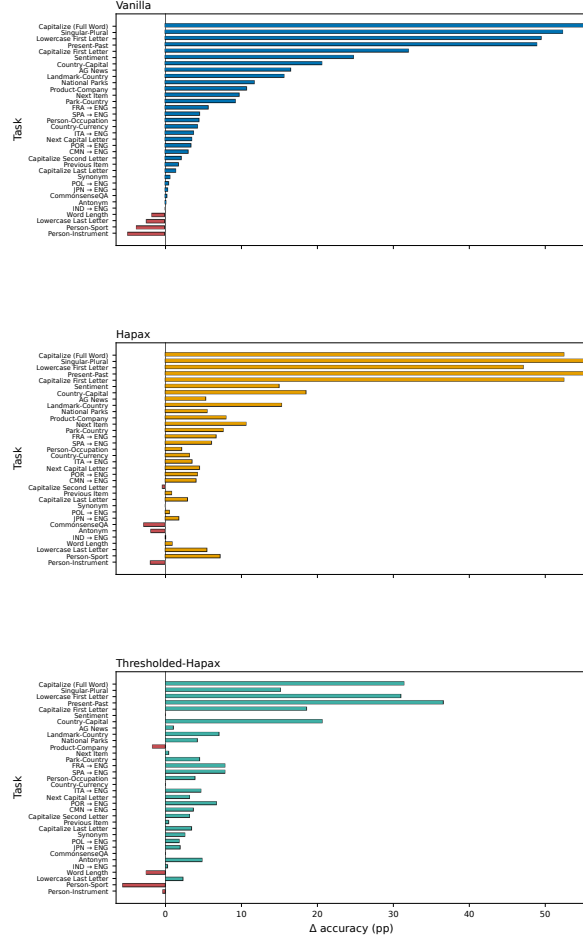


Figure A1: For Vanilla, HAPAX, and THRESHOLDED-HAPAX, increasing the number of demonstrations from 1-shot to 5-shot improves accuracy on the majority of tasks. This suggests that the models benefit from the in-context demonstrations, rather than relying solely on zero-shot or parametric task knowledge.

A.2 Ablation Study on Copying

We conduct ablation studies to understand the causal impact of each individual attention heads on the expected prediction r_s on a random repetition task $r_1 r_2 \dots r_{s-1}$. Mean ablation replaces the activation of a model component with its average activation from a reference distribution (Wang et al., 2023). For each head, we compute the mean activation over 10000 samples from the Pile dataset that the model was trained on (Gao et al., 2020). While

conducting mean ablations, we only ablate the last token position of the attention heads. We choose the last token to avoid disrupting the influence of the rest of the circuit, mainly the previous token heads. Given a sequence $r_1 r_2 \dots r_s r_1 r_2 \dots r_{s-1}$, previous token heads store information at position r_s (the final token of the first half) that induction heads use when predicting at position r_{s-1} in the second half. If we were to ablate all token positions, we would also disrupt the information flow from previous token heads, which would mistakenly lead us to identify such heads as induction heads. We measure the probability difference of the target token r_s before and after ablation, $p_{\text{clean}}(r_s | x) - p_{\text{ablated}}(r_s | x)$. We also include the ablation results of Vanilla and Thresholded-HAPAX for comparison with the HAPAX model. For interpreting the results of ablations, we chose to use probability difference metric as it correlated most accurately with task accuracy.

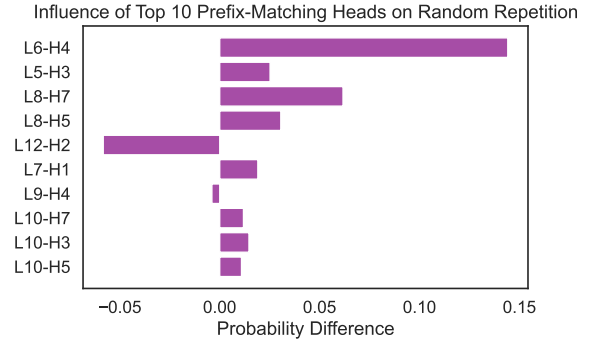


Figure A2: Probability Difference for random repetition task of the Top 10 Prefix-Matching heads for vanilla model.

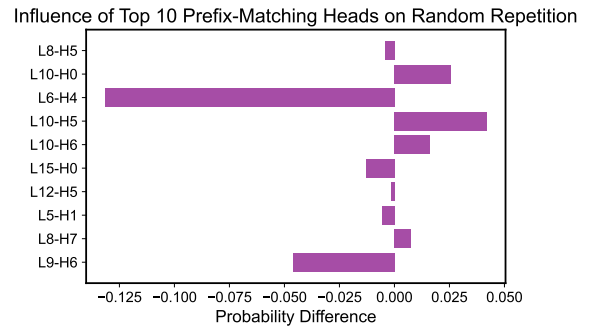


Figure A3: Probability Difference for random repetition task of the Top 10 Prefix-Matching heads for Hapax model.

Table A1: Abstractive task performance summary for the 1B model at step 19900 over 34 tasks. Macro-averaged accuracies (%) within each family are shown with 95% confidence intervals (normal approximation on the macro mean per category). **Sig.** gives the number of tasks with significant accuracy differences under McNemar tests with joint Benjamini–Hochberg correction ($\alpha = 0.05$). The **> Vanilla** column gives the number and fraction of those significant tasks where the corresponding variant has higher accuracy than vanilla. *Standard* includes the target label in demonstrations; *Target excl.* withholds the target label.

Task Category	HAPAX				Thresh-HAPAX			
	Van.	HAPAX	Sig.	> Van.	Van.	Thresh	Sig.	> Van.
<i>Algorithmic</i> ($n = 8$)								
Standard	28.6 $\pm$ 0.9	27.3 $\pm$ 0.8	5	1/5 (20%)	28.6 $\pm$ 0.9	15.3 $\pm$ 0.8	7	0/7 (0%)
Target excl.	25.7 $\pm$ 0.8	31.7 $\pm$ 0.9	8	7/8 (88%)	25.7 $\pm$ 0.8	19.1 $\pm$ 0.8	7	3/7 (43%)
<i>Linguistic</i> ($n = 4$)								
Standard	30.3 $\pm$ 2.2	39.8 $\pm$ 1.9	4	4/4 (100%)	30.3 $\pm$ 2.2	18.3 $\pm$ 2.0	4	2/4 (50%)
Target excl.	30.3 $\pm$ 2.2	39.9 $\pm$ 1.9	4	4/4 (100%)	30.3 $\pm$ 2.2	18.3 $\pm$ 2.0	4	2/4 (50%)
<i>Knowledge</i> ($n = 11$)								
Standard	18.1 $\pm$ 1.2	18.8 $\pm$ 1.2	6	4/6 (67%)	18.1 $\pm$ 1.2	10.0 $\pm$ 0.9	5	0/5 (0%)
Target excl.	12.4 $\pm$ 1.1	21.4 $\pm$ 1.3	9	9/9 (100%)	12.4 $\pm$ 1.1	12.9 $\pm$ 1.0	7	5/7 (71%)
<i>Translation</i> ($n = 8$)								
Standard	4.3 $\pm$ 0.4	6.1 $\pm$ 0.5	4	4/4 (100%)	4.3 $\pm$ 0.4	7.9 $\pm$ 0.6	5	5/5 (100%)
Target excl.	4.3 $\pm$ 0.4	6.1 $\pm$ 0.5	4	4/4 (100%)	4.3 $\pm$ 0.4	7.9 $\pm$ 0.6	5	5/5 (100%)
<i>Sentence-level</i> ($n = 3$)								
Standard	39.2 $\pm$ 1.1	9.4 $\pm$ 0.8	3	0/3 (0%)	39.2 $\pm$ 1.1	0.4 $\pm$ 0.1	3	0/3 (0%)
Target excl.	4.7 $\pm$ 0.4	18.1 $\pm$ 1.0	3	2/3 (67%)	4.7 $\pm$ 0.4	1.1 $\pm$ 0.2	3	1/3 (33%)
All ($n = 34$)								
Standard	20.6 $\pm$ 0.5	19.5 $\pm$ 0.5	22	13/22 (59%)	20.6 $\pm$ 0.5	10.9 $\pm$ 0.4	24	7/24 (29%)
Target excl.	15.1 $\pm$ 0.5	22.1 $\pm$ 0.5	28	26/28 (93%)	15.1 $\pm$ 0.5	12.8 $\pm$ 0.5	26	16/26 (62%)

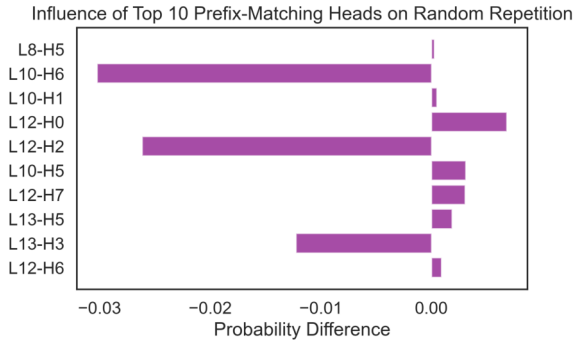


Figure A4: Probability Difference for random repetition task of the Top 10 Prefix-Matching heads for Thresholded-HAPAX.

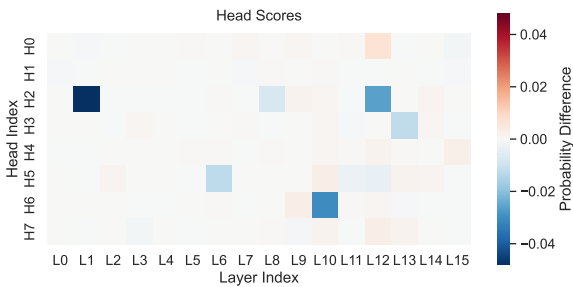


Figure A5: Probability difference values of random repetition task for Thresholded-HAPAX.

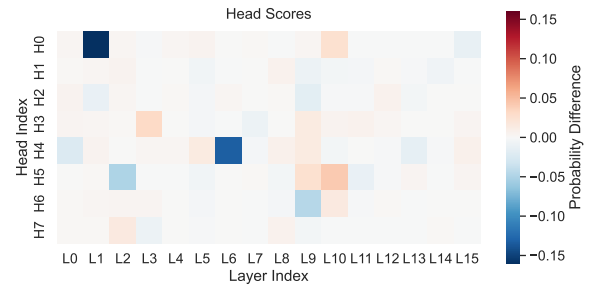


Figure A6: Probability difference values of random repetition task for HAPAX.

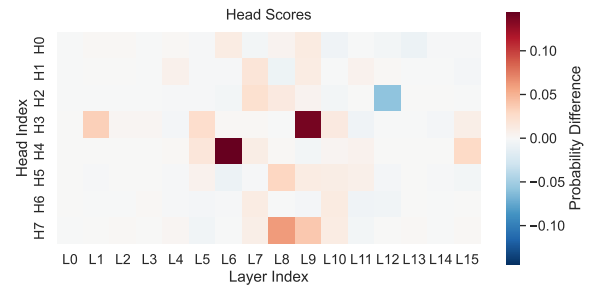


Figure A7: Probability difference values of random repetition task for Vanilla model.

Table A2: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX. Values show accuracy $\pm$ 95% CI margin (binomial normal-approximation per task). Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
AG News	32.7 $\pm$ 1.3†	5.6 $\pm$ 0.6	Antonym	0.7 $\pm$ 0.3	2.2 $\pm$ 0.6†
Capitalize (Full Word)	78.2 $\pm$ 2.8†	59.0 $\pm$ 3.4	Capitalize First Letter	37.1 $\pm$ 3.3	66.8 $\pm$ 3.2†
Capitalize Last Letter	9.3 $\pm$ 2.0†	3.8 $\pm$ 1.3	Capitalize Second Letter	11.9 $\pm$ 2.3†	2.4 $\pm$ 1.1
Chinese $\rightarrow$ English	4.1 $\pm$ 1.2	6.8 $\pm$ 1.6†	CommonsenseQA	20.4 $\pm$ 1.1†	5.2 $\pm$ 0.6
Country-Capital	29.6 $\pm$ 6.5	42.3 $\pm$ 7.0†	Country-Currency	5.4 $\pm$ 4.6	5.4 $\pm$ 4.6
French $\rightarrow$ English	11.0 $\pm$ 2.0	11.9 $\pm$ 2.0	Indonesian $\rightarrow$ English	1.0 $\pm$ 0.6	0.6 $\pm$ 0.5
Italian $\rightarrow$ English	4.8 $\pm$ 1.3	5.1 $\pm$ 1.4	Japanese $\rightarrow$ English	0.4 $\pm$ 0.4	2.3 $\pm$ 0.9†
Landmark-Country	32.8 $\pm$ 3.2	36.6 $\pm$ 3.3†	Lowercase First Letter	71.1 $\pm$ 3.1†	66.5 $\pm$ 3.2
Lowercase Last Letter	7.0 $\pm$ 1.8	7.2 $\pm$ 1.8	National Parks	18.2 $\pm$ 3.6	20.8 $\pm$ 3.7
Next Capital Letter	6.5 $\pm$ 1.7	6.3 $\pm$ 1.7	Next Item	12.4 $\pm$ 4.3	26.7 $\pm$ 5.8†
Park-Country	12.8 $\pm$ 2.4	16.8 $\pm$ 2.7†	Person-Instrument	23.9 $\pm$ 3.7†	1.4 $\pm$ 1.0
Person-Occupation	18.5 $\pm$ 2.7†	4.4 $\pm$ 1.4	Person-Sport	23.3 $\pm$ 4.6	28.6 $\pm$ 5.0
Polish $\rightarrow$ English	1.7 $\pm$ 0.8	2.2 $\pm$ 0.9	Portuguese $\rightarrow$ English	4.7 $\pm$ 1.3	8.1 $\pm$ 1.7†
Present-Past	55.3 $\pm$ 5.7	78.2 $\pm$ 4.7†	Previous Item	3.1 $\pm$ 2.3	4.4 $\pm$ 2.7
Product-Company	18.6 $\pm$ 3.3	19.5 $\pm$ 3.4	Sentiment	64.6 $\pm$ 2.7†	17.4 $\pm$ 2.2
Singular-Plural	63.9 $\pm$ 6.6	76.6 $\pm$ 5.8†	Spanish $\rightarrow$ English	7.0 $\pm$ 1.6	11.7 $\pm$ 2.0†
Synonym	1.2 $\pm$ 0.4	2.4 $\pm$ 0.6†	Word Length	7.7 $\pm$ 1.8	6.6 $\pm$ 1.7

Table A3: Performance comparison on **extractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX models. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
Adjective V Verb 3	58.8 $\pm$ 3.1†	41.7 $\pm$ 3.1	Adjective V Verb 5	52.7 $\pm$ 3.1†	31.2 $\pm$ 2.9
Alphabetically Last 5	18.6 $\pm$ 2.4	15.5 $\pm$ 2.2	Animal V Object 3	49.4 $\pm$ 3.1†	26.0 $\pm$ 2.7
Animal V Object 5	37.1 $\pm$ 3.0†	17.5 $\pm$ 2.4	Choose First Of 3	94.9 $\pm$ 1.4†	19.1 $\pm$ 2.4
Choose First Of 5	95.6 $\pm$ 1.3†	17.1 $\pm$ 2.3	Alphabetically First 3	35.5 $\pm$ 3.0†	26.8 $\pm$ 2.7
Alphabetically First 5	19.9 $\pm$ 2.5†	13.8 $\pm$ 2.1	Alphabetically Last 3	32.6 $\pm$ 2.9†	24.4 $\pm$ 2.7
Choose Middle Of 5	25.8 $\pm$ 2.7†	13.4 $\pm$ 2.1	Color V Animal 3	63.4 $\pm$ 3.0	60.2 $\pm$ 3.0
Color V Animal 5	48.4 $\pm$ 3.1	49.2 $\pm$ 3.1	Choose Last Of 3	65.6 $\pm$ 2.9†	46.5 $\pm$ 3.1
Choose Last Of 5	51.5 $\pm$ 3.1†	44.0 $\pm$ 3.1	Choose Middle Of 3	39.7 $\pm$ 3.0†	26.9 $\pm$ 2.7
Concept V Object 3	56.8 $\pm$ 3.1	54.2 $\pm$ 3.1	Conll2003 Person	72.0 $\pm$ 3.0†	19.9 $\pm$ 2.7
Fruit V Animal 3	43.7 $\pm$ 3.1†	17.7 $\pm$ 2.4	Concept V Object 5	45.3 $\pm$ 3.1	52.2 $\pm$ 3.1†
Conll2003 Location	58.5 $\pm$ 3.1†	34.7 $\pm$ 3.0	Object V Concept 5	68.9 $\pm$ 2.9†	36.3 $\pm$ 3.0
Conll2003 Organization	54.7 $\pm$ 3.3†	17.8 $\pm$ 2.5	Fruit V Animal 5	28.0 $\pm$ 2.8†	8.8 $\pm$ 1.8
Object V Concept 3	73.5 $\pm$ 2.7†	43.0 $\pm$ 3.1	Squad Val	14.5 $\pm$ 2.3†	5.4 $\pm$ 1.5
Verb V Adjective 3	70.1 $\pm$ 2.8†	44.9 $\pm$ 3.1	Verb V Adjective 5	61.0 $\pm$ 3.0†	36.3 $\pm$ 3.0

Table A4: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and Thresholded-HAPAX. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX THRESH (%)	Task	Vanilla (%)	HAPAX THRESH (%)
AG News	32.7 $\pm$ 1.3†	1.1 $\pm$ 0.3	Antonym	0.7 $\pm$ 0.3	7.3 $\pm$ 1.0†
Capitalize (Full Word)	78.2 $\pm$ 2.8†	35.9 $\pm$ 3.3	Capitalize First Letter	37.1 $\pm$ 3.3†	24.0 $\pm$ 2.9
Capitalize Last Letter	9.3 $\pm$ 2.0†	4.1 $\pm$ 1.4	Capitalize Second Letter	11.9 $\pm$ 2.3†	3.7 $\pm$ 1.3
Chinese $\rightarrow$ English	4.1 $\pm$ 1.2	10.1 $\pm$ 1.9†	CommonsenseQA	20.4 $\pm$ 1.1†	0.0 $\pm$ 0.0
Country-Capital	29.6 $\pm$ 6.5	29.1 $\pm$ 6.5	Country-Currency	5.4 $\pm$ 4.6	2.2 $\pm$ 2.5
French $\rightarrow$ English	11.0 $\pm$ 2.0	14.5 $\pm$ 2.2†	Indonesian $\rightarrow$ English	1.0 $\pm$ 0.6	1.0 $\pm$ 0.6
Italian $\rightarrow$ English	4.8 $\pm$ 1.3	6.6 $\pm$ 1.6	Japanese $\rightarrow$ English	0.4 $\pm$ 0.4	3.8 $\pm$ 1.2†
Landmark-Country	32.8 $\pm$ 3.2†	21.8 $\pm$ 2.8	Lowercase First Letter	71.1 $\pm$ 3.1†	46.6 $\pm$ 3.4
Lowercase Last Letter	7.0 $\pm$ 1.8†	3.3 $\pm$ 1.2	National Parks	18.2 $\pm$ 3.6	16.9 $\pm$ 3.5
Next Capital Letter	6.5 $\pm$ 1.7	4.2 $\pm$ 1.4	Next Item	12.4 $\pm$ 4.3	8.9 $\pm$ 3.7
Park-Country	12.8 $\pm$ 2.4	15.0 $\pm$ 2.6	Person-Instrument	23.9 $\pm$ 3.7†	0.8 $\pm$ 0.8
Person-Occupation	18.5 $\pm$ 2.7†	4.8 $\pm$ 1.5	Person-Sport	23.3 $\pm$ 4.6†	0.0 $\pm$ 0.5
Polish $\rightarrow$ English	1.7 $\pm$ 0.8	3.0 $\pm$ 1.1	Portuguese $\rightarrow$ English	4.7 $\pm$ 1.3	9.9 $\pm$ 1.9†
Present-Past	55.3 $\pm$ 5.7†	38.9 $\pm$ 5.6	Previous Item	3.1 $\pm$ 2.3	1.3 $\pm$ 1.4
Product-Company	18.6 $\pm$ 3.3†	8.8 $\pm$ 2.4	Sentiment	64.6 $\pm$ 2.7†	0.0 $\pm$ 0.1
Singular-Plural	63.9 $\pm$ 6.6†	23.4 $\pm$ 5.8	Spanish $\rightarrow$ English	7.0 $\pm$ 1.6	14.2 $\pm$ 2.2†
Synonym	1.2 $\pm$ 0.4	3.7 $\pm$ 0.7†	Word Length	7.7 $\pm$ 1.8†	0.9 $\pm$ 0.6

Table A5: Performance comparison on **extractive** in-context learning tasks (5-Shot) for Vanilla and Thresholded-HAPAX. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; † denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX THRESH (%)	Task	Vanilla (%)	HAPAX THRESH (%)
Adjective V Verb 3	58.8 $\pm$ 3.1	56.1 $\pm$ 3.1	Adjective V Verb 5	52.7 $\pm$ 3.1[†]	39.4 $\pm$ 3.0
Alphabetically Last 5	18.6 $\pm$ 2.4[†]	13.0 $\pm$ 2.1	Animal V Object 3	49.4 $\pm$ 3.1[†]	27.3 $\pm$ 2.8
Animal V Object 5	37.1 $\pm$ 3.0[†]	16.3 $\pm$ 2.3	Choose First Of 3	94.9 $\pm$ 1.4[†]	30.7 $\pm$ 2.9
Choose First Of 5	95.6 $\pm$ 1.3[†]	27.0 $\pm$ 2.8	Alphabetically First 3	35.5 $\pm$ 3.0[†]	28.1 $\pm$ 2.8
Alphabetically First 5	19.9 $\pm$ 2.5[†]	14.5 $\pm$ 2.2	Alphabetically Last 3	32.6 $\pm$ 2.9[†]	24.5 $\pm$ 2.7
Choose Middle Of 5	25.8 $\pm$ 2.7[†]	7.4 $\pm$ 1.6	Color V Animal 3	63.4 $\pm$ 3.0[†]	18.6 $\pm$ 2.4
Color V Animal 5	48.4 $\pm$ 3.1[†]	9.5 $\pm$ 1.8	Choose Last Of 3	65.6 $\pm$ 2.9[†]	53.6 $\pm$ 3.1
Choose Last Of 5	51.5 $\pm$ 3.1	58.7 $\pm$ 3.1[†]	Choose Middle Of 3	39.7 $\pm$ 3.0[†]	21.5 $\pm$ 2.5
Concept V Object 3	56.8 $\pm$ 3.1[†]	35.9 $\pm$ 3.0	Conll2003 Person	72.0 $\pm$ 3.0[†]	25.8 $\pm$ 1.4
Fruit V Animal 3	43.7 $\pm$ 3.1[†]	21.5 $\pm$ 2.5	Concept V Object 5	45.3 $\pm$ 3.1[†]	18.3 $\pm$ 2.4
Conll2003 Location	58.5 $\pm$ 3.1[†]	17.5 $\pm$ 1.1	Object V Concept 5	68.9 $\pm$ 2.9[†]	51.9 $\pm$ 3.1
Conll2003 Organization	54.7 $\pm$ 3.3[†]	21.2 $\pm$ 1.3	Fruit V Animal 5	28.0 $\pm$ 2.8[†]	9.9 $\pm$ 1.9
Object V Concept 3	73.5 $\pm$ 2.7[†]	58.4 $\pm$ 3.1	Squad Val	14.5 $\pm$ 2.3[†]	8.5 $\pm$ 0.8
Verb V Adjective 3	70.1 $\pm$ 2.8[†]	33.6 $\pm$ 2.9	Verb V Adjective 5	61.0 $\pm$ 3.0[†]	20.8 $\pm$ 2.5

Table A6: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX by ensuring that the target answer for each test instance is not included in the 5-shot examples. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; † denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
AG News	0.3 $\pm$ 0.1	7.4 $\pm$ 0.7[†]	Antonym	0.6 $\pm$ 0.3	2.5 $\pm$ 0.6[†]
Capitalize (Full Word)	78.2 $\pm$ 2.8[†]	59.0 $\pm$ 3.4	Capitalize First Letter	36.5 $\pm$ 3.3	74.0 $\pm$ 3.0[†]
Capitalize Last Letter	3.1 $\pm$ 1.2	6.3 $\pm$ 1.7[†]	Capitalize Second Letter	0.3 $\pm$ 0.3	3.2 $\pm$ 1.2[†]
Chinese→English	4.1 $\pm$ 1.2	6.8 $\pm$ 1.6[†]	CommonsenseQA	11.2 $\pm$ 0.9[†]	10.0 $\pm$ 0.8
Country-Capital	30.7 $\pm$ 6.6	42.3 $\pm$ 7.0[†]	Country-Currency	5.4 $\pm$ 4.6	8.6 $\pm$ 5.7
French→English	11.0 $\pm$ 2.0	11.9 $\pm$ 2.0	Indonesian→English	1.0 $\pm$ 0.6	0.6 $\pm$ 0.5
Italian→English	4.8 $\pm$ 1.3	5.1 $\pm$ 1.4	Japanese→English	0.4 $\pm$ 0.4	2.3 $\pm$ 0.9[†]
Landmark-Country	31.0 $\pm$ 3.1	39.2 $\pm$ 3.3[†]	Lowercase First Letter	72.4 $\pm$ 3.1	76.8 $\pm$ 2.9[†]
Lowercase Last Letter	3.7 $\pm$ 1.3	10.1 $\pm$ 2.1[†]	National Parks	17.1 $\pm$ 3.5	22.8 $\pm$ 3.9[†]
Next Capital Letter	3.3 $\pm$ 1.2	8.6 $\pm$ 1.9[†]	Next Item	12.4 $\pm$ 4.3	27.6 $\pm$ 5.8[†]
Park-Country	11.3 $\pm$ 2.3	16.8 $\pm$ 2.7[†]	Person-Instrument	0.0 $\pm$ 0.3	3.5 $\pm$ 1.6[†]
Person-Occupation	0.4 $\pm$ 0.4	2.6 $\pm$ 1.1[†]	Person-Sport	10.1 $\pm$ 3.3	42.8 $\pm$ 5.4[†]
Polish→English	1.7 $\pm$ 0.8	2.2 $\pm$ 0.9	Portuguese→English	4.7 $\pm$ 1.3	8.1 $\pm$ 1.7[†]
Present-Past	55.3 $\pm$ 5.7	78.2 $\pm$ 4.7[†]	Previous Item	2.7 $\pm$ 2.1	3.6 $\pm$ 2.4
Product-Company	15.5 $\pm$ 3.1	25.9 $\pm$ 3.8[†]	Sentiment	2.7 $\pm$ 0.9	36.8 $\pm$ 2.8[†]
Singular-Plural	63.9 $\pm$ 6.6	76.6 $\pm$ 5.8[†]	Spanish→English	7.0 $\pm$ 1.6	11.7 $\pm$ 2.0[†]
Synonym	1.4 $\pm$ 0.4	2.3 $\pm$ 0.6[†]	Word Length	8.1 $\pm$ 1.9	15.4 $\pm$ 2.5[†]

Table A7: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and Thresholded-HAPAX by ensuring that the target answer for each test instance is not included in the 5-shot examples. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; † denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX THRESH (%)	Task	Vanilla (%)	HAPAX THRESH (%)
AG News	0.3 $\pm$ 0.1	3.4 $\pm$ 0.5[†]	Antonym	0.6 $\pm$ 0.3	6.9 $\pm$ 1.0[†]
Capitalize (Full Word)	78.2 $\pm$ 2.8[†]	35.9 $\pm$ 3.3	Capitalize First Letter	36.5 $\pm$ 3.3[†]	26.9 $\pm$ 3.0
Capitalize Last Letter	3.1 $\pm$ 1.2	8.0 $\pm$ 1.9[†]	Capitalize Second Letter	0.3 $\pm$ 0.3	7.8 $\pm$ 1.9[†]
Chinese→English	4.1 $\pm$ 1.2	10.1 $\pm$ 1.9[†]	CommonsenseQA	11.2 $\pm$ 0.9[†]	0.0 $\pm$ 0.0
Country-Capital	30.7 $\pm$ 6.6	29.1 $\pm$ 6.5	Country-Currency	5.4 $\pm$ 4.6	3.2 $\pm$ 3.4
French→English	11.0 $\pm$ 2.0	14.5 $\pm$ 2.2[†]	Indonesian→English	1.0 $\pm$ 0.6	1.0 $\pm$ 0.6
Italian→English	4.8 $\pm$ 1.3	6.6 $\pm$ 1.6	Japanese→English	0.4 $\pm$ 0.4	3.8 $\pm$ 1.2[†]
Landmark-Country	31.0 $\pm$ 3.1[†]	25.7 $\pm$ 3.0	Lowercase First Letter	72.4 $\pm$ 3.1[†]	61.1 $\pm$ 3.3
Lowercase Last Letter	3.7 $\pm$ 1.3	5.0 $\pm$ 1.5	National Parks	17.1 $\pm$ 3.5	24.2 $\pm$ 4.0[†]
Next Capital Letter	3.3 $\pm$ 1.2	6.3 $\pm$ 1.7[†]	Next Item	12.4 $\pm$ 4.3	9.8 $\pm$ 3.9
Park-Country	11.3 $\pm$ 2.3	16.0 $\pm$ 2.7[†]	Person-Instrument	0.0 $\pm$ 0.3	1.6 $\pm$ 1.1[†]
Person-Occupation	0.4 $\pm$ 0.4	7.8 $\pm$ 1.8[†]	Person-Sport	10.1 $\pm$ 3.3[†]	2.5 $\pm$ 1.7
Polish→English	1.7 $\pm$ 0.8	3.0 $\pm$ 1.1	Portuguese→English	4.7 $\pm$ 1.3	9.9 $\pm$ 1.9[†]
Present-Past	55.3 $\pm$ 5.7[†]	38.9 $\pm$ 5.6	Previous Item	2.7 $\pm$ 2.1	1.3 $\pm$ 1.4
Product-Company	15.5 $\pm$ 3.1	20.7 $\pm$ 3.5[†]	Sentiment	2.7 $\pm$ 0.9[†]	0.0 $\pm$ 0.1
Singular-Plural	63.9 $\pm$ 6.6[†]	23.9 $\pm$ 5.8	Spanish→English	7.0 $\pm$ 1.6	14.2 $\pm$ 2.2[†]
Synonym	1.4 $\pm$ 0.4	3.4 $\pm$ 0.7[†]	Word Length	8.1 $\pm$ 1.9[†]	2.0 $\pm$ 1.0

Table A8: Task categories and their constituent tasks. The four-way categorization follows [Hendel et al. \(2023\)](#), extended with a Sentence-level family covering classification tasks that operate over full passages.

Task family	Tasks included	Description
Algorithmic (8)	Capitalize (Full Word), Capitalize First Letter, Capitalize Last Letter, Capitalize Second Letter, Lowercase First Letter, Lowercase Last Letter, Next Capital Letter, Word Length	Tasks solvable by a character-level rule, such as capitalization, character selection, or counting.
Linguistic (4)	Antonym, Present-Past, Singular-Plural, Synonym	Tasks that require lexical or morphological knowledge.
Knowledge (11)	Country-Capital, Country-Currency, Landmark-Country, National Parks, Next Item, Park-Country, Person-Instrument, Person-Occupation, Person-Sport, Previous Item, Product-Company	Tasks that require parametric world knowledge.
Translation (8)	Chinese→English, French→English, Indonesian→English, Italian→English, Japanese→English, Polish→English, Portuguese→English, Spanish→English	Tasks where an input word is mapped from a source language to its target-language equivalent.
Sentence-level (3)	AG News, CommonsenseQA, Sentiment	Classification tasks over full passages.

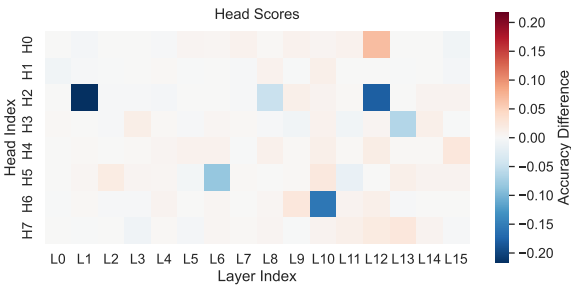


Figure A8: Accuracy difference values of random repetition task for Thresholded-HAPAX.

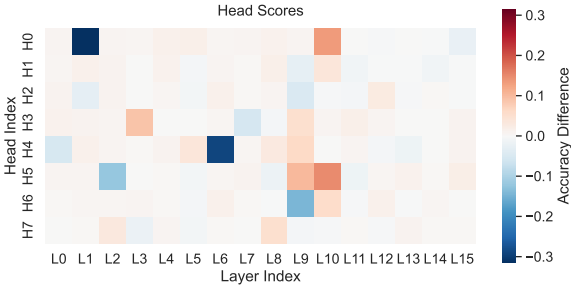


Figure A9: Accuracy difference values of random repetition task for HAPAX

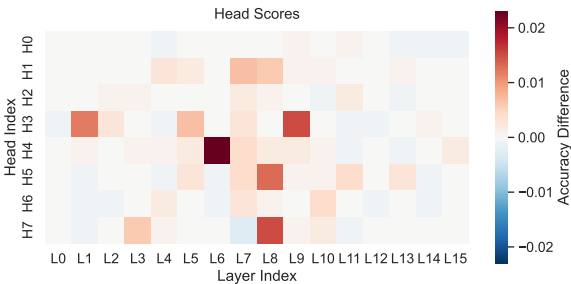


Figure A10: Accuracy difference values of random repetition task for Vanilla model.

Natural Text Non-Repeating Performance

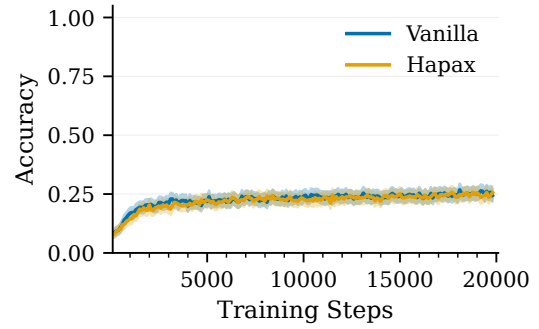


Figure A11: The non-repeating natural text repetition accuracy. Instead of giving a repeated sequence $r_1 r_2 \dots r_s r_1 r_2 \dots r_{s-1}$ we only give $r_1 r_2 \dots r_{s-1}$. This shows that the initial increase seen in the natural text repetition also occurs for non-repeating samples.

A.3 Loss Analysis

Figure A12 and A13 display the loss relevant analysis.

A.4 Logit Lens on Prefix-Matching Attention Heads

Within A14, we give the per-head contributions of logit lens analysis.

A.5 Function-Vector and Concept Head Analysis

Within this section, we use Causal Mediation Analysis to find Function Vector heads proposed by [Todd et al. \(2024\)](#) and Concept Head Score analysis from [Feucht et al. \(2025\)](#) to analyze whether the HAPAX model can develop such heads that have been found to be important for In-Context Learn-

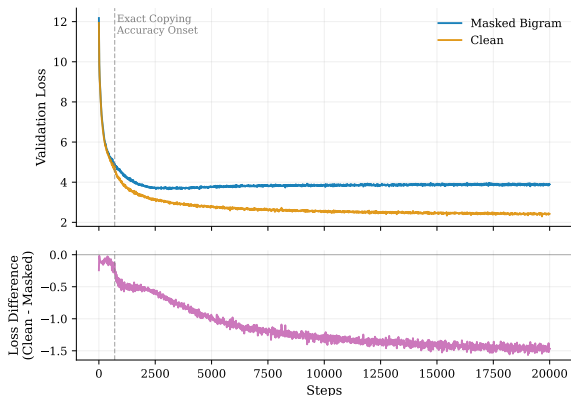


Figure A12: Validation Loss of Vanilla and Hapax model. Despite having increased loss value, as mentioned in the main paper, the validation loss also is mostly affected by the prediction of exact copying instances.

ing (Figure A15 and Figures A17). Using this, we find that the HAPAX model can develop Function Vector and Concept heads. Secondly, we verify that Function Vector heads are causally important for abstractive ICL, which is in line with (Yin and Steinhardt, 2025) (Figure A15). Another finding is that as we ablate previous-token heads, we observe that within the Vanilla model, only ablating the top 2 previous-token heads is enough to pull the accuracy to near 0. However, with Hapax and Thresholded-Hapax, we observe that the distribution these models are trained on created a circuitry where progressively the influence of previous-token heads becomes less destructive, meaning that the training provides more distributed ICL strategies rather than the tightly coupled version in Vanilla model (Figure A16).

A.6 Attention Pattern Scores

Figures A18, A19, A20 display attention patterns of the trained models.

B Cosine Similarity Threshold Selection

The main purpose of Thresholded-HAPAX is to create a stricter training regime for suppression of inductive copying. Our hypothesis is that even if we mask exact verbatim copying signals, there might still be token representations with high representational similarity, high enough to still provide a weak signal for verbatim copying. To test this, we select the cosine similarity threshold by using the embedding space of the last step of the vanilla model as a proxy for finding similar tokens. We create a cosine similarity matrix and analyze

the cosine similarities of the closest neighbors and their edit distances. After doing this, we select the threshold of 0.3, which corresponds to the top-4 closest neighbors on average (Figure B21) and an average edit distance of 3.65 (Figure B22). Tokens representing different strings may still have highly similar embeddings, so our thresholding deliberately overestimates these hypothesized similar tokens, which makes us more confident that we are removing copying signal. Given our limited compute budget, we adopt this strict threshold to ensure that the trained model consistently masks the majority of similar tokens. Although this eventually causes harm for abstractive tasks when compared to the regular HAPAX model, we observe consistent gains for translation, suggesting that the model keeps capabilities for such instances despite the heavy masking. This control over copying also cannot solely be explained by undertraining, because the Thresholded-HAPAX model is near zero at copying while still achieving 29.1% accuracy on non-trivial tasks like Country-Capital. Therefore, even if it performs worse than the vanilla model, it serves as a proof of concept that the model can achieve non-trivial capabilities in abstractive tasks despite having even more reduced copying and prefix-matching scores.

To further justify our use of input embedding cosine similarity (instead of edit distance), we analyze the relationship between string-level edit distance and embedding cosine similarity for the top-50 nearest neighbors in the vanilla model’s embedding space (Figure B24). With a cosine similarity threshold of 0.3, 7.24% of the token pairs are treated as equal, and only 1% of these pairs have an edit distance higher than 3. In contrast, if we instead threshold by edit distance at 3, 34.8% of the pairs are treated as equal, and 28.56% of these have a cosine similarity below 0.3. Moreover, an edit distance threshold of 3 still fails to capture 1% of the pairs whose cosine similarity is above 0.3. Therefore, edit distance treats many more pairs as equal but still does not capture some high-cosine-similarity token pairs, which we hypothesize are more closely related to copying signal. Since attention heads attend by hidden state similarity, cosine similarity is closer to what an attention head would consider similar. We use cosine similarity as our base and use the average edit distance at different cosine similarity thresholds to guide us, so that we make sure we are removing tokens above a certain cosine similarity while also understanding

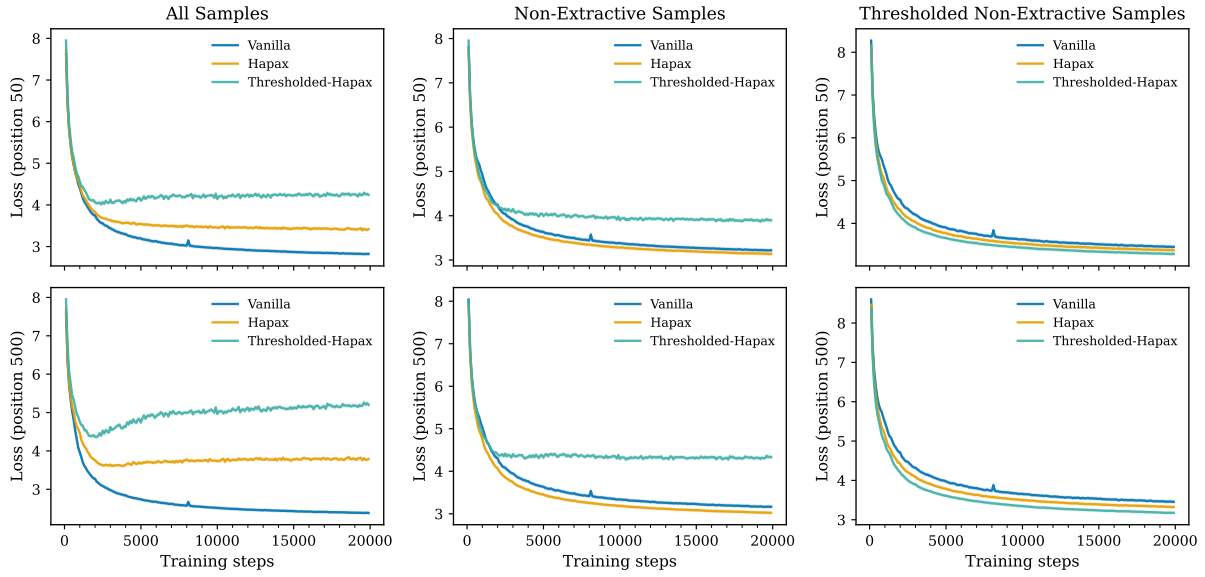


Figure A13: Loss values at individual token positions. The loss value for Non-Extractive Samples is lower for both token positions than the Vanilla in HAPAX.

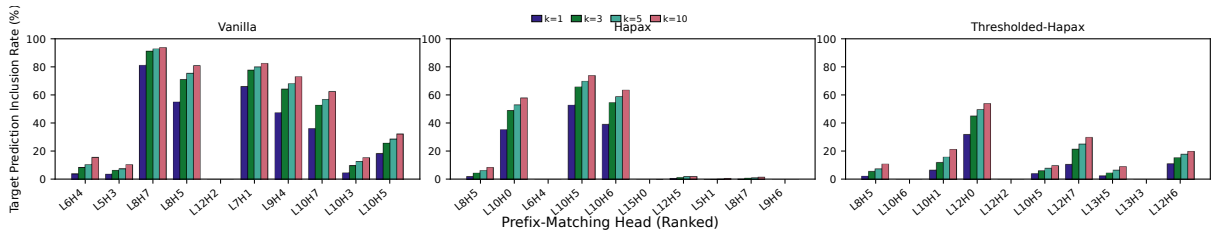


Figure A14: Per-head analysis of the percentage of times the target token is included in the top-k promoted tokens for the top 10 prefix-matching heads.

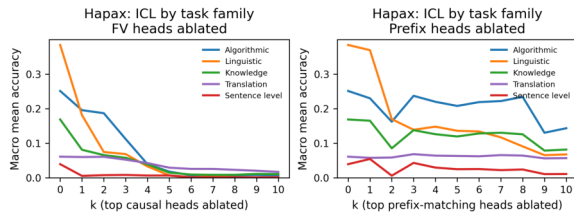


Figure A15: Similar to the findings from (Yin and Steinhart, 2025), we find FV heads and find them to be causally more important than heads with high prefix-matching scores.

how close they are on average to their represented tokens.

C HAPAX Fluency Evaluation

We evaluate the effects of HAPAX training on the model’s ability to generate fluent natural text and compare the results to its vanilla counterpart as a baseline. Inspired by the Fluency score from Axbench (Wu et al., 2025), we use an automated LLM-based approach to assess the fluency of sen-

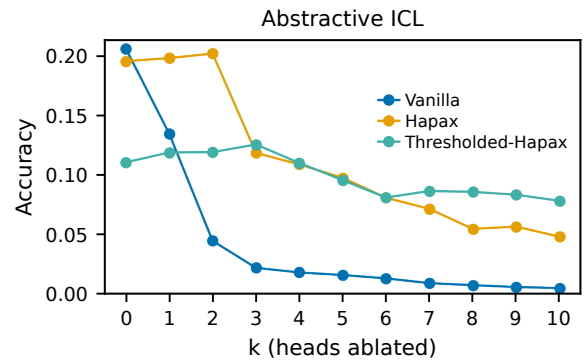


Figure A16: Ablation of previous-token head is more destructive in Vanilla, whereas the influence of ablation is progressively lower on the accuracy for hapax and thresholded hapax.

tence completions generated by each model, with discrete ratings of 0, 1, and 2 for fluent language. Sequence continuations, with a maximum length of 128 tokens, are generated based on a diverse seed of 20 tokens, sampled from the RedPajama dataset (Weber et al., 2024).

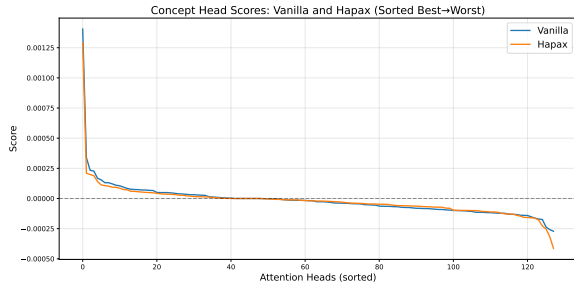


Figure A17: Using the concept head scores from [Feucht et al. \(2025\)](#), we display the scores of the models from best to worst. The trends are similar for both the Vanilla and HAPAX models, indicating that suppressing inductive copying does not noticeably affect the causal concept head metrics.

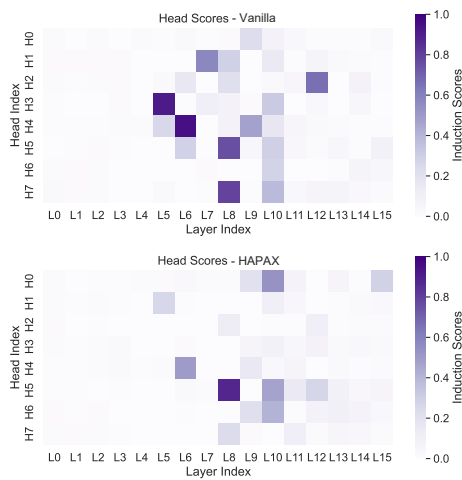


Figure A18: Prefix-Matching Scores across Vanilla and HAPAX models. HAPAX model has fewer heads that have a strong prefix-matching score.

We find in Figure C26 that our HAPAX trained model generates somewhat fluent text 70% more times than the vanilla model. The latter, performing poorly, generates non-fluent language 84% of the time. Looking at some qualitative examples of generation in Figure C25, we observe that the vanilla model defaults to repeating itself rather early, creating incoherent language, which explains its negative results. On the other hand, the HAPAX model is able to produce chained narratives with much more fluid and natural structure.

D Experimental Details

D.1 Number of Samples Used & Data Selection

After giving the few-shot examples, we let the model generate 8 tokens. If the correct prediction requires more than 8 token predictions, we

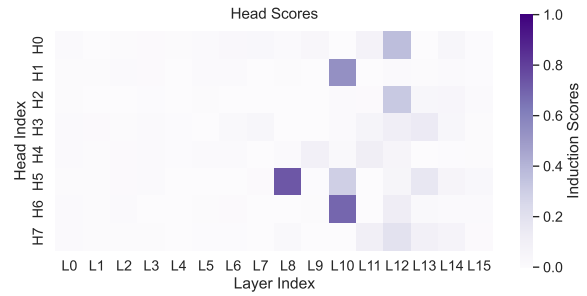


Figure A19: Prefix Matching Scores for Thresholded-HAPAX.

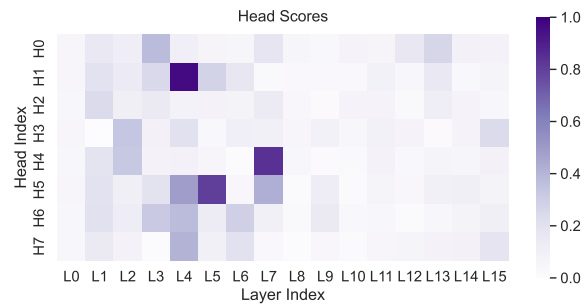


Figure A20: Previous Token Head of vanilla model at step 5000. L4 is the first layer that has a strong previous token head, which is L4 H1.

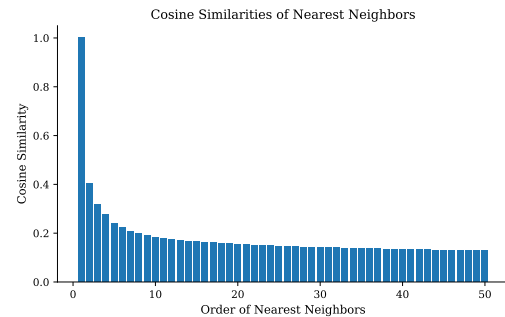


Figure B21: Cosine similarities of nearest N neighbors for clean model.

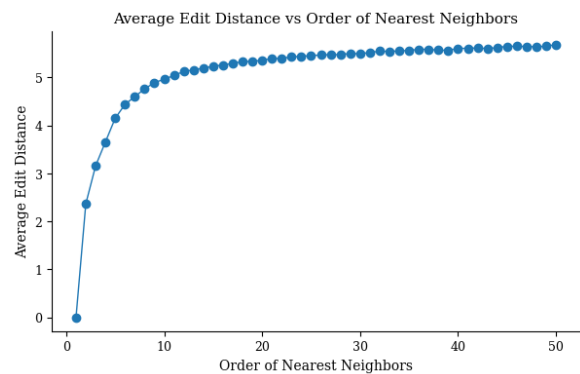


Figure B22: Average edit distance of nearest neighbors for vanilla model.

Table D9: Per-task evaluation sample sizes for FV ICL tasks. **Pool** is the benchmark size. **Used** is the number of samples used after the filtering logic is applied.

Task	Pool	Used	Task	Pool	Used
<i>FV abstractive ICL</i>					
AG News	7,600	5,000	Antonym	2,398	2,398
Capitalize (Full Word)	813	813	Capitalize First Letter	813	813
Capitalize Last Letter	814	814	Capitalize Second Letter	787	787
Chinese → English	981	981	CommonsenseQA	10,962	5,000
Country-Capital	197	189	Country-Currency	197	93
French → English	981	981	Indonesian → English	981	981
Italian → English	981	981	Japanese → English	981	981
Landmark-Country	836	836	Lowercase First Letter	814	814
Lowercase Last Letter	814	814	National Parks	451	451
Next Capital Letter	814	814	Next Item	225	225
Park-Country	749	732	Person-Instrument	510	510
Person-Occupation	821	821	Person-Sport	318	318
Polish → English	981	981	Portuguese → English	981	981
Present-Past	293	293	Previous Item	225	225
Product-Company	522	522	Sentiment	1,167	1,167
Singular-Plural	205	205	Spanish → English	981	981
Synonym	2,880	2,880	Word Length	814	814
<i>FV extractive ICL</i>					
Adjective vs Verb (3)	1,000	1,000	Adjective vs Verb (5)	1,000	1,000
Alphabetically First (3)	1,000	1,000	Alphabetically First (5)	1,000	1,000
Alphabetically Last (3)	1,000	1,000	Alphabetically Last (5)	1,000	1,000
Animal vs Object (3)	1,000	1,000	Animal vs Object (5)	1,000	1,000
Choose First of 3	1,000	1,000	Choose First of 5	1,000	1,000
Choose Last of 3	1,000	1,000	Choose Last of 5	1,000	1,000
Choose Middle of 3	1,000	1,000	Choose Middle of 5	1,000	1,000
Color vs Animal (3)	1,000	1,000	Color vs Animal (5)	1,000	1,000
Concept vs Object (3)	1,000	1,000	Concept vs Object (5)	1,000	1,000
CoNLL: Location	4,499	4,499	CoNLL: Organization	3,843	3,836
CoNLL: Person	3,544	3,539	Fruit vs Animal (3)	1,000	1,000
Fruit vs Animal (5)	1,000	1,000	Object vs Concept (3)	1,000	1,000
Object vs Concept (5)	1,000	1,000	SQuAD Validation	4,944	4,839
Verb vs Adjective (3)	1,000	1,000	Verb vs Adjective (5)	1,000	1,000

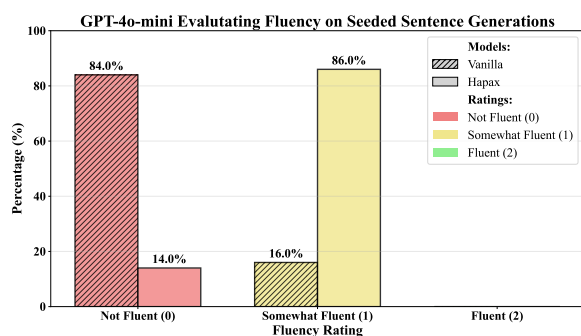


Figure C26: GPT-4o-mini as a judge rates 100 samples of natural text generated by each model based on fluency.

as our evaluation metric. For each synset, we use only the first lemma from the source language to predict the corresponding English translation. All evaluations are 5-shot, and we translate from each source language into English to ensure consistent token counts across all tasks. An example prompt is shown in Figure D27. Compatible with the licensing terms, the data is only to be used for research purposes.

D.3 Low Cardinality Label Space

Some function vector tasks have low cardinality label spaces, meaning the target prediction is likely

```
"boite de conserve" -
"can "
"adresse" - "address"
"abus" - "maltreatment"
"bétail" - "livestock"
"argile" - "mud"
"chose" - "
```

Figure D27: French–English example. Accuracy uses prefix matching on the gold word (optional leading space). The continuation must equal the gold token or immediately follow it with a separator character: space, newline, tab, ' , , . , or end.

to appear among the few-shot demonstrations. For instance, with 5-shot demonstrations and a 4-label task, a model could achieve correct predictions by distributional matching alone, simply sampling labels from the demonstrations. To disentangle genuine task understanding from this confound, we also evaluate with target exclusion. The Person-Instrument task has only 4 labels (Figure D28), and Capitalize Second Letter, while nominally having 27 labels, is heavily skewed with the top 4 covering 59.7% of samples (Figure D29).

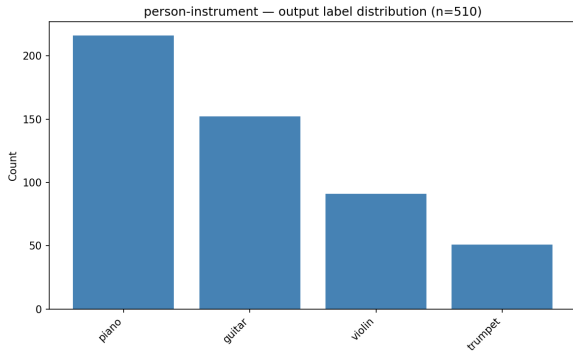


Figure D28: Label distribution of Person-Instrument task.

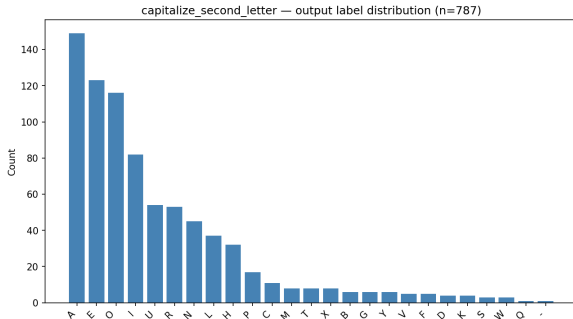


Figure D29: Label distribution of Capitalize Second Letter task.

E Explored Training Variants

Before finalizing the HAPAX training protocol, we explored several variants to understand the impact of inductive copying on learning abstractive ICL capabilities. The methods proposed here focus mainly on the first part of the induction circuit, namely the prefix-matching behavior. The motivation behind these methods is to eliminate the initial circuit component, which we expected would remove inductive copying behavior. We summarize these variants and discuss why they fail and are therefore not suitable for our purposes.

E.1 Loss Penalty Based Interventions for Prefix-Matching Score

To analyze the impact of inductive copying on the development of abstractive ICL capabilities, we included a loss function that penalizes the prefix-matching attention pattern commonly observed in induction heads. Given our loss function $\mathcal{L}$, we add the prefix-matching score across all heads to our loss function:

$$\mathcal{L}_{\text{pm}} = \lambda \frac{1}{LH} \sum_{l=1}^L \sum_{h=1}^H \text{PrefixMatching}(l, h), \quad (6)$$

where λ is our hyperparameter for fine-tuning. We propose two different versions. In the first one, we compute the prefix-matching score on the synthetically generated random repetition sequence. The random sequences are given in separate forward passes along with regular training. In the second version, instead of using synthetic data, we compute the prefix-matching score on the training data itself. We use the same positions as we use to mask the loss in Section 3.1.

The initial approach overfits on the synthetic data and consequently, the prefix-matching pattern and copying is intact for natural language. The second approach reduces prefix-matching score for both synthetic and natural settings. However, although we observe that the loss function can control the prefix-matching score, the model can still do random copying. We give random repetition accuracies for two different lambda values in Figure E30 for the non-synthetic variation. When we analyze individual attention patterns on the random repetition task, we find that the model learns to copy from two-tokens ahead, instead of displaying the regular prefix-matching attention pattern. As an example, in Figure E31, red squares indicate the regular prefix-matching positions. However, when we penalize the attention pattern, the model can still read the information from the 1-offset positions. Therefore, attention-pattern based training interventions are ill-suited for our purposes. This observation also motivate us to focus on *incentives*, as we did for HAPAX, rather than directly optimizing for a specific behavior, since the model’s solution space tends to discover alternative ways to achieve the objective, similar to reward hacking.

E.2 Reinitializing Induction Heads

To analyze whether weight reinitialization would allow us to suppress inductive copying, we have cre-

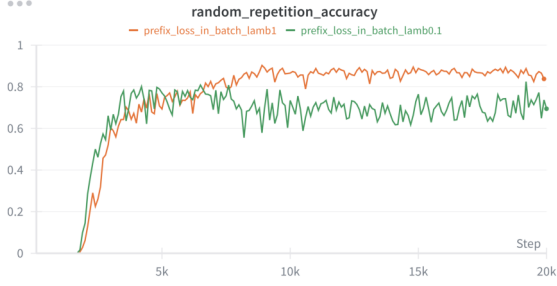


Figure E30: Random repetition accuracy for models trained with prefix-matching loss. Two lambda values that reduces the prefix-matching score are still able to learn random repetition.

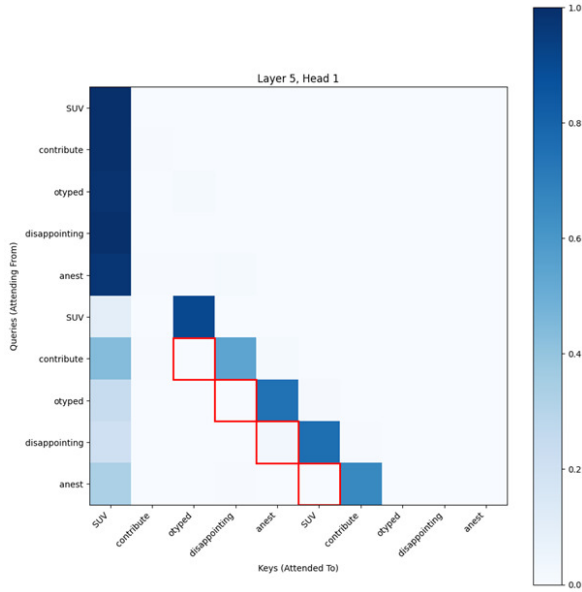


Figure E31: Example attention pattern of a model trained with a loss function that penalizes prefix-matching scores. Red squares indicate prefix-matching positions. Although the model does not display the typical attention pattern for induction, it learns to read the copying information from 1-offset positions.

ated two different training protocols. First, we reinitialized any attention head whose prefix-matching score was above a threshold τ , which is 0.3 in our case. Secondly, in addition to reinitializing the weights, we have reinitialized the optimizer states for the corresponding attention heads that go above the threshold. For the first case, we have observed that the model can continue its training but the reinitialization creates more induction heads as compared to what it would learn in the vanilla version (Figure E33). When we also reinitialize the corresponding optimizer states, the model training procedure gets stuck (Figure E32, E34) and causes every attention head to have an increased prefix-matching score. Therefore, when we keep the same data distribution, the model tends to converge on similar solutions for copying behavior, which makes the presented methods unsuitable for our purposes.

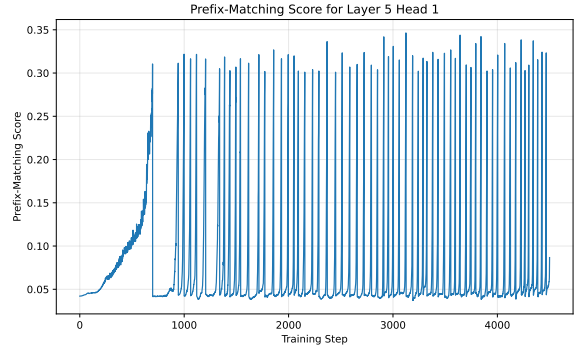


Figure E32: When the optimizer state for induction heads is reinitialized in combination with the weights, the model gets stuck and tries to relearn induction heads. We plot prefix-matching scores before we reinitialize, which is why certain data points exceed 0.3.

F Choice of Statistical Testing

For downstream tasks where evaluation reduces to accuracy, each example yields a paired binary outcome across models (correct vs. incorrect). McNemar’s test is appropriate for this paired binary setting, as it assesses whether disagreements between the two models on the same examples are symmetric. Prior work has shown that McNemar’s test is well-calibrated for classifier comparison in such conditions (Dietterich, 1998). Accordingly, we use McNemar’s test for our main accuracy-based comparisons. To control the false discovery rate, we apply the Benjamini-Hochberg procedure at FDR level 0.05 within each table.

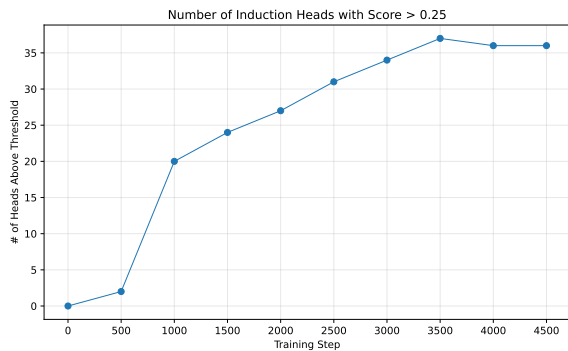


Figure E33: Number of induction heads that exceed the 0.25 threshold for the reinitialized model (160M). Since we reinitialize heads that exceed 0.3, we plot heads that exceed 0.25. The reinitialization causes the model to create three times more induction heads as compared to its vanilla version.

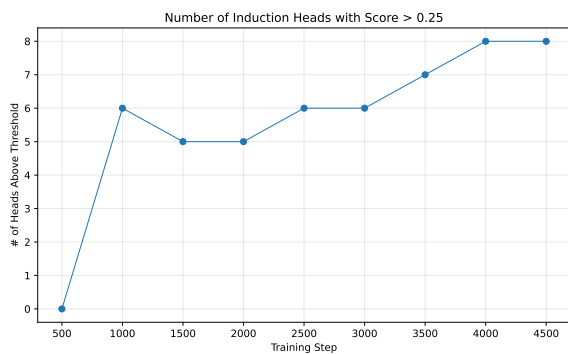


Figure E34: Number of attention heads that exceed 0.25 threshold for a vanilla model (160M).

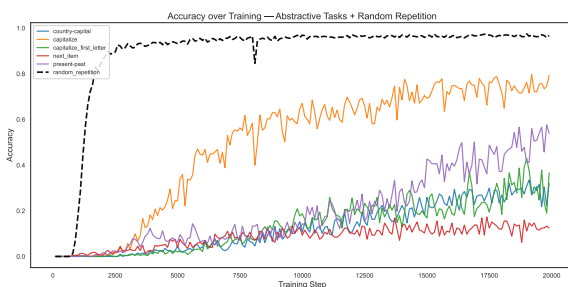


Figure E35: Accuracies of sample tasks over training for the Vanilla model. Random repetition accuracy is the first task to improve, and its rise coincides with the training step where the loss drops sharply (Figure A12). This motivates examining whether the decrease in loss and the increase in inductive copying translate into meaningful gains on abstractive tasks, which according to our evidence are not tightly coupled.

G Additional Seed Runs

In order to understand whether the emergence of abstractive ICL, prefix-matching scores, and TLD scores show similar qualitative patterns, we have trained two additional seeds, differing from the configuration specified in the official Pythia run. The first difference is that the actual specified seed is 1234, whereas now we use seeds 1337 and 2807. Secondly, due to compute limitations, our results do not use the same machine and configuration used for the main runs given within the paper. Differing from the main run and the original Pythia run, we have used ROCm 6.2.4, where we used the torch 2.5.1+rocm6.2 library instead of CUDA. Due to incompatibility with Flash Attention, SDPA has been used. The remaining libraries match the main run. On this hardware (MI250X), we experienced underflowing issues with fp16 training, so bf16 precision has been used throughout training. These runs took approximately 3030 MI250X-hours for 18000 steps. These runs, although not run in an equivalent environment to the main run, allow us to see whether the qualitative patterns hold.

For TLD in Figures G36 and G37, Vanilla rises to a peak of 0.45 at step 1600 and then holds a plateau of 0.42 for the remainder of training, while HAPAX remains near zero through step 1800 and then declines steadily into negative values, in line with the decrease happening in Figure 5. As a check of the Figure 6 experiment, we observe in Figures G38 and G39 that Vanilla training reproduces the rise-then-decay trajectory seen in Figure 6, with 8 and 9 (respectively for each seed) out of 128 heads reaching $>80\%$ prefix-matching score, whereas HAPAX transitions gradually, with only 1 and 2 out of 128 heads exceeding 80% at any point over the 18000 steps. Despite the trends observed in TLD and Prefix-Matching, the target-excluded curves of both models in Figure H42 emerge at similar times, where both remain below 3% for target excluded until step 3000 and rise together thereafter. This indicates that HAPAX can develop abstractive capabilities despite the reduction in TLD and prefix-matching scores, in line with the pattern observed in Figure 4. This gives additional evidence that HAPAX can suppress the phase transition and its associated metrics while still allowing for the emergence of abstractive ICL.

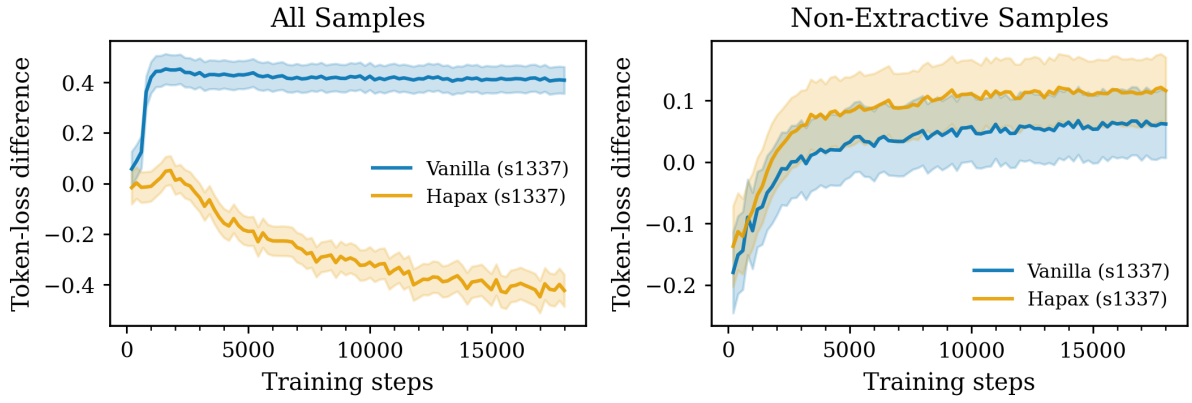


Figure G36: Comparison of token-loss metrics across training for seed 1337.

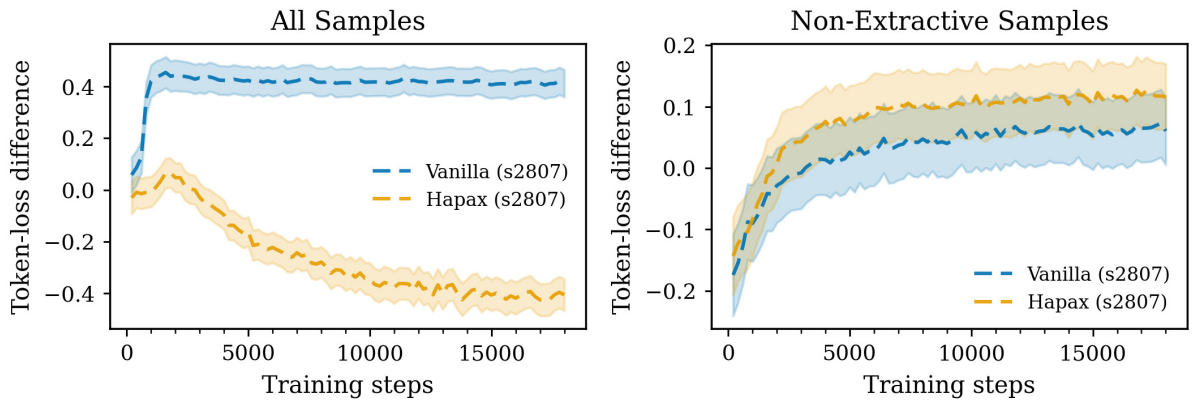


Figure G37: Comparison of token-loss metrics across training for seed 2807.

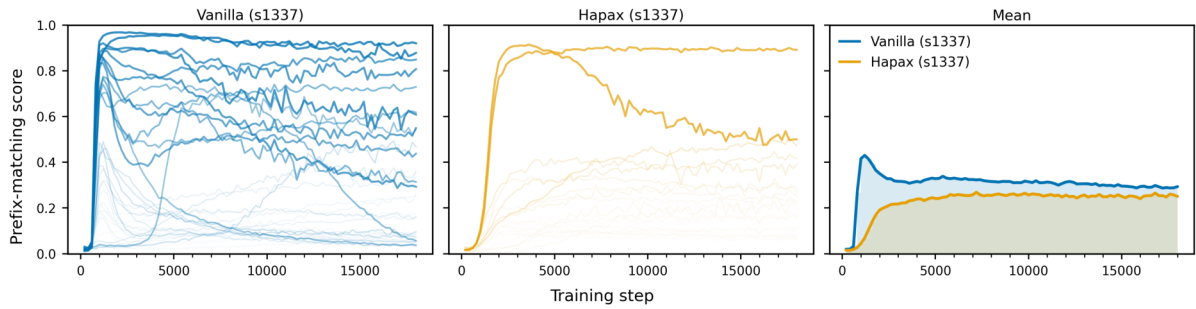


Figure G38: Comparison of prefix-matching scores across training for seed 1337.

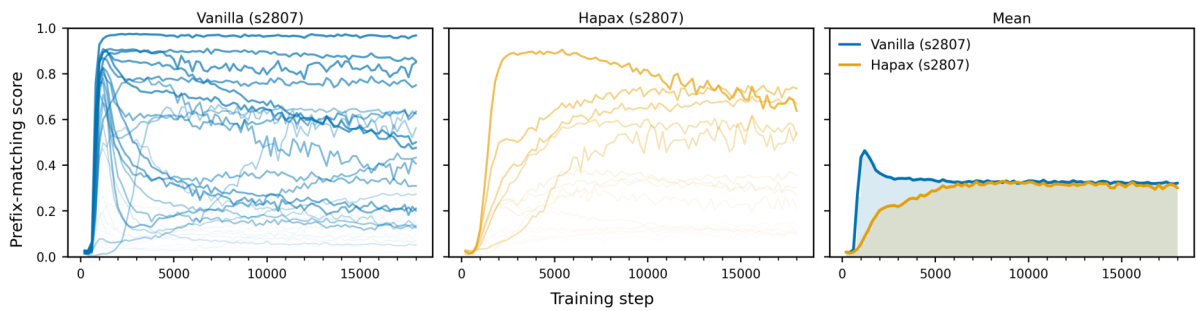


Figure G39: Comparison of prefix-matching scores across training for seed 2807.

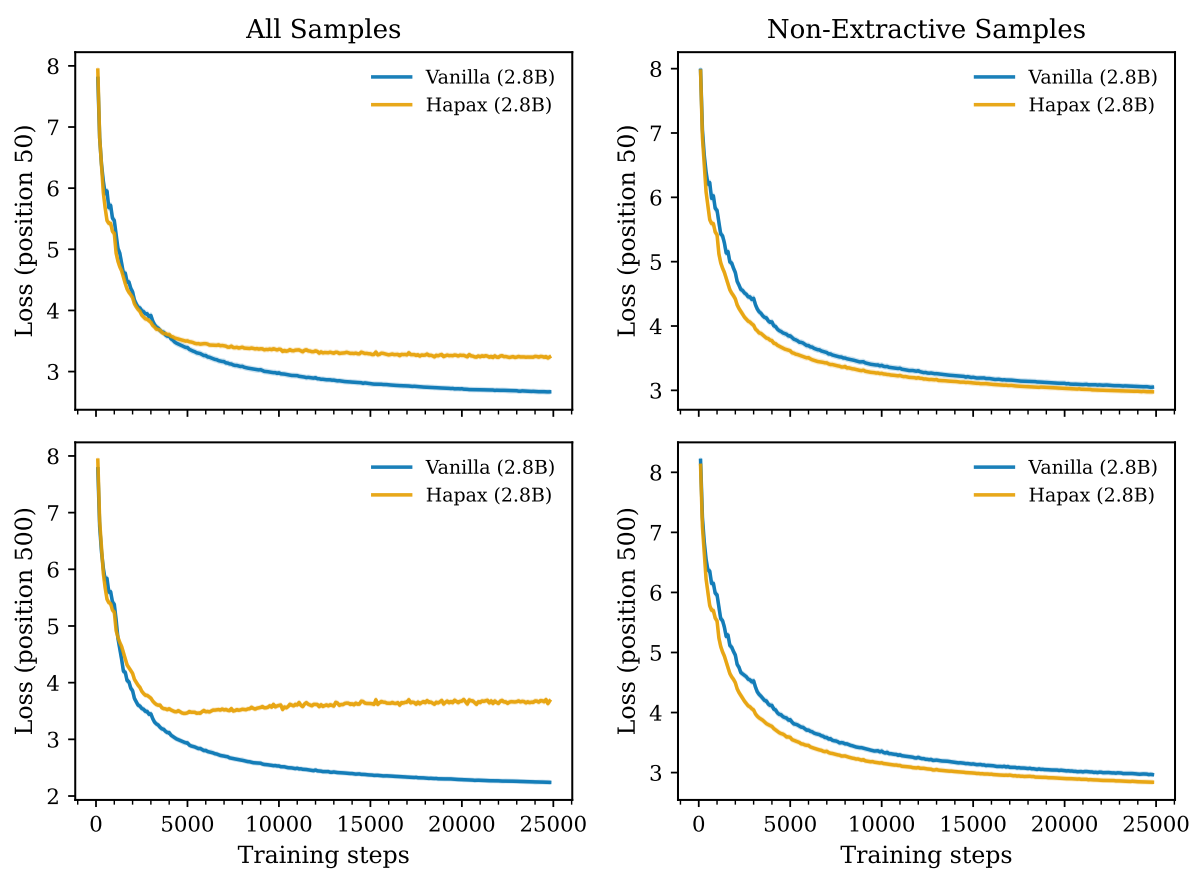


Figure H40: Loss values at individual token positions for 2.8B models. The loss value for Non-Extractive Samples is lower for both token positions than the Vanilla in HAPAX.

Table H10: Abstractive task performance summary for the 2.8B model at step 24900 over 34 tasks. Macro-averaged accuracies (%) within each family are shown with 95% confidence intervals (normal approximation on the macro mean per category). **Sig.** gives the number of tasks with significant accuracy differences under McNemar tests with joint Benjamini–Hochberg correction ($\alpha = 0.05$). The **> Vanilla** column gives the number and fraction of those significant tasks where the corresponding variant has higher accuracy than vanilla. *Standard* includes the target label in demonstrations; *Target excl.* withholds the target label.

Task Category	Vanilla vs. HAPAX (2.8B)			
	Van.	HAPAX	Sig.	> Van.
<i>Algorithmic</i> ($n = 8$)				
Standard	25.1 ± 0.9	30.0 ± 0.8	6	3/6 (50%)
Target excl.	23.2 ± 0.8	34.8 ± 0.8	7	7/7 (100%)
<i>Linguistic</i> ($n = 4$)				
Standard	36.1 ± 2.1	41.8 ± 1.9	3	3/3 (100%)
Target excl.	36.1 ± 2.1	41.9 ± 1.9	3	3/3 (100%)
<i>Knowledge</i> ($n = 11$)				
Standard	21.4 ± 1.3	24.9 ± 1.4	8	6/8 (75%)
Target excl.	16.5 ± 1.2	28.3 ± 1.4	9	9/9 (100%)
<i>Translation</i> ($n = 8$)				
Standard	9.6 ± 0.6	13.3 ± 0.7	7	7/7 (100%)
Target excl.	9.6 ± 0.6	13.3 ± 0.7	7	7/7 (100%)
<i>Sentence-level</i> ($n = 3$)				
Standard	41.2 ± 1.1	23.2 ± 1.0	3	0/3 (0%)
Target excl.	12.7 ± 0.9	55.6 ± 0.9	3	3/3 (100%)
All ($n = 34$)				
Standard	23.0 ± 0.6	25.2 ± 0.6	27	19/27 (70%)
Target excl.	18.4 ± 0.5	30.3 ± 0.6	29	29/29 (100%)

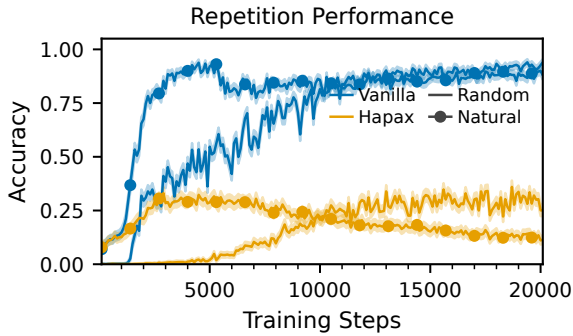


Figure H41: Repetition performance in both random token and natural text settings for 2.8B models. HAPAX models struggle with repetition, which is solvable with induction circuits. This shows the HAPAX training can control copying behavior. Accuracy is measured over 1000 randomly generated samples with $s = 25$.

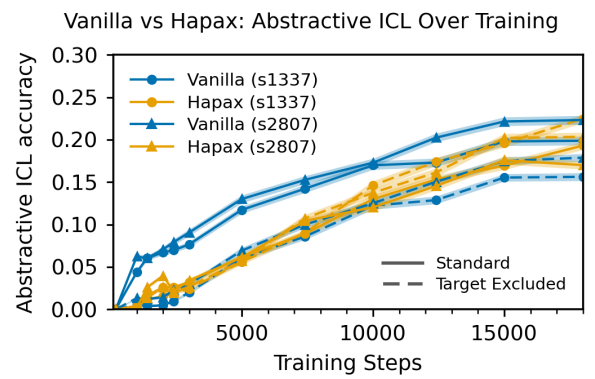


Figure H42: Comparison of abstractive-icl accuracy plots across training for seeds 1337 and 2807.

Table H11: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX 2.8B. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
AG News	42.0 $\pm$ 1.4†	23.7 $\pm$ 1.2	Antonym	2.3 $\pm$ 0.6	7.9 $\pm$ 1.1†
Capitalize (Full Word)	72.8 $\pm$ 3.1	81.3 $\pm$ 2.7†	Capitalize First Letter	40.3 $\pm$ 3.4	60.8 $\pm$ 3.4†
Capitalize Last Letter	7.2 $\pm$ 1.8†	4.8 $\pm$ 1.5	Capitalize Second Letter	8.3 $\pm$ 1.9†	4.1 $\pm$ 1.4
Chinese $\rightarrow$ English	11.0 $\pm$ 2.0	14.1 $\pm$ 2.2†	CommonsenseQA	19.9 $\pm$ 1.1†	17.0 $\pm$ 1.0
Country-Capital	48.7 $\pm$ 7.1	61.4 $\pm$ 6.9†	Country-Currency	12.9 $\pm$ 6.8	21.5 $\pm$ 8.4
French $\rightarrow$ English	19.2 $\pm$ 2.5	25.1 $\pm$ 2.7†	Indonesian $\rightarrow$ English	1.7 $\pm$ 0.8	2.3 $\pm$ 0.9
Italian $\rightarrow$ English	10.8 $\pm$ 1.9	14.7 $\pm$ 2.2†	Japanese $\rightarrow$ English	2.9 $\pm$ 1.0	4.7 $\pm$ 1.3†
Landmark-Country	31.3 $\pm$ 3.1	40.0 $\pm$ 3.3†	Lowercase First Letter	55.0 $\pm$ 3.4	74.2 $\pm$ 3.0†
Lowercase Last Letter	8.2 $\pm$ 1.9†	5.5 $\pm$ 1.6	National Parks	24.6 $\pm$ 4.0	31.3 $\pm$ 4.3†
Next Capital Letter	4.2 $\pm$ 1.4	4.5 $\pm$ 1.4	Next Item	14.7 $\pm$ 4.6	26.2 $\pm$ 5.7†
Park-Country	10.7 $\pm$ 2.2	19.7 $\pm$ 2.9†	Person-Instrument	21.0 $\pm$ 3.5†	1.6 $\pm$ 1.1
Person-Occupation	17.5 $\pm$ 2.6†	3.0 $\pm$ 1.2	Person-Sport	28.3 $\pm$ 5.0	37.1 $\pm$ 5.3†
Polish $\rightarrow$ English	4.3 $\pm$ 1.3	5.9 $\pm$ 1.5†	Portuguese $\rightarrow$ English	11.2 $\pm$ 2.0	17.0 $\pm$ 2.4†
Present-Past	67.9 $\pm$ 5.3	71.3 $\pm$ 5.2	Previous Item	1.8 $\pm$ 1.7	5.3 $\pm$ 2.9
Product-Company	23.9 $\pm$ 3.7	26.4 $\pm$ 3.8	Sentiment	61.9 $\pm$ 2.8†	28.8 $\pm$ 2.6
Singular-Plural	71.2 $\pm$ 6.2	82.9 $\pm$ 5.2†	Spanish $\rightarrow$ English	15.7 $\pm$ 2.3	22.4 $\pm$ 2.6†
Synonym	3.0 $\pm$ 0.6	4.9 $\pm$ 0.8†	Word Length	4.3 $\pm$ 1.4	4.4 $\pm$ 1.4

Table H12: Performance comparison on **abstractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX 2.8B by ensuring that the target answer for each test instance is not included in the 5-shot examples. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
AG News	3.4 $\pm$ 0.5	35.9 $\pm$ 1.3†	Antonym	2.4 $\pm$ 0.6	8.2 $\pm$ 1.1†
Capitalize (Full Word)	72.8 $\pm$ 3.1	81.3 $\pm$ 2.7†	Capitalize First Letter	35.9 $\pm$ 3.3	71.6 $\pm$ 3.1†
Capitalize Last Letter	4.9 $\pm$ 1.5	8.1 $\pm$ 1.9†	Capitalize Second Letter	1.9 $\pm$ 1.0	6.4 $\pm$ 1.7†
Chinese $\rightarrow$ English	11.0 $\pm$ 2.0	14.1 $\pm$ 2.2†	CommonsenseQA	14.0 $\pm$ 1.0	44.3 $\pm$ 1.4†
Country-Capital	48.7 $\pm$ 7.1	61.4 $\pm$ 6.9†	Country-Currency	14.0 $\pm$ 7.0	23.7 $\pm$ 8.6
French $\rightarrow$ English	19.2 $\pm$ 2.5	25.1 $\pm$ 2.7†	Indonesian $\rightarrow$ English	1.7 $\pm$ 0.8	2.3 $\pm$ 0.9
Italian $\rightarrow$ English	10.8 $\pm$ 1.9	14.7 $\pm$ 2.2†	Japanese $\rightarrow$ English	2.9 $\pm$ 1.0	4.7 $\pm$ 1.3†
Landmark-Country	30.0 $\pm$ 3.1	40.9 $\pm$ 3.3†	Lowercase First Letter	57.6 $\pm$ 3.4	87.6 $\pm$ 2.3†
Lowercase Last Letter	5.3 $\pm$ 1.5	9.5 $\pm$ 2.0†	National Parks	22.4 $\pm$ 3.8	33.3 $\pm$ 4.3†
Next Capital Letter	3.8 $\pm$ 1.3	4.8 $\pm$ 1.5	Next Item	14.7 $\pm$ 4.6	26.2 $\pm$ 5.7†
Park-Country	10.8 $\pm$ 2.2	19.0 $\pm$ 2.8†	Person-Instrument	2.4 $\pm$ 1.3	5.3 $\pm$ 1.9†
Person-Occupation	2.8 $\pm$ 1.1	2.6 $\pm$ 1.1	Person-Sport	20.8 $\pm$ 4.5	56.6 $\pm$ 5.4†
Polish $\rightarrow$ English	4.3 $\pm$ 1.3	5.9 $\pm$ 1.5†	Portuguese $\rightarrow$ English	11.2 $\pm$ 2.0	17.0 $\pm$ 2.4†
Present-Past	67.9 $\pm$ 5.3	71.3 $\pm$ 5.2	Previous Item	0.4 $\pm$ 0.7	4.4 $\pm$ 2.7†
Product-Company	14.9 $\pm$ 3.1	38.5 $\pm$ 4.2†	Sentiment	20.6 $\pm$ 2.3	86.7 $\pm$ 1.9†
Singular-Plural	71.2 $\pm$ 6.2	82.9 $\pm$ 5.2†	Spanish $\rightarrow$ English	15.7 $\pm$ 2.3	22.4 $\pm$ 2.6†
Synonym	3.0 $\pm$ 0.6	5.1 $\pm$ 0.8†	Word Length	3.6 $\pm$ 1.3	8.8 $\pm$ 2.0†

H 2.8B Evaluations

We train additional Vanilla and HAPAX models with 2.8B parameters to evaluate whether the preservation of abstractive capabilities is also observed at a larger model scale. The models are trained on 52.4B tokens; the compute-optimal budget estimated by Hoffmann et al. (2022) is approximately 56B tokens. Due to compute constraints, we omit the Thresholded-HAPAX variant from the 2.8B experiments. We use the same setting as in our 1B training, but train for 25K steps. Under standard evaluation, Vanilla achieves 23.0% accuracy, whereas HAPAX achieves 25.2% (Table H10). These results suggest that the preservation of abstractive capabilities under HAPAX is also observed in this 2.8B setting, despite reduced copying and a

degradation of the TLD metric (Figures H40–H44).

I Responsible NLP Checklist Information

I.1 Statistics for Data

We discuss relevant dataset statistics in Section 3.1 and Appendix D. Furthermore, we used a 80%/20% split for the training and validation data.

I.2 Model Size and Budget

We report the model sizes and training setup in Section 3.1. Each 1B model was trained on 8 A100 GPUs for 72 hours, corresponding to approximately 576 A100 GPU-hours per run. Each 2.8B model was trained on 8 A100 GPUs for 168 hours, corresponding to approximately 1344 A100 GPU-hours per run.

Table H13: Performance comparison on **extractive** in-context learning tasks (5-Shot) for Vanilla and HAPAX models. Values show accuracy $\pm$ 95% CI margin. Bold indicates higher performance; $\dagger$ denotes statistical significance ($p < 0.05$) according to McNemar’s test.

Task	Vanilla (%)	HAPAX (%)	Task	Vanilla (%)	HAPAX (%)
Color vs Animal (3)	56.6 $\pm$ 3.1†	48.1 $\pm$ 3.1	CoNLL: Location	48.0 $\pm$ 1.5†	35.1 $\pm$ 1.4
Concept vs Object (3)	76.7 $\pm$ 2.6†	55.4 $\pm$ 3.1	CoNLL: Person	58.3 $\pm$ 1.8†	12.1 $\pm$ 1.2
Choose Middle of 3	40.8 $\pm$ 3.0†	25.7 $\pm$ 2.7	Alphabetically Last (3)	31.5 $\pm$ 2.9†	25.0 $\pm$ 2.7
Alphabetically Last (5)	19.2 $\pm$ 2.4†	14.9 $\pm$ 2.2	Object vs Concept (3)	58.0 $\pm$ 3.1†	37.6 $\pm$ 3.0
Object vs Concept (5)	57.6 $\pm$ 3.1†	31.5 $\pm$ 2.9	Fruit vs Animal (5)	22.1 $\pm$ 2.6†	7.8 $\pm$ 1.7
Animal vs Object (3)	50.5 $\pm$ 3.1†	34.8 $\pm$ 3.0	Verb vs Adjective (3)	77.7 $\pm$ 2.6†	64.4 $\pm$ 3.0
Fruit vs Animal (3)	37.4 $\pm$ 3.0†	14.3 $\pm$ 2.2	Choose First of 5	77.1 $\pm$ 2.6†	42.0 $\pm$ 3.1
Choose First of 3	72.3 $\pm$ 2.8†	32.6 $\pm$ 2.9	Choose Last of 3	63.7 $\pm$ 3.0†	43.0 $\pm$ 3.1
Color vs Animal (5)	49.4 $\pm$ 3.1†	41.5 $\pm$ 3.1	Verb vs Adjective (5)	68.1 $\pm$ 2.9†	53.8 $\pm$ 3.1
SQuAD Validation	17.1 $\pm$ 1.1†	10.6 $\pm$ 0.9	CoNLL: Organization	47.5 $\pm$ 1.7†	17.0 $\pm$ 1.3
Choose Last of 5	71.1 $\pm$ 2.8†	45.6 $\pm$ 3.1	Adjective vs Verb (3)	54.6 $\pm$ 3.1†	27.6 $\pm$ 2.8
Alphabetically First (3)	32.5 $\pm$ 2.9†	23.4 $\pm$ 2.6	Animal vs Object (5)	38.9 $\pm$ 3.0†	25.4 $\pm$ 2.7
Concept vs Object (5)	65.6 $\pm$ 2.9†	47.8 $\pm$ 3.1	Choose Middle of 5	20.8 $\pm$ 2.5†	12.5 $\pm$ 2.0
Adjective vs Verb (5)	43.4 $\pm$ 3.1†	18.1 $\pm$ 2.4	Alphabetically First (5)	20.9 $\pm$ 2.5†	14.3 $\pm$ 2.2

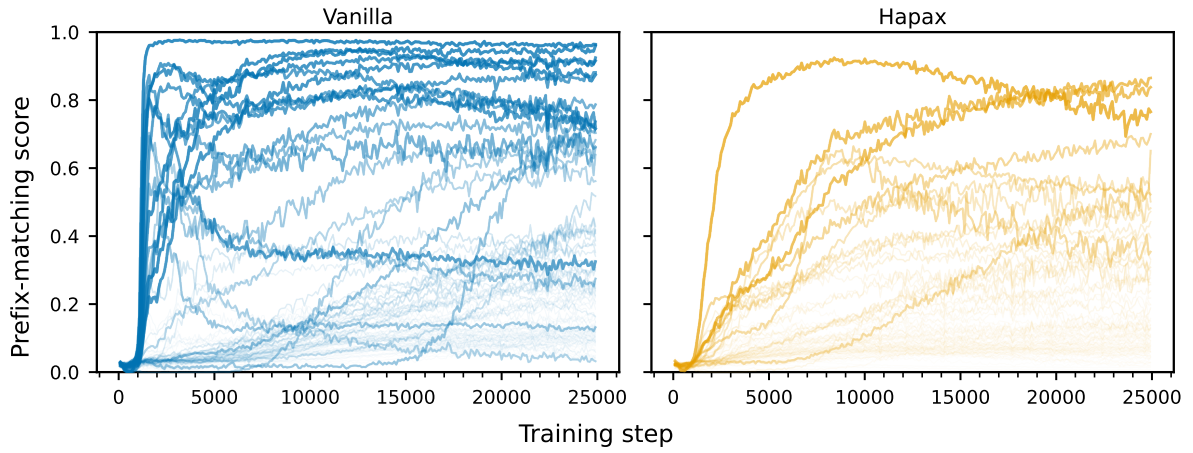


Figure H43: Prefix-matching scores of attention heads across training for 2.8B. Each line is one head. Opacity is proportional to peak score attained throughout training (heads with peak < 0.1 omitted).

I.3 Artifact Use & Licensing

The Pile (Gao et al., 2020) and the Function Vectors tasks (Todd et al., 2024) are subject to the MIT License. In accordance with the licensing terms, these are solely used for research purposes.

I.4 AI Usage

In accordance with the author guidelines, we made use of LLM-based services such as Claude as assistive tools for correcting grammar and refine writing.

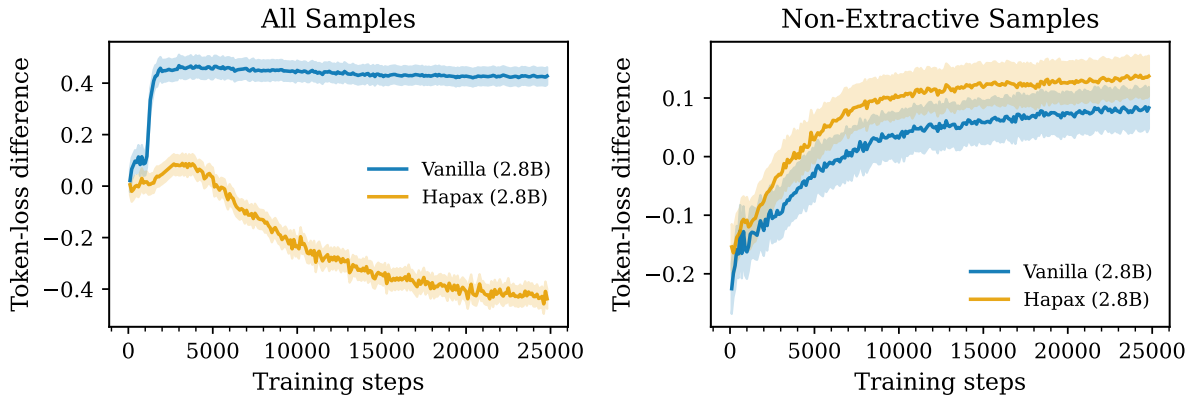


Figure H44: Comparison of token-loss metrics across training on 2.8B models. Unlike the regular (All Samples) TLD, our proposed variants on non-extractive samples progress gradually, with each HAPAX variant being able to achieve non-trivial ICL capabilities on its corresponding distribution. Since the regular distribution is dominated by induction-predictable positions, its gains primarily reflect exact copying rather than abstractive ICL. Computing the metric on targeted subsets thus reveals distinct emergence patterns across the distributions of interest. Metrics are computed on randomly sampled validation data.