

---

# Beyond Induction Heads: In-Context Meta Learning Induces Multi-Phase Circuit Emergence

---

Gouki Minegishi<sup>1</sup> Hiroki Furuta<sup>1</sup> Shohei Taniguchi<sup>1</sup> Yusuke Iwasawa<sup>1</sup> Yutaka Matsuo<sup>1</sup>

## Abstract

Transformer-based language models exhibit In-Context Learning (ICL), where predictions are made adaptively based on context. While prior work links induction heads to ICL through a sudden jump in accuracy, this can only account for ICL when the answer is included within the context. However, an important property of practical ICL in large language models is the ability to meta-learn how to solve tasks from context, rather than just copying answers from context; how such an ability is obtained during training is largely unexplored. In this paper, we experimentally clarify how such meta-learning ability is acquired by analyzing the dynamics of the model’s circuit during training. Specifically, we extend the copy task from previous research into an In-Context Meta Learning setting, where models must infer a task from examples to answer queries. Interestingly, in this setting, we find that there are multiple phases in the process of acquiring such abilities, and that a unique circuit emerges in each phase, contrasting with the single-phases change in induction heads. The emergence of such circuits can be related to several phenomena known in large language models, and our analysis lead to a deeper understanding of the source of the transformer’s ICL ability.

## 1. Introduction

Transformer-based language models (Vaswani et al., 2017) show an intriguing ability to perform In-Context Learning (ICL) (Brown et al., 2020; Xie et al., 2021; Garg et al., 2022; Dong et al., 2024). ICL is the ability to predict the response to a query based on context without any additional weight updates. A widely adopted application of ICL is *few-shot*

*learning* in which only a small number of examples in the context guide the model’s response to a new query. Due to its unique capability, ICL has gained a lot of attention in the research community, and there have been several approaches such as Bayesian inference (Xie et al., 2021) and meta-gradient descent (Von Oswald et al., 2023) to uncover its underlying mechanisms.

One of the popular approaches to understanding ICL is mechanistic interpretability; reverse-engineering the computations performed by models (Elhage et al., 2021). A key focus within this framework is the study of *circuits*, sub-graphs with distinct functionality that serve as fundamental building blocks of neural network behavior (Wang et al., 2022; Conmy et al., 2023a). Notably, Olsson et al. (2022) uncovered *induction heads*, a specific circuit mechanism that plays a crucial role in enabling ICL. Induction heads recognize the repeating pattern [A] [B] ... [A] within the context and predict [B] as the next token through a match-and-copy operation (Figure 1-(a)). The existence of induction heads is further investigated under more complex tasks, such as performing semantic matching (Ren et al., 2024), serving as subcomponents of circuits for natural language tasks within LLMs (Wang et al., 2022; Merullo et al., 2024), and engaging in intricate interactions with multi-head attention (Singh et al., 2024).

However, the copy mechanism as described in the induction head explains only a fraction of the few-shot ICL. Let us consider, for instance, the following ICL scenario in a Country-to-Capital task, based on Hendel et al. (2023):

France → Paris, Spain → Madrid, Japan → ?  
example query prediction

It is well known that ICL can enhance performance in this scenario however, this improvement cannot be explained merely by retrieving similar examples through induction heads. A straightforward way to explain this ability is to assume that the model infers the task from the examples and then uses this inferred task to make predictions. For example, Hendel et al. (2023); Todd et al. (2024) demonstrates that tasks are internally represented as vectors (i.e., task vectors) within the LLM. This task inference ability is

---

<sup>\*</sup>Equal contribution <sup>1</sup>The University of Tokyo. Correspondence to: Gouki Minegishi <minegishi@weblab.t.u-tokyo.ac.jp>.

recognized as a form of meta-learning (Min et al., 2022a). However, it remains unclear exactly what kind of circuit implements this meta-learning or how the circuit is acquired.

In this study, our goal is to elucidate how such meta-learning capability is acquired. To that end, we extend the copy task from previous research (Reddy, 2023) to a problem setting, which we call In-Context Meta-Learning (ICML) setting, that requires task inference. We then train a simplified transformer on this extended setting, and analyze changes in its internal circuits during the training process. In this setting, as shown in Figure 1-(a), there exists a set of multiple tasks, and the answers differ from task to task, so the model needs to infer the task from the examples to answer the query. Interestingly, we observe that learning dynamics emerge in this setting that differ significantly from the case of simple copying tasks. First, we find that the model undergoes learning phase while acquiring meta-learning capabilities, unlike the single phase typically observed in copying tasks. More specifically, we find that in the first phase, a bigram-type circuit emerges that focuses solely on the query, ignoring the context and relying only on the model’s weights. In the second phase, a circuit emerges that pays attention only to the labels in the context. Finally, a circuit emerges that chunks each example pair into a single token.

We introduce novel metrics to measure these three circuits and show that the abrupt change of these metrics aligns closely with the sudden jumps in accuracy. Notably, the label-focused circuit that emerges in the second phase suggests that during acquiring meta-learning capabilities, the model may initially learn to identify tasks by examining only the set of labels, without considering the correspondence between classes and labels. The existence of the label-focused circuit also corresponds to the phenomenon in previous studies (Min et al., 2022b) that LLMs maintain high ICL performance even under random label assignments, which is one explanation for the unique nature of LLMs.

We also examine the case of a multi-head model, which is a more practical setting; sudden jumps in accuracy become less apparent, and different heads can still specialize in *parallel* — for instance, one head may converge on a particular circuit, while another becomes a different one. Although this parallel specialization leads to smoother accuracy improvements, our circuit-level metrics uncover hidden circuit emergence, revealing that even though learning phases remain invisible in the accuracy curve, the underlying circuits still change abruptly. This observation suggests that even when a clear phase changes are not observed on the loss curve, as in the case of LLM training, abrupt changes can occur on the circuits, which leads to bridging the gap between toy experiments in the study of mechanistic interpretability and practical scenarios.

## 2. Related Works

### 2.1. In-Context Learning

Brown et al. (2020) demonstrated with GPT-3 the remarkable ability of LLMs to perform a wide range of tasks using only a few examples provided in the input prompt. Few-shot ICL is the ability of LLMs to solve new tasks by examining a sequence of (input, label) pairs that share a common concept within the context. Rather than updating their internal parameters, these models rely solely on the contextual examples to deduce the task’s rules.

In general, ability to learn from few-shot examples is associated with *meta-learning* (Wang et al., 2020; Hospedales et al., 2021), and success of the ICL demonstrate the strong ability of LLM to meta-learn. In effective ICL, the model infers the underlying task from the examples provided and refines its predictions based on the inferred task. Although this meta-learning-based ability is widely used, the underlying mechanisms enabling LLMs to perform these tasks remain poorly understood, and some puzzling results have been observed. For example, Min et al. (2022b) demonstrated that even when the labels in the examples are randomized, the accuracy improves. Additionally, Chan et al. (2022) have demonstrated that data distributional properties significantly influence ICL performance.

To understand ICL, various approaches have been proposed. For example, Von Oswald et al. (2023); Dai et al. (2023) demonstrated that transformers can solve linear regression problems within the context by leveraging meta-gradients. Based on this, analytical methods have been applied to study the ability of transformers to handle a range of tasks, including discrete functions (Bhattamishra et al., 2023), nonlinear functions (Kim & Suzuki, 2024), and classification problems (von Oswald et al., 2023).

### 2.2. Mechanistic Interpretability

One promising approach to understanding ICL is *mechanistic interpretability* (MI), which seeks to uncover the internal mechanisms of models (Olah et al., 2020; Elhage et al., 2021). A key focus of MI is the study of *circuits*, which are subgraphs with distinct functionality that serve as fundamental building blocks of neural network behavior (Wang et al., 2022; Conmy et al., 2023b; Merullo et al., 2024).

One such circuit studied in the context of ICL is the *induction head* (Olsson et al., 2022). The induction heads are a two-layer structure; in particular, the latter layer is commonly called the induction head, and the earlier layer is referred to as the previous token head. Previous token head attends to and copies the preceding token into the current token. When few-shot examples are present in the context, it chunks each  $(x, \ell)$  pair into a single token. Induction heads then perform a match-and-copy operation, matching a query

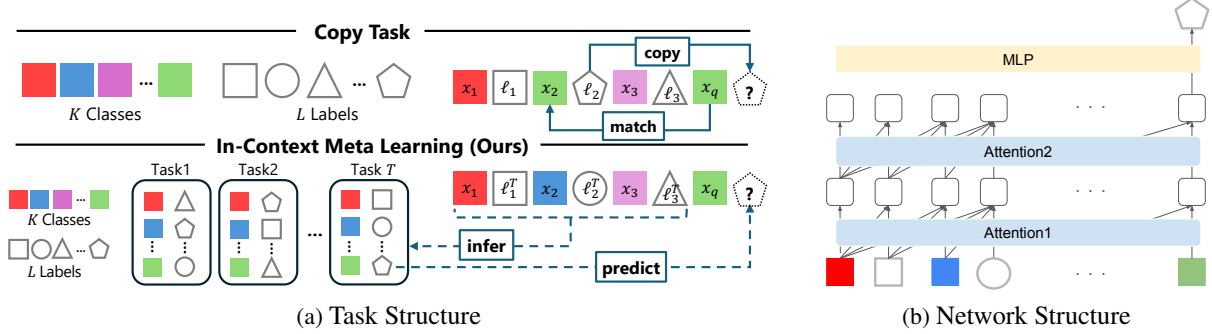


Figure 1. (a) **Task Structure**: Previous studies focused on a copying-task setup, where the query’s answer remains unchanged by context, allowing the model to either memorize pairs or *match* and *copy* from context. In contrast, this work explores a more practical scenario where  $(x, \ell)$  pairs vary by task, requiring the model to *infer* the task from examples and *predict* the query’s answer. (b) **Network Structure**: we mainly use two layers of attention followed by a token-wise MLP layer. The task is consistent within the context.

derived from the current token with a key derived from the previous token head’s output. For more details on the induction head, see Appendix A. Further research has shown that induction heads can perform soft matching (Crosbie & Shutova, 2024), emerge naturally in multi-head attention settings (Singh et al., 2024), and are present in LLMs (Cho et al., 2024).

Despite these advancements of induction heads, these studies have primarily focused on tasks where the context explicitly includes the label to be copied, such as direct copying tasks. Therefore, induction heads alone cannot fully explain the meta-learning capabilities in more practical scenarios.

### 3. Experimental Setup

#### 3.1. In-Context Meta Learning

To analyze the meta-learning capabilities of ICL, building on prior works (Chan et al., 2022; Reddy, 2023), we design a simple experimental setting named the In-Context Meta-Learning (ICML<sup>1</sup>) described in Figure 1-(a). Unlike previous approaches, where copying labels or memorizing  $(x, \ell)$  pairs was sufficient to predict the answer, our setting instead requires the model to meta-learn the underlying task  $(\tau)$  from  $(x, \ell)$  context pairs. The network is trained to predict the label of a target  $x_q$  given an alternating sequence of  $N$  items and  $N$  labels:

$$\underbrace{x_1, \ell_1^T, x_2, \ell_2^T, \dots, x_N, \ell_N^T}_{\text{examples}}, \underbrace{x_q}_{\text{query}}, \underbrace{?}_{\text{prediction}}$$

Here,  $\tau$  represents the task, where each task defines a unique  $(x, \ell)$  pair with labels  $\ell$  randomly assigned to items  $x$ . The total number of tasks is denoted as  $T$ , and the context presented to the model consistently corresponds to the same task. Since the query  $x_q$  may not be appeared in the in-context

examples, the network needs to infer the task  $\tau$ , instead of simply copying a label, from the context.

Following Reddy (2023), we represent each item  $x$  and label  $\ell$  in a  $(P + D)$ -dimensional space. Of these dimensions,  $P$  is dedicated to positional information via a one-hot encoding (with  $P = 65$  across all experiments), while  $D$  captures the content. To encourage translation-invariant operations, each input sequence is randomly placed within a window of size  $(2N + 1)$  spanning the range  $[0, P - 1]$ . Each class  $k$  is associated with a  $D$ -dimensional mean vector  $\mu_k$ , whose entries are drawn independently from  $\mathcal{N}(0, 1/D)$ . For an item  $x_i$  assigned to class  $k$ , we add noise  $\eta$  (sampled from the same distribution) scaled by  $\epsilon$ , giving  $x_i = \frac{\mu_k + \epsilon \eta}{\sqrt{1 + \epsilon^2}}$ , where  $\epsilon$  governs within-class variation and the denominator ensures  $\|x_i\| \approx 1$ . Finally, each class is linked to one of  $L$  labels, with  $L \leq K$ . To control the proportion to which a query can be solved by copying from the context, the same item as the query is included in the context with probability  $p_B$ . We use  $T = 3$ ,  $K = 64$ ,  $L = 32$ ,  $N = 4$ ,  $D = 63$ ,  $\epsilon = 0.1$ ,  $p_B = 0$ , unless otherwise specified. In our ICML setup, we can reproduce the standard match-and-copy induction head mechanism from Reddy (2023) by setting  $T = 1$ ,  $p_B = 1$ . For detailed results, see Appendix A.

#### 3.2. Network Structure

Following prior research (Reddy, 2023), we use a two-layer attention-only transformer shown in Figure 1-(b), where each layer  $\mu$  comprises  $m$  heads (indexed by  $h$ ), and a causal mask ensures position  $i$  attends only to positions  $j \leq i$ . A two-layer MLP classifier then produces the label probabilities. For the complete set of equations and hyperparameter details, see Appendix B. In this architecture, each head  $h$  in layer  $\mu$  computes attention weights  $\{p_{ij}^{(\mu, h)}\}$ , quantifying how strongly position  $i$  (query) attends to position  $j$  (key). These outputs are aggregated across heads and passed to the MLP, which makes the final label predictions.

<sup>1</sup>Code is available at <https://github.com/gouki510/In-Context-Meta-Learning>

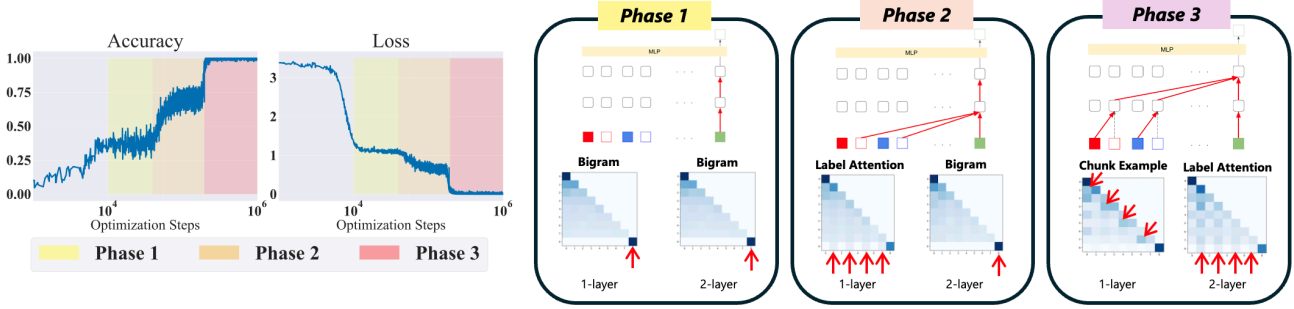


Figure 2. **(left)** Changes in accuracy and loss across three distinct phases during training, with lighter-shaded curves indicating different random seeds. Each phase is highlighted with a different background color: Phase 1 (yellow), Phase 2 (orange), and Phase 3 (red). **(right)** Visualization of the attention maps (circuits) corresponding to each phase, with characteristic attention patterns indicated by red arrows and their circuits displayed above. Specific attention types, such as Bigram, Label Attention, and Chunk Example, emerge at different phases, reflecting the model’s adaptation to the task. Quantitative results for these attention maps are provided in Figure 4.

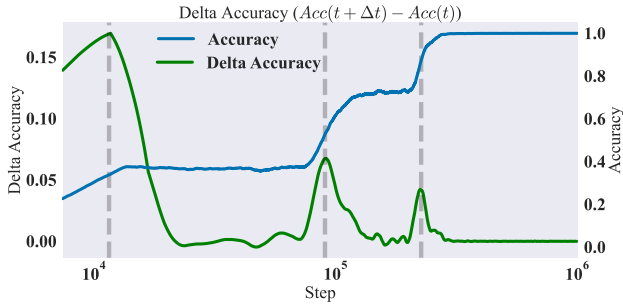


Figure 3. Accuracy (blue) and  $\Delta \text{Accuracy}$  (green) as functions of the training step. Here,  $\Delta \text{Accuracy} = \text{Accuracy}(t + \Delta t) - \text{Accuracy}(t)$  with  $\Delta t = 100$ . Vertical dashed lines indicate where  $\Delta \text{Accuracy}$  exceeds 0.025, marking the transition points between the three observed phases (Phase 1, Phase 2, Phase 3).

The classifier is a two-layer MLP with ReLU activations, followed by a softmax layer producing probabilities over  $L$  labels. We train this network to classify the query item  $x_q$  into one of the  $L$  labels using cross-entropy loss. Both the query/key dimension and the MLP hidden layer dimension are set to 128. We use a batch size of 128 and optimize with vanilla stochastic gradient descent at a learning rate of 0.01.

## 4. Abrupt Learning and Emergent Circuits

### 4.1. Three-Phase Dynamics and Circuit Overview

We conducted experiments under the ICML setting with three tasks (i.e.,  $T = 3$ ). As shown on the left side of Figure 2, the results reveal three distinct phases of accuracy changes, each accompanied by a corresponding drop in loss. The observed dynamics are as follows: the first accuracy plateau occurs at around 30–40%, the second at approximately 75%, and the final phase reaches 100%. To clearly these three phases, we define the following metric:

$$\Delta \text{Accuracy} = \text{Accuracy}(t + \Delta t) - \text{Accuracy}(t),$$

where  $t$  denotes the optimization step and we set  $\Delta t = 100$ . In Figure 3, we plot this quantity along with the model’s accuracy, marking vertical lines at steps where  $\Delta \text{Accuracy} > 0.025$ . These lines serve as boundaries between the three observed phases. Based on this threshold, we partition the model’s behavior into Phase 1, Phase 2, and Phase 3 throughout the remainder of this paper.

On the right side of Figure 2, we visualize the attention maps from the two layers of the model during each phase. The attention patterns emerging during the learning process can be categorized into the following three types:

1. **Bigram:** Strong attention is focused on the token in the context that corresponds to the query token ( $x_q$ ).
2. **Label Attention:** Strong attention is focused on the label tokens of the  $(x, \ell)$  pair within the context.
3. **Chunk Example:** Attention aggregates the  $(x, \ell)$  token pair in the context into a single token, similar to the induction head’s previous token head.

As visualized on the right side of Figure 2, the combinations of these attention types differ between the first and second layers across the three phases:

**Phase 1 (Non-Context Circuit; NCC):** Both layers use bigram attention, *ignoring the context* and relying solely on the model’s weights. At this stage, the model predict label base on only query, limiting accuracy to around  $1/T$ . In this case, the accuracy stagnates at around 30–40%.

**Phase 2 (Semi-Context Circuit; SCC):** The first layer exhibits label attention, while the second layer focuses on the query token (bigram attention). The model not only leverages weights memory but also attends to label tokens (i.e., *half of the context*), in the context to infer possible answers, resulting in improved accuracy of around 75%.

**Phase 3 (Full-Context Circuit; FCC):** The first layer aggregates the  $(x, \ell)$  pair into a single token (chunk example),



Table 1. Summary of circuits, accuracy, and layer-wise attention.

Circuit	Accuracy ( $T = 3$ )	Layer 1	Layer 2
NCC	30–40%	Bigram	Bigram
SCC	$\approx 75\%$	Label Attention	Bigram
FCC	100%	Chunk Example	Label Attention

Table 2. Formulas of the three attention metrics.

Metric	Formula
Bigram	$p_{2N+1, 2N+1}^{\mu, h}$
Label Attention	$\sum_{k=1}^N p_{2N+1, 2k}^{\mu, h}$
Chunk Example	$\frac{1}{N} \sum_{k=1}^N p_{2k, 2k-1}^{\mu, h}$

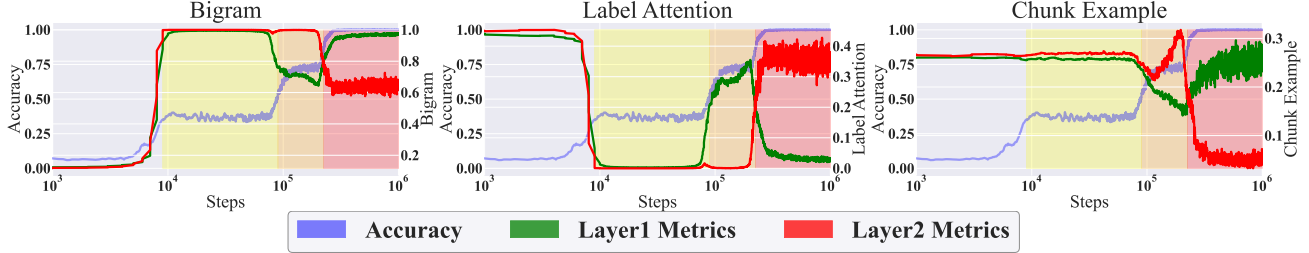


Figure 4. Evolution of the three attention metrics (Bigram, Label Attention, and Chunk Example) across optimization steps for the first (green) and second (red) layers. The shaded regions represent the three learning phases: Phase 1 (yellow), Phase 2 (orange), and Phase 3 (red), defined by  $\Delta$ Accuracy (Figure 3). Each metric shifts cleanly at the phase boundaries, demonstrating a close correspondence between accuracy improvements and circuit-level transformations.

while the second layer focuses on these aggregated tokens (label attention) to predict label, resulting in *using the entire context*. Through this abstraction of the pairwise relationship (i.e., task inference), the model can produce correct answers for the query. Once the model learns this circuit, it achieves 100% accuracy.

The relationship between each circuit and its corresponding attention pattern is summarized in Table 1.

## 4.2. Quantifying Circuit Emergence

To quantitatively measure these circuits, we propose three metrics based on the attention maps of each layer. Let  $p_{i,j}^{\mu,h}$  represent the attention from token  $j$  to token  $i$  in the  $h$ -th head of the  $\mu$ -th layer. Let the context length be  $2N + 1$  (in this case,  $N = 4$ ). We define three primary attention-based metrics, with precise formulas provided in Table 2. Here, we briefly describe what each metric represents: (1) *Bigram Metrics* capture the attention from the query token to itself; (2) *Label Attention Metrics* measure the total attention from the query token to the label tokens within the context; (3) *Chunk Example Metrics* assess the attention from  $x$  to  $\ell$  within each  $(x, \ell)$  pair.

The plots in Figure 4 illustrate how these metrics evolve in the first and second layers across the three phases. For the Bigram Metrics, both the first and second layers show high values at the moment of the initial jump in accuracy, marking the formation of the NCC. Then, at the beginning of Phase 2, the bigram metrics in the first layer decrease significantly while those in the second layer remain high, and label attention in the first layer rises — together leading to the formation of the SCC. At the start of Phase 3,

the chunk example metrics in the first layer increase, and the label attention metrics in the second layer also become high, resulting in the formation of the FCC. Importantly, these metric transitions align closely with the corresponding jumps in model accuracy, supporting the view that these metrics provide a valid and quantitative perspective on the circuit changes observed during the three phases, as depicted in the right side of Figure 2.

## 4.3. Deeper Look at the Semi-Context Circuit

**How SCC Drives Accuracy** In Phase 2, the model forms the SCC, using label information from the context in addition to the query. We provide a theoretical analysis of why this leads to improved accuracy and empirically validate our theory through controlled experiments. To clarify SCC’s behavior, we tested the following simplified conditions:

1. The number of classes ( $K$ ) equals the number of labels ( $L$ ), with no duplication.
2. The input context (including the query) contains no duplicate classes.
3. The number of tasks ( $T$ ) is set to 2, and there are no common  $(x, \ell)$  pairs shared across tasks.
4. To specifically focus on SCC, a mask is applied to circuits associated with SCC during training (details are provided in the Appendix C).

In Phase 1, since there are two tasks, the model has a 50% chance of predicting correctly by random guessing. In other words, the model’s prediction reduces to a binary choice for each input query ( $x_q$ ). Once label information becomes usable, the binary choice can potentially be narrowed fur-

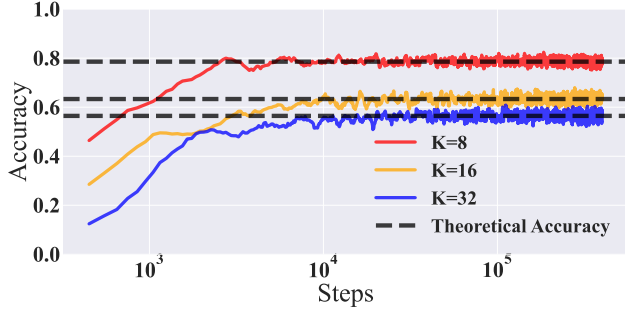


Figure 5. Comparison of theoretical accuracy (dashed lines) and model accuracy for different class counts ( $K$ ). The close alignment between theoretical predictions and experimental results confirms the validity of the theoretical analysis.

ther. This occurs when one of the labels corresponding to the two options is present in the context. In this scenario, the label in the context is definitively not the correct answer for the query, as per the defined conditions. Thus, the answer becomes uniquely determinable, increasing accuracy. Following the derivation in Appendix D, the probability of one of the labels appearing in the context is

$$p = 1 - \frac{\binom{K-2}{4}}{\binom{K-1}{4}}.$$

Therefore, the theoretical accuracy achievable with SCC can be expressed as:

$$\text{Theoretical Accuracy} = p \cdot 1 + (1 - p) \cdot 0.5.$$

Figure 5 shows the theoretical accuracy alongside the accuracy achieved by a model trained with only Phase 2 attention circuits remaining. The class/label counts were varied as  $K = \{8, 16, 32\}$ . The near-perfect agreement between the theoretical and empirical results confirms both the validity of our derivation and the role of SCC in boosting accuracy.

**Random-Label Robustness of SCC** We focus on the tendency of SCC to make predictions “based solely on labels and query.” We hypothesize that this circuit explains the puzzling phenomenon that the improvement in ICL performance observed even when using random labels, as noted in Min et al. (2022b). Min et al. (2022b) has demonstrated that replacing labels randomly within examples results in only a marginal performance drop, suggesting that ICL does not rely heavily on  $(x, \ell)$  pairs. To investigate this phenomenon, we define an Out-of-Distribution (OOD) evaluation where the labels in each context pair are randomly permuted. Specifically, we consider:

$$\underbrace{x_1, \ell_{\pi(1)}^T, x_2, \ell_{\pi(2)}^T, \dots, x_N, \ell_{\pi(N)}^T}_{\text{examples}}, \underbrace{x_q}_{\text{query}}, \underbrace{?}_{\text{prediction}}$$

Here,  $\pi$  is a random permutation on  $\{1, 2, \dots, N\}$ , meaning that  $\ell_{\pi(i)}^T$  replaces the original label  $\ell_i^T$ . By measuring the

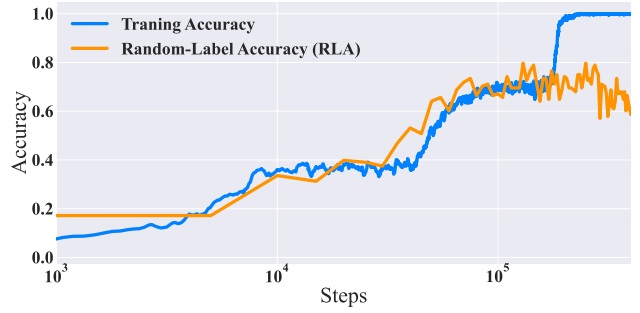


Figure 6. Comparison of training accuracy and random-label accuracy (RLA). The plot demonstrates the rise in both metrics, with RLA following a trend similar to emergence Phase 2. This indicates that SCC acquired in Phase 2 contributes to improved accuracy even with shuffled labels.

model’s accuracy under these shuffled labels, we obtain the *Random-Label-Accuracy (RLA)*. In the Figure 6, we compare this RLA with the training accuracy. Similar to the rise observed in Phase 2, when SCC is acquired, the RLA also increases. This suggests that the reason for the improved performance with random labels, as seen in Min et al. (2022b), is the existence of circuits similar to SCC within LLMs.

#### 4.4. Effects of Data Property on Circuits Emergence

Previous studies have indicated that certain properties of the training data, such as burstiness, can influence the emergence of ICL (Chan et al., 2022) and induction heads (Reddy, 2023). In this work, we explore how these data properties affect the development of circuits in our ICML setting, with the aim of advancing our understanding of the multi-phase emergence of these circuits. As mentioned in Section 3, the variables capturing the characteristics of the data include the number of tasks  $T$ , the number of classes  $K$ , the noise magnitude  $\epsilon$ . In addition, and following Chan et al. (2022), we adopt rank-frequency distributions over both classes and tasks:  $f(k) \sim k^{-\alpha}$  and  $f(\tau) \sim \tau^{-\beta}$ , which follow a power-law form commonly known as *Zipf’s law* (Zipf, 1949) (see Appendix E for details). The default values are  $T = 3$ ,  $K = 64$ ,  $\epsilon = 0.1$ ,  $\alpha = 0$ , and  $\beta = 0$ . The results of varying these parameters are shown in the Figure 7. For results obtained by varying  $p_B$ , see Appendix F.

In Figure 7-(a), we present the results of varying the number of tasks  $T$ . As  $T$  increases, Phase 1 accuracy decreases (approximately proportional to  $1/T$ ). When  $T = 1$ , the setup aligns with previous studies (see Figure 1), where the model’s accuracy increases in a single phase rather than undergoing multiple phases. Conversely, for  $T \geq 2$ , the model consistently exhibits three distinct phases. This indicates that the multi-phase phenomenon is robust to the number of tasks, and that introducing additional tasks in the ICL

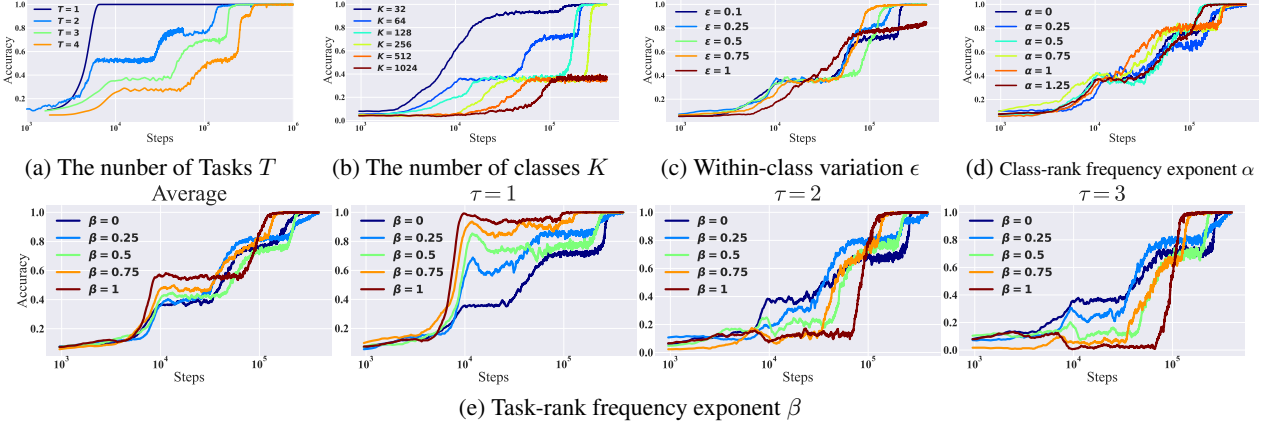


Figure 7. The relationship between learning phase dynamics and data distribution properties is explored by varying key parameters: the number of tasks ( $T$ ), the number of classes ( $K$ ), the noise magnitude ( $\epsilon$ ), the sampling bias for classes ( $k^{-\alpha}$ ), and the sampling bias for tasks ( $\tau^{-\beta}$ ). Default values are  $T = 3$ ,  $K = 64$ ,  $\epsilon = 0.1$ ,  $\alpha = 0$ , and  $\beta = 0$ . The plots show how these variations influence accuracy and the emergence of learning phases.

setting can provide new empirical insights.

In Figure 7-(b), when  $K$  is small (e.g.,  $K = 32$ ), the model tends to skip Phase 1 and transition directly to Phase 2. In contrast, when  $K$  is large (e.g.,  $K = 128, 256$ ), the model skips Phase 2 and jumps directly from Phase 1 to Phase 3. This can be explained by the theoretical values derived in Section 4.3, where increasing the number of classes brings the accuracy in Phase 2 closer to that in Phase 1, effectively making Phase 2 unobservable for large  $K$ .

In Figure 7-(c), increasing  $\epsilon$  (the within-class variation) leads to skipping Phase 2. Moreover, when  $\epsilon$  is 1, Phase 1 is also skipped. Following the results of Chan et al. (2022), higher values of  $\epsilon$  make it more difficult for the model to memorize the  $(x, \ell)$  pairs in its weights, and thus it shifts its focus toward leveraging the context. The observation that NCC is skipped entirely when  $\epsilon = 1$  aligns with this trend. Although SCC is a circuit that uses the context, it inherits the nature of NCC, causing it to be skipped as  $\epsilon$  increases. In Figure 7-(d), we see that increasing  $\alpha$  likewise tends to skip Phase 1 or Phase 2. The heightened sampling bias makes it more challenging to memorize pairs in the weights, so the model more readily exploits context-based information. As a result, the NCC or SCC does not emerge. In summary, the results suggest that when the model finds it difficult to memorize  $(x, \ell)$  pairs (larger  $\epsilon$  or  $\alpha$ ) neither NCC nor SCC emerges.

In Figure 7-(e), we examine how varying the task sampling bias  $\beta$  affects both the average accuracy across tasks and the accuracy of each individual task. While changing  $\beta$  leads to only minor differences in the overall average trend, the accuracy on a per-task basis varies considerably with  $\beta$ . In particular, when  $\beta$  is high (e.g.,  $\beta = 1$ ), the model tends to memorize the most frequent task (i.e.,  $\tau = 0$ ) first, causing the remaining tasks to skip NCC and progress directly to

forming FCC. Additional results for larger values of  $T$  and varying context length ( $N$ ) are provided in Appendix I and Appendix J, respectively.

## 5. Multi-Head Enhances Circuit Discovery

### 5.1. Parallel Circuit Exploration

To investigate a more practical scenario, we extend our analysis to multi-head attention. Figure 8 compares the accuracy changes for models with two heads and one head. In the left panel of Figure 8, we observe that learning phases become less pronounced when using multi-head attention. A closer examination of the attention maps for each head (as shown in the right panel of Figure 8) reveals that different heads specialize in distinct functions. Specifically, one head learns circuits resembling NCC, while another head becomes FCC. This parallel specialization provides a smoother trajectory of accuracy improvement, in contrast to the multi-learning phase observed in single-head models.

These findings suggest that multi-head attention allows for parallel exploration of circuits, improving the efficiency of circuit discovery. As a result, the multiple phase characteristic of single-head models are absent in multi-head configurations. This behavior aligns with observations in LLMs, where multi-head attention enables different heads to serve distinct functions, leading to smoother accuracy improvements, as seen in Figure 8. Results for a larger number of heads are provided in the Appendix G.

### 5.2. Hidden Circuit Emergence

In Figure 8, we observe multiple attention heads lead to smoothing the accuracy improvement. To gain deeper insights into this phenomenon, we analyze how the internal circuits evolve by using the circuit metrics summarized

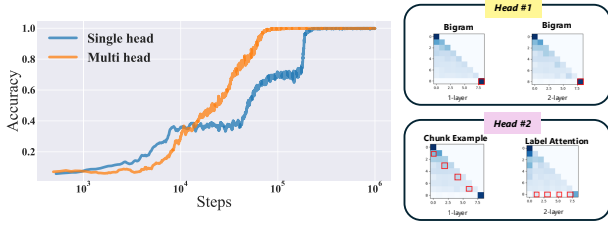


Figure 8. Comparison of accuracy dynamics between single-head (blue) and multi-head (orange) attention models (left). The multi-head model exhibits smoother accuracy improvements, without the distinct learning phases observed in the single-head model. On the right, the attention maps for the two heads in the multi-head model are visualized. Head 1 specializes in NCC, while Head 2 adopts circuits resembling FCC. These findings indicate that multi-head attention allows parallel circuit discovery, enhancing the efficiency of the learning process.

in Table 2. In Figure 9 (left), we present the circuit metrics for Bigram and Label Attention in Head 2. Notably, around the 30,000th training step, the Bigram metric exhibits a pronounced increase in the second layer, whereas the Label Attention metric is notably larger in the first layer. The right panel displays the corresponding attention maps, which clearly demonstrate an SCC-like pattern, illustrating how the model’s attention shifts between bigram-driven and label-focused mechanisms. The attention maps on the Figure 9 (right) correspond to the model’s behavior at 30,000 training steps, as indicated by the vertical dashed line. A complete set of metrics is provided in Appendix H.

These results suggest that, even though we do not observe abrupt learning in accuracy under the multi-head configuration, a *hidden circuit* emerge within the model’s internal mechanisms. This hidden phenomenon implies that, in more practical scenarios (such as large-scale language model where the loss typically decreases in a smooth fashion), the model’s internal circuits may still undergo significant emergent shifts.

## 6. Discussion

We introduced controlled experimental called In-Context Meta-Learning (ICML), designed to move beyond simple copy tasks by requiring task inference. We then investigate how a 2-layer, attention-only transformer acquires ICL abilities, inspired by induction head research (Olsson et al., 2022; Reddy, 2023). Although our model is much smaller than those used in large-scale interpretability research (Wang et al., 2022; Merullo et al., 2024; Templeton et al., 2024; Gao et al., 2024), this controlled design revealed novel insights, including multi-learning phases that illuminate how the model’s internal circuits evolve. Moreover, the observed random-label robustness (Section 4.3) and multi-head behaviors, where the loss decreases smoothly (Section 5), both

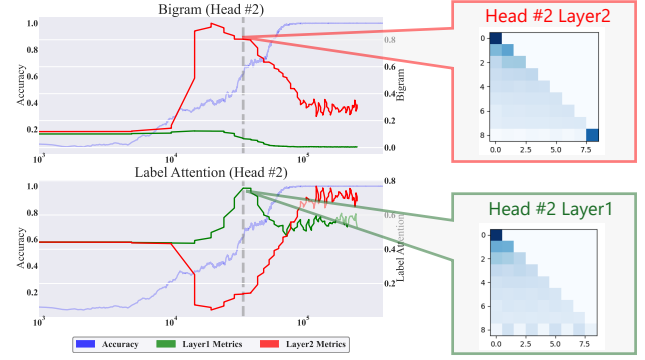


Figure 9. Circuit metrics (left) and attention maps (right) for Bigram (Head 2) and Label Attention (Head 2) in multi head setting. The left plots depict the progression of Accuracy (blue), Layer 1 Metrics (green), and Layer 2 Metrics (red) over training steps. The attention maps on the right correspond to the model’s behavior at 30,000 training steps, as indicated by the vertical dashed line.

align with findings in LLMs. These results connect small-scale experiments to practical LLMs, clarifying ICL mechanisms. Additional related work is presented in Appendix K.

**Relationship to Prior Internal-Circuit Research** Previous investigations taking an internal-circuit approach to ICL have largely focused on induction heads, which employ a match-and-copy mechanism (Ren et al., 2024; Cho et al., 2024). In contrast, by adopting a more practical meta-learning perspective, our study reveals multi-phase circuits that initially memorize examples and then evolve to infer the underlying task, which differs from the single-learning phase commonly observed in induction heads. While both induction heads and our Full-Context Circuits (FCC) chunk contextual  $(x, \ell)$  pairs into a single token in the first layer, the second layer diverges: induction heads retrieve only a label, whereas FCC further aggregates (Chunk Example  $\rightarrow$  Label Attention in Table 1). This shared mechanism in the first layer implies that even a simple copy task contributes to meta-learning-like ICL capabilities. In addition, consistent with earlier findings (Chan et al., 2022; Singh et al., 2023; Reddy, 2023), these results highlight the key role of dataset characteristics in circuit formation and ICL performance.

**Implication for LLMs** Our analysis links circuits to the established concept of task vectors (Hendel et al., 2023; Todd et al., 2024). A task vector represents the abstracted representation a model forms from examples, and although such vectors have been recognized, the internal circuit-based mechanisms that produce them remain poorly understood. Our findings offers a step toward elucidating these mechanisms. In addition, we examine multi-head attentions. Prior work (Singh et al., 2024) has identified redundancy in induction heads under multi-head architectures. Our findings indicate that, rather than mere redundancy, multiple distinct circuits emerge in parallel in the multi-head setting, resulting



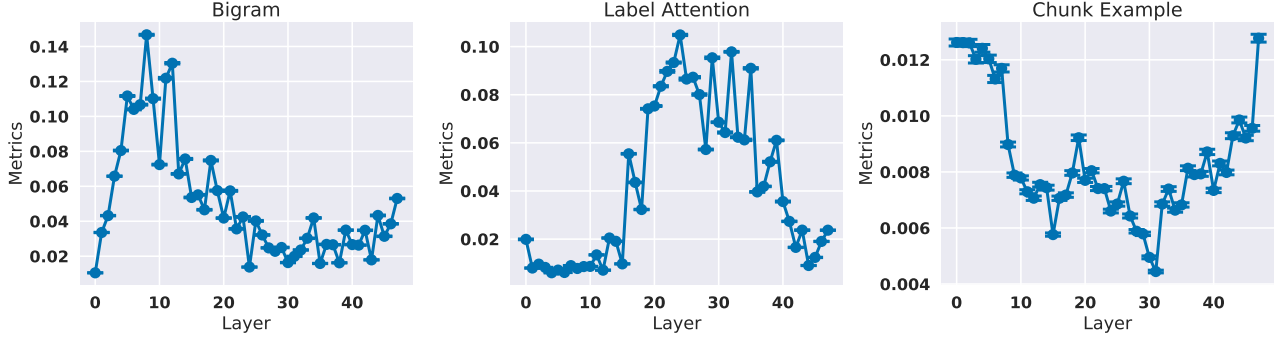


Figure 10. Layer-wise analysis of Bigram, Label Attention, and Chunk Example metrics in a pretrained LLM (GPT2-XL). We observe that chunk example scores peak in earlier layers while label attention scores are higher in middle or later layers, consistent with the final circuit (FCC) behavior in our 2-layer attention-only model, where the first layer emphasizes chunk example and the second layer specializes in label attention.

in smoother performance gains. This observation bridges the discontinuous concept of circuits with the continuous performance improvements seen in LLMs.

To test whether the circuits we observe in our controlled toy setting also appear in real-world pretrained models, we conduct an additional analysis using a standard sentiment classification task. Specifically, we use the SST2<sup>2</sup> dataset from the GLUE benchmark, consisting of 872 sentiment-labeled samples. We use GPT2-XL<sup>3</sup> (48-layer decoder transformer, pretrained). Each prompt contains two labeled examples followed by a query example without its label, in a 2-shot setup. The prompt format is as follows:

```
Review:{text}\nSentiment:{label}
Review:{text}\nSentiment:{label}
Review:{text}\nSentiment:
```

An actual example of such a prompt is provided in Appendix N. As the model is fully trained, we cannot observe circuit formation over training; instead, we analyze attention patterns across layers in response to the fixed prompts. We define three attention-based metrics, using raw attention probabilities  $p(i, j)$  from token  $j$  to token  $i$ , averaged over all heads in each layer:

1. **Bigram Attention:**  $p(\text{query}, \text{query})$   
Measures the self-attention of the final token (query).
2. **Label Attention:**  $\frac{1}{K} \sum_{k=1}^K p(\text{query}, \text{label}_k)$   
Measures how much the query attends to each of the  $K = 2$  label tokens.
3. **Chunk Example Attention:**  $\frac{1}{K} \sum_{k=1}^K \frac{\sum_i p(\text{label}_k, \text{text}_{k,i})}{\sum_i \text{text}_{k,i}}$   
Measures how strongly each label token attends to the corresponding review tokens.

<sup>2</sup><https://huggingface.co/datasets/stanfordnlp/sst2>

<sup>3</sup><https://huggingface.co/openai-community/gpt2-xl>

Figure 10 shows that the Chunk Example metric is higher in early layers, while Label Attention dominates later layers—mirroring our two-layer model’s progression from chunk example to label focus. This pattern aligns with our earlier findings in small Transformers, suggesting these circuits generalize to LLMs.

We further investigate circuit behaviors in more standard Transformer architectures (see Appendix L), under next-token prediction objectives (see Appendix M), and in models deeper than two layers (see Appendix O). In all these cases, we observe consistent structural patterns, supporting the robustness and generality of our circuit-based interpretation.

## 7. Conclusion

We introduced In-Context Meta-Learning (ICML), a controlled setting for analyzing how attention-only transformers acquire in-context learning abilities. Unlike simpler induction-head settings limited to match-and-copy circuits, our approach allowed us to explore how internal circuits function in a more practical task-inference context. Our analysis revealed a multi-phase learning process, where early layers bind example pairs (chunk example) and later layers abstract task-relevant patterns (label attention). These circuits proved robust to random labels and benefited from multi-head attention, resulting in smoother learning dynamics. We further showed similar circuit patterns emerging in pretrained models, such as GPT2-XL, on real-world natural language tasks, suggesting our findings generalize beyond toy settings. While our work is still far from fully capturing the complexity of real LLM behaviors, connecting controlled experiments in mechanistic interpretability to realistic use-cases of LLMs is becoming increasingly important. Such efforts will help advance interpretability research and play a crucial role in the development of safer AI systems.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## Acknowledgements

HF was supported by JSPS KAKENHI Grant Number JP22J21582.

## References

- Bhattachamishra, S., Patel, A., Blunsom, P., and Kanade, V. Understanding in-context learning in transformers and llms by learning to learn discrete functions. *arXiv preprint arXiv:2310.03016*, 2023.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Chan, S. C. Y., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A., Richemond, P. H., McClelland, J., and Hill, F. Data distributional properties drive emergent in-context learning in transformers. *arXiv preprint arXiv:2205.05055*, 2022.
- Cho, H., Kato, M., Sakai, Y., and Inoue, N. Revisiting in-context learning inference circuit in large language models. *arXiv preprint arXiv:2410.04468*, 2024.
- Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. *arXiv preprint arXiv:2304.14997*, 2023a.
- Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability, 2023b. URL <https://arxiv.org/abs/2304.14997>.
- Crosbie, J. and Shutova, E. Induction heads as an essential mechanism for pattern matching in in-context learning. *arXiv preprint arXiv:2407.07011*, 2024.
- Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.
- D’Angelo, F., Croce, F., and Flammarion, N. Selective induction heads: How transformers select causal structures in context. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=bnJgzAQjWf>.
- Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., and Sui, Z. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2024.
- Edelman, E., Tsilivis, N., Edelman, B. L., Malach, E., and Goel, S. The evolution of statistical induction heads: In-context learning markov chains. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 64273–64311. Curran Associates, Inc., 2024.
- Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. <https://transformer-circuits.pub/2021/framework/index.html>.
- Furuta, H., Minegishi, G., Iwasawa, Y., and Matsuo, Y. Towards empirical interpretation of internal circuits and properties in grokked transformers on modular polynomials. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL <https://openreview.net/forum?id=MzSf70uXJO>.
- Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders, 2024. URL <https://arxiv.org/abs/2406.04093>.
- Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022.
- He, T., Doshi, D., Das, A., and Gromov, A. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=aVh9KRZdRk>.
- Hendel, R., Geva, M., and Globerson, A. In-context learning creates task vectors. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL <https://openreview.net/forum?id=QYvFULF19n>.

- Hoogland, J., Wang, G., Farrugia-Roberts, M., Carroll, L., Wei, S., and Murfet, D. Loss landscape degeneracy drives stagewise development in transformers, 2025. URL <https://arxiv.org/abs/2402.02364>.
- Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.
- Kim, J. and Suzuki, T. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. *arXiv preprint arXiv:2402.01258*, 2024.
- Merullo, J., Eickhoff, C., and Pavlick, E. Circuit component reuse across tasks in transformer language models. *arXiv preprint arXiv:2310.08744*, 2024.
- Min, S., Lewis, M., Zettlemoyer, L., and Hajishirzi, H. MetaICL: Learning to learn in context. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201. URL <https://aclanthology.org/2022.naacl-main.201/>.
- Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022b.
- Minegishi, G., Iwasawa, Y., and Matsuo, Y. Bridging lottery ticket and grokking: Understanding grokking from inner structure of networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL <https://openreview.net/forum?id=eQeYyupltm>.
- Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.
- Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020. URL <https://distill.pub/2020/circuits/zoom-in>.
- Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.
- Park, C. F., Lubana, E. S., and Tanaka, H. Competition dynamics shape algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=XgH1wfHSX8>.
- Raventos, A., Paul, M., Chen, F., and Ganguli, S. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=BtAz4a5xDg>.
- Reddy, G. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *arXiv preprint arXiv:2312.03002*, 2023.
- Ren, J., Guo, Q., Yan, H., Liu, D., Zhang, Q., Qiu, X., and Lin, D. Identifying semantic induction heads to understand in-context learning. *arXiv preprint arXiv:2402.13055*, 2024.
- Singh, A. K., Chan, S. C. Y., Moskovitz, T., Grant, E., Saxe, A. M., and Hill, F. The transient nature of emergent in-context learning in transformers. *arXiv preprint arXiv:2311.08360*, 2023.
- Singh, A. K., Moskovitz, T., Hill, F., Chan, S. C., and Saxe, A. M. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=O8rrXl71D5>.
- Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Summers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL <https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html>.
- Todd, E., Li, M., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=AwxytyMwaG>.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.,

2017. URL [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J. a., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.

von Oswald, J., Niklasson, E., Schlegel, M., Kobayashi, S., Zucchet, N., Scherrer, N., Miller, N., Sandler, M., y Arcas, B. A., Vladymyrov, M., Pascanu, R., and Sacramento, J. Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*, 2023.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Zipf, G. K. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA, 1949.



## A. Induction Head

Figure 11 illustrates an induction circuit consisting of a *previous token head* in Layer 1 and an *induction head* in Layer 2. After Layer 1, the side-by-side  $x$  and  $\ell$  tokens are chunked into a single token. In Layer 2, two operations occur: **matching** of  $x$  via queries and keys (in purple) and **copying** of  $\ell$  (in red).

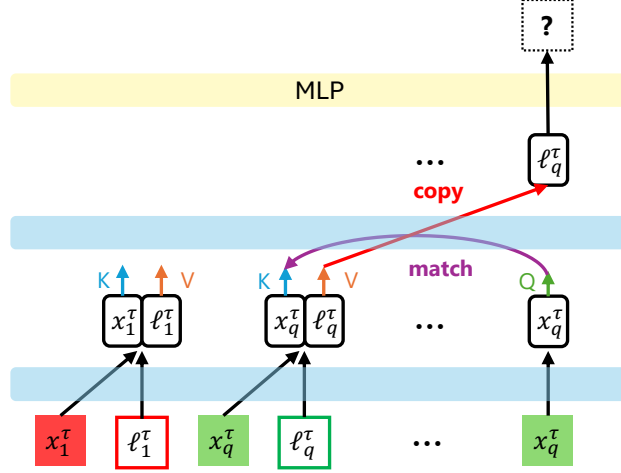


Figure 11. The circuit consists of a previous token head in Layer 1 and an induction head in Layer 2. After Layer1, the side-by-side  $x$  and  $\ell$  tokens are chunked into a single token. In Layer 2, we highlight two operations: **matching** of  $x$  via queries and keys (in purple), and **copying** of  $\ell$  (in red).

When a sample is drawn with probability  $p_B$ , the burstiness parameter  $B$  introduced by Chan et al. (2022); Reddy (2023) becomes relevant, determining how many times items from the query class appear in an input sequence (where  $N$  is a multiple of  $B$ ). In our ICML setup, we specifically examine the case with  $T = 1$ ,  $p_B = 1$ , and  $B = 1$ , as shown in Figure 12. We observe that the first attention layer encodes each  $(x, \ell)$  pair into a single token, while the second layer strongly attends to one of these pairs, effectively implementing the match-and-copy mechanism characteristic of an induction head. Notably, our setting thus subsumes the standard induction head experiments proposed in Reddy (2023).

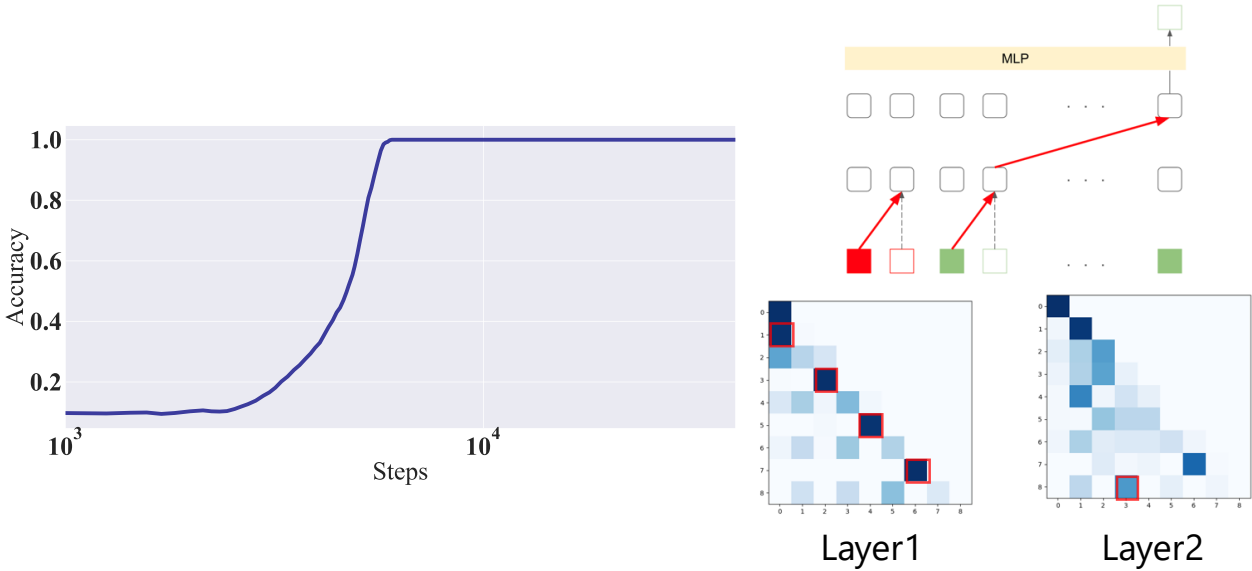


Figure 12. (left) The emergence of induction heads is observed as single-learning phase. (right) The attention maps on the right illustrate the circuit mechanism, where Layer 1 groups  $(x, \ell)$  pairs into single-token representations, and Layer 2 then copies this label.

## B. Model Details

### B.1. Network Architecture

Our model features two layers of **multi-head attention** with a causal mask, followed by a two-layer MLP classifier. Each attention layer  $\mu \in \{1, 2\}$  has  $m$  heads, labeled by  $h$ . Let  $(u_1, \dots, u_n)$  be the input sequence (subject to a causal mask ensuring  $i$  can only attend to  $j \leq i$ ). The outputs of the first layer are  $\{v_i\}$ ; those of the second layer are  $\{w_i\}$ .

**Attention Computation.** Within layer  $\mu$ , head  $h$  computes attention weights

$$p_{ij}^{(\mu, h)} = \frac{\exp((K_\mu^{(h)} u_j)^\top (Q_\mu^{(h)} u_i))}{\sum_{k \leq i} \exp((K_\mu^{(h)} u_k)^\top (Q_\mu^{(h)} u_i))}, \quad (1)$$

where  $Q_\mu^{(h)}$  and  $K_\mu^{(h)}$  are the learnable query and key matrices for head  $h$  in layer  $\mu$ . Next, each head outputs a weighted sum of the *value*-transformed inputs:

$$\text{Head}_i^{(\mu, h)} = \sum_{j \leq i} p_{ij}^{(\mu, h)} (V_\mu^{(h)} u_j), \quad (2)$$

where  $V_\mu^{(h)}$  is the corresponding value matrix.

**Multi-Head Aggregation.** The outputs of all  $m$  heads in layer  $\mu$  are concatenated and projected by a trainable matrix  $W_{O\mu}$ , yielding

$$v_i = u_i + W_O^1 [\text{Head}_i^{(1,1)}; \dots; \text{Head}_i^{(1,m)}], \quad (3)$$

$$w_i = v_i + W_O^2 [\text{Head}_i^{(2,1)}; \dots; \text{Head}_i^{(2,m)}]. \quad (4)$$

Here,  $[\dots]$  indicates concatenation over the head outputs, and each  $W_{O\mu}$  is a learnable linear projection.

**Classifier.** The two-layer MLP receives the final attention outputs  $\{w_i\}$  (e.g., specifically  $w_n$ , if  $n$  indexes the query token). A hidden layer with ReLU activation is followed by a softmax that produces label probabilities.

### B.2. Training Details

Table 3. Training and Model Configuration

Hyperparameter	Value
Loss Function	Cross-entropy
Optimizer	Vanilla SGD
Learning Rate	0.01
Batch Size	128
Dimension of query/key/value	128
MLP Hidden Layer Dimension	128
Causal Mask	Restrict sums to $j \leq i$

## C. Controlled Circuit Pruning Experiments

To validate the relationship between the identified circuits and model performance, we conducted controlled pruning experiments. In these experiments, all components except the circuits corresponding to a specific phase were pruned at initialization, isolating the contribution of each circuit. For comparison, we also trained a fully trainable model, referred to as the *full model*, which could attend to all identified attention patterns.

As shown in Figure 13, networks trained with only the circuits from a particular phase plateaued at accuracies corresponding to that phase. This result provides strong evidence that the circuits identified in each phase are directly responsible for the observed performance.

Interestingly, when the Phase 3 circuit was provided from the beginning (pink curve in Figure 13), the model achieved 100% accuracy in single step. In contrast, the full model exhibited a more gradual improvement, sequentially discovering and leveraging the circuits corresponding to each phase. This highlights the dynamic nature of the full model’s training process, where it incrementally constructs and refines the required circuits during training.

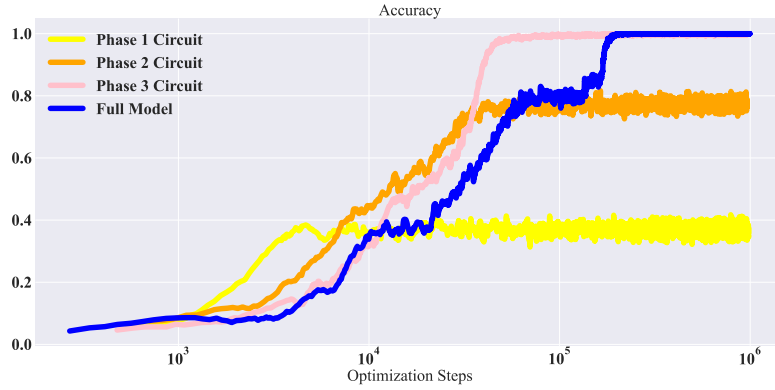


Figure 13. Controlled pruning experiments to validate the relationship between identified circuits and model performance. Networks trained with only the circuits from a specific phase plateaued at accuracies corresponding to that phase (yellow: Phase 1, orange: Phase 2, pink: Phase 3). This demonstrates that the identified circuits are directly responsible for the observed performance in each phase.

## D. Derivation of the Theoretical Accuracy

In the main text, we define

$$p = 1 - \frac{\binom{K-2}{4}}{\binom{K-1}{4}}, \quad (5)$$

and use it to obtain the “Theoretical Accuracy” as

$$\text{Theoretical Accuracy} = p \cdot 1 + (1 - p) \cdot 0.5. \quad (6)$$

This appendix provides a more detailed derivation of these formulas, along with the underlying conditions.

### Task Conditions.

1. The number of classes ( $K$ ) equals the number of labels ( $L$ ), with no duplication.
2. The input context (including the query) contains no duplicate classes.
3. Only two tasks are considered ( $T = 2$ ).
4. There are no common  $(x, \ell)$  pairs shared between different tasks.
5. To focus on SCC, a mask is applied to circuits associated with SCC during training.

**Excluding Both  $L_1$  and  $L_2$ .** We are interested in the probability that the context does *not* contain  $L_1$  or  $L_2$ .

- The total number of ways to choose 4 distinct classes from the  $K - 1$  classes (excluding the query's class) is  $\binom{K-1}{4}$ .
- To exclude both  $L_1$  and  $L_2$ , we must choose all 4 classes from the remaining  $K - 2$  classes, leading to  $\binom{K-2}{4}$  possible ways to form the context with neither  $L_1$  nor  $L_2$  present.

Hence, the probability that *neither*  $L_1$  nor  $L_2$  is in the context is

$$\frac{\binom{K-2}{4}}{\binom{K-1}{4}}.$$

**Probability  $p$  and the Accuracy Calculation.** We denote by  $p$  the probability that *at least one* of  $L_1$  or  $L_2$  appears in the context:

$$p = 1 - \frac{\binom{K-2}{4}}{\binom{K-1}{4}}.$$

Under the task rules, if at least one of these two labels appears in the context, it *cannot* be the label for the query, so the other one must be correct. This yields 100% accuracy in that scenario. Conversely, if *neither*  $L_1$  nor  $L_2$  is found in the context (probability  $1 - p$ ), the model is forced to guess between two equally likely options, resulting in 50% accuracy. Therefore,

$$\text{Theoretical Accuracy} = p \cdot 1 + (1 - p) \cdot 0.5,$$

as stated in the main text.



## E. Rank-Frequency

In natural language processing and other real-world domains, both data instances and task distributions often follow a power-law structure, commonly referred to as *Zipf's law* (Zipf, 1949). This law states that the frequency of an item or task is inversely proportional to its rank, meaning that a small number of elements occur frequently, while the majority appear rarely. Formally, this is expressed as:

$$f(k) \propto k^{-\alpha}, \quad (7)$$

where  $k$  denotes the rank of an item, and  $\alpha$  controls the degree of skewness. Figure 14 illustrates how increasing  $\alpha$  leads to a more imbalanced distribution, with a steep drop in frequency beyond the highest-ranked elements.

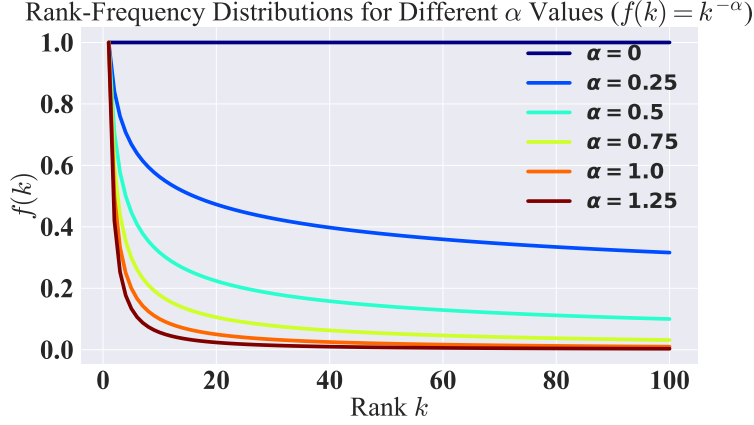


Figure 14. Rank-frequency distributions for different values of the power-law exponent  $\alpha$ , following the Zipfian distribution  $f(k) = k^{-\alpha}$ . As  $\alpha$  increases, the distribution becomes more skewed, with a few high-frequency items dominating while the majority appear infrequently.

In our setting, not only data but also task sampling follows a similar Zipfian distribution:

$$f(\tau) \sim \tau^{-\beta}, \quad (8)$$

where  $\beta$  determines the skewness of the task distribution.

## F. Effects of Birstiness on Circuit Emergence

When a sample is drawn with probability  $p_B$ , the burstiness parameter  $B$  introduced by Chan et al. (2022); Reddy (2023) becomes relevant, determining how many times items from the query class appear in an input sequence (where  $N$  is a multiple of  $B$ ). Figure 15 examines the impact of burstiness  $B$  and probability  $p_B$ . The left panel shows accuracy curves for different values of  $B$  at a fixed  $p_B = 0.25$ . As  $B$  increases, Phase 1, where NCC memorizes pairs through weight updates — tends to be skipped. The right panel presents accuracy curves for different values of  $p_B$  while keeping  $B = 1$  fixed. As  $p_B$  increases, the model’s accuracy improves more smoothly, and distinct learning phases become less pronounced. These results align with previous studies showing that increased burstiness tends to shift the model away from weight-based solutions and toward context-dependent reasoning (Chan et al., 2022; Reddy, 2023).

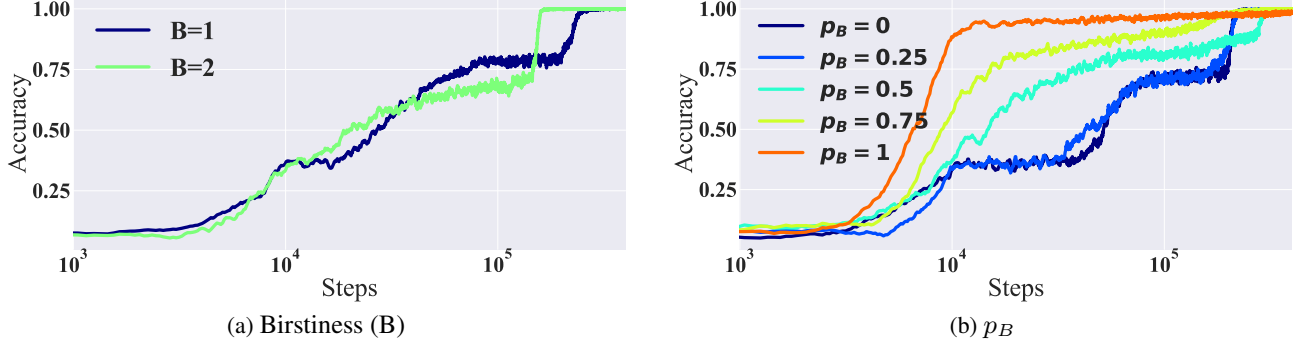


Figure 15. **(Left)** Accuracy curves for different values of  $B$  at a fixed  $p_B = 0.25$ . Increasing  $B$  tends to skip Phase 1, where NCC memorizes pairs through weights. **(Right)** Accuracy curves for different values of  $p_B$  with  $B = 1$ . As  $p_B$  increases, the learning process becomes smoother, reducing the occurrence of distinct learning phases.

## G. Multi-Heads Experiments

Figure 16 (Left) shows accuracy curves over training steps for different numbers of attention heads (1, 2, 4, 8, and 16). Models with multiple heads exhibit a smooth increase in accuracy, whereas the single-head configuration undergoes multi-learning phases, where accuracy improves in distinct jumps rather than gradually.

Figure 16 (Right) visualizes attention patterns in a 4-head attention model across two layers. The four heads naturally divide into two functional roles: two heads focus on NCC, while the other two heads focus on FCC.

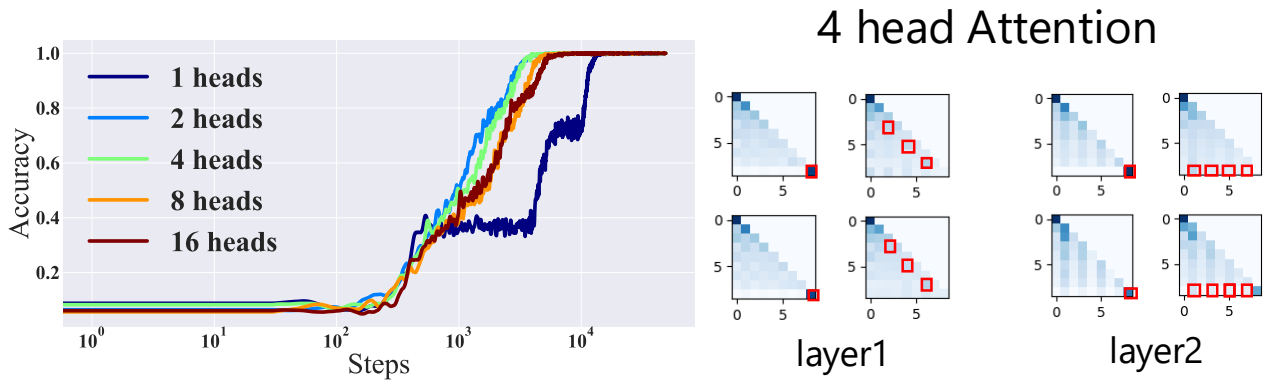


Figure 16. **(Left)** Accuracy curves over training steps for different numbers of attention heads (1, 2, 4, 8, and 16). Models with multiple heads exhibit a smooth increase in accuracy, whereas the single-head configuration shows multi-learning phases. **(Right)** Visualization of attention patterns in a 4-head attention model, separated by layer. Two heads focus on NCC, while the other two focus on FCC. Red squares highlight key attention positions indicative of each role.

## H. Circuit Metrics in Multi-head Attention

Figure 17 presents circuit metrics for each attention head, analyzed by layer in a two-head attention model. Head 1 consistently maintains high bigram values across both Layer 1 and Layer 2. This indicates that it primarily performs token-level copying operations, forming an NCC. In contrast, Head 2 exhibits a different pattern. As training progresses, the chunk example metric increases in Layer 1, while the label attention metric becomes dominant in Layer 2, forming an FCC.

These findings reinforce the idea that multi-head attention facilitates specialization, allowing different heads to develop distinct computational circuits that enhance the model’s meta-learning capabilities.

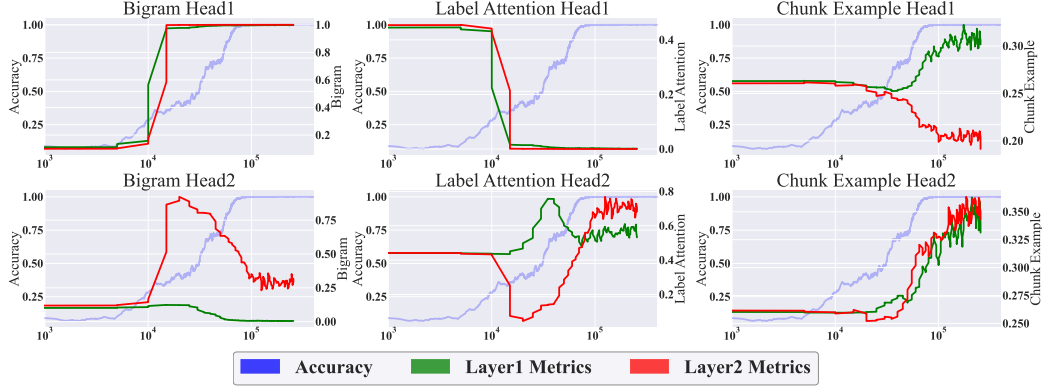


Figure 17. Circuit metrics for each attention head, analyzed by layer in two heads attentions. Head 1 maintains high bigram values across both Layer 1 and Layer 2, indicating the formation of an NCC. In contrast, Head 2 exhibits increasing chunk example values in Layer 1 and high label attention values in Layer 2, suggesting the formation of an FCC.

## I. Impact of Increasing Task Numbers ( $T$ )

Figure 18 shows accuracy (left) and loss (right) for models trained with increasing numbers of tasks ( $T = 3, 6, 9, 12, 15, 18$ ). Even with higher  $T$ , models exhibit sudden accuracy jumps. As  $T$  increases, initial accuracy decreases, and it takes longer for models to achieve sharp accuracy improvements, highlighting the challenges of training under more realistic conditions.

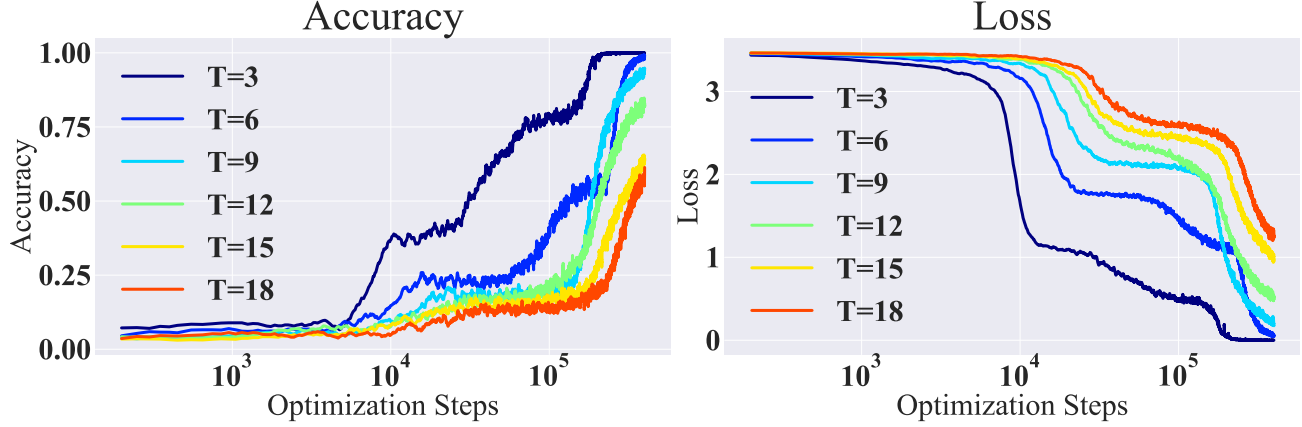


Figure 18. Left: Accuracy, Right: Loss, for increasing numbers of tasks ( $T$ ) set to 3, 6, 9, 12, 15, and 18, bringing the setup closer to real-world conditions. Even with higher  $T$ , the model still exhibits sudden jump in accuracy. As  $T$  increases, the accuracy in the first phase (around  $1/T$ ) decreases, and it takes longer to reach the next sharp jump in accuracy.

## J. Impact of Increasing Context Length ( $N$ )

Figure 19-(a) presents accuracy curves when increasing the number of few-shot examples ( $N$ ), resulting in a longer total context. Multiple learning phases are visible for contexts up to  $N = 8$ . For  $N \geq 16$ , the model quickly achieves perfect accuracy, indicating easier learning with more context. Figure 19-(b) shows attention maps at  $N = 16$  (context length=33) with clear chunk-example attention patterns in layer 1 and label-attention patterns in layer 2, consistent with behaviors observed at lower contexts.

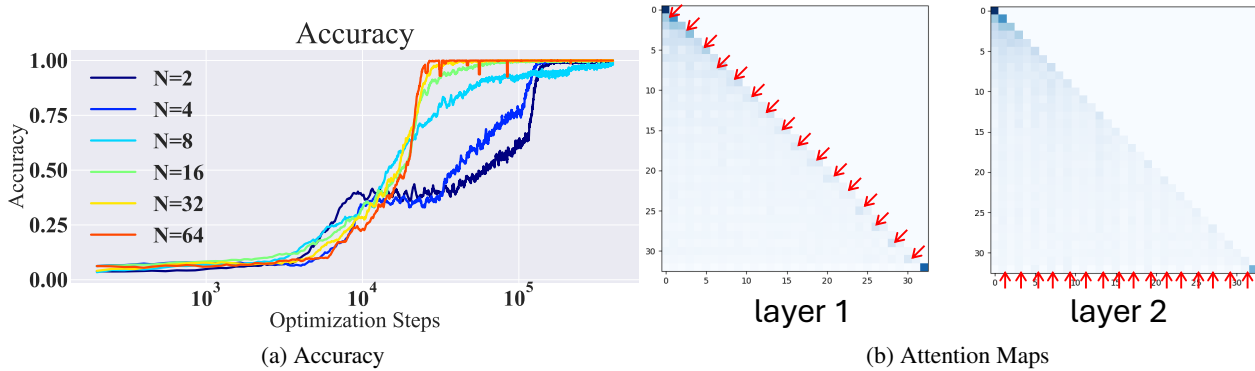


Figure 19. (a) Accuracy curves when increasing the number of few-shot examples ( $N$ ) in the context, making the total context length  $2N + 1$ . Up to about  $N = 8$ , multiple learning phases are visible. For  $N \geq 16$ , the model exhibits only a single learning phase before reaching 100 % accuracy. This behavior suggests that having a larger context makes it much easier for the model to learn a circuit that leverages the context, eliminating the need for intermediate phases. (b) Attention maps of a two-layer, attention-only Transformer with  $N = 16$  (context length = 33) at 100% accuracy. The first layer (left) shows a chunk example attention pattern, whereas the second layer (right) focuses on label attention. These observations are consistent with the circuits seen at  $N = 4$ .



## K. Extended Related Work

Our study relates closely to multiple strands of literature on in-context learning (ICL) and multi-phase emergence. Here, we elaborate on these connections and highlight distinct aspects of our approach.

**In-Context Learning Literature Beyond Copy Tasks** Previous studies on in-context learning frequently employ linear regression frameworks (Von Oswald et al., 2023; Raventos et al., 2023), offering analytically tractable models for theoretical analysis. While these approaches use identical tasks across context examples, they typically optimize for mean squared error (MSE), diverging from the conditions encountered in real-world in-context scenarios. In contrast, our experimental design closely mirrors practical applications of LLMs, emphasizing realistic task settings and diverse few-shot contexts.

Several works (D’Angelo et al., 2025; Edelman et al., 2024; Park et al., 2025) have examined in-context learning in the context of Markov chain tasks, highlighting meta-learning phenomena. Although the meta-learning dynamics studied are related to ours, these Markov chain settings differ significantly from the typical few-shot example-pair format that characterizes standard LLM usage. Our research specifically targets these conventional scenarios, aiming for greater ecological validity in interpreting LLM behavior.

Research (He et al., 2024) exploring in-context learning on modular arithmetic tasks (Nanda et al., 2023; Furuta et al., 2024; Minegishi et al., 2025) investigates phenomena like out-of-distribution generalization and the role of attention mechanisms at convergence. While extending beyond simpler tasks, such studies typically do not focus on tracking the dynamic acquisition of internal circuits throughout training. Our work uniquely captures these acquisition dynamics, directly linking them to observed behaviors like random-label accuracy emergence and multi-head smoothing.

**Multi-Phase Emergence Literature** Prior investigations into multi-phase emergence (Edelman et al., 2024) have shown transformers acquiring functional capabilities via discrete phase transitions. This finding aligns closely with our observations. However, whereas previous studies utilized Markov chain prediction tasks—progressing through uniform, unigram, and bigram phases—our experiments examine a more practically relevant progression: from Bigram to Semi-Context, and eventually Full-Context circuits. This advancement better reflects complexities observed in realistic few-shot learning scenarios.

Several developmental interpretability studies (Hoogland et al., 2025) also highlight multi-phase transitions, primarily by analyzing geometric properties of loss landscapes, such as the Local Learning Coefficient (LLC). Our analysis diverges by explicitly characterizing the mechanistic circuits underlying these transitions. An exciting avenue for future research could involve bridging these LLC-based geometric perspectives with our mechanistic circuit analyses to enrich interpretability methodologies.

## L. Experiment on Standard Transformer

Figure 20 shows accuracy and attention maps for a standard transformer (with attention and MLP layers) trained on the same task as the simpler 2-layer attention-only model from the main text. At 50k steps, the model shows clear label-attention and bigram patterns similar to the simpler model. At 400k steps, more complex circuits emerge: a chunk-example pattern is visible in layer 1, and clearer label-attention develops in layer 2 (red arrows highlight these patterns). These results confirm that insights from the simpler model apply to standard transformers.

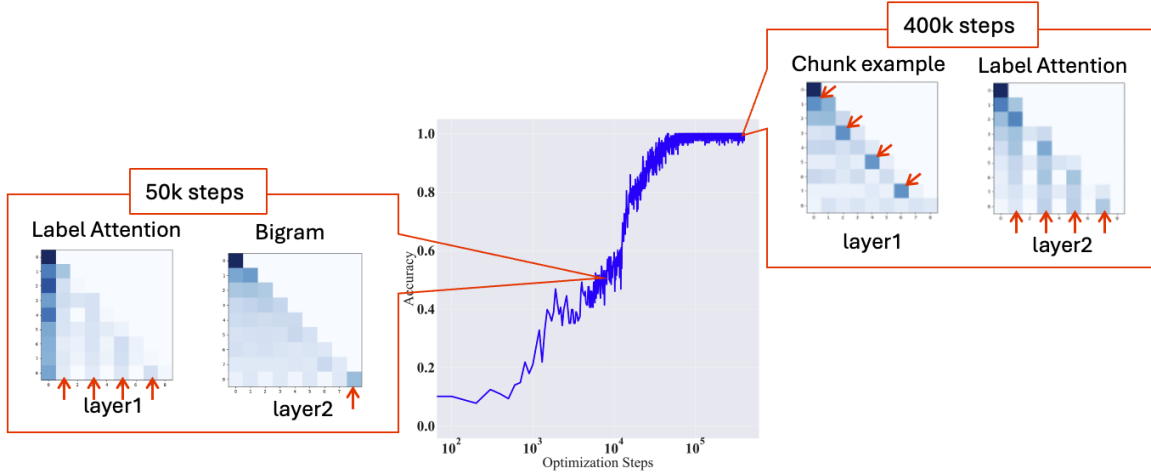


Figure 20. Accuracy and attention map visualizations (left: at 50k steps; right: at 400k steps) from training a standard Transformer rather than a 2-layer attention-only model. The label-attention & bigram circuit and chunk-example & label-attention circuit emerge here, same as in the 2-layer attention-only Transformer experiments.

## M. Experiment with Next Token Prediction

Figure 21 shows accuracy curves for two conditions: predicting only the final label token (Last Label Only) and predicting every label token (All Labels). Both conditions show clear learning phases, but the All Labels setting does not reach perfect accuracy due to the need for contextual information. The attention patterns (right) indicate that even when predicting all labels, the model develops the same final circuit (chunk example attention in Layer 1 and label attention in Layer 2) as in the simpler scenario.

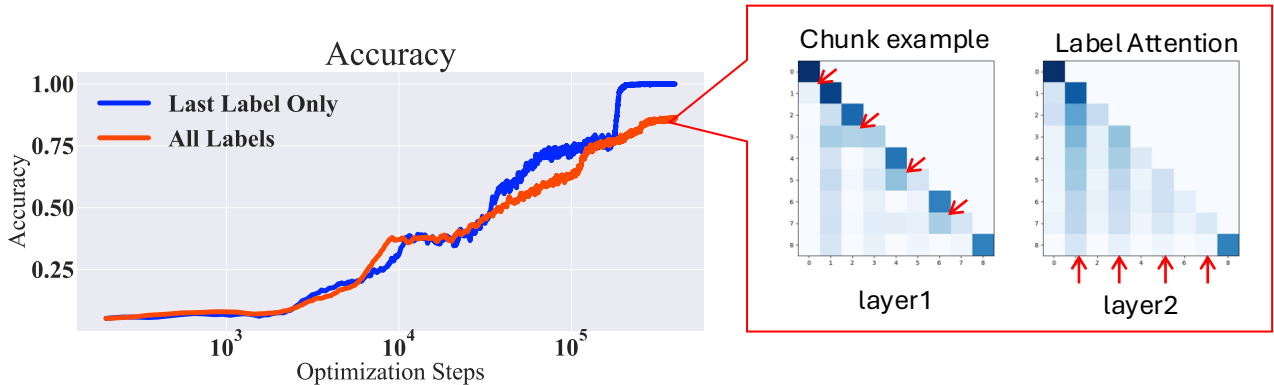


Figure 21. Accuracy curves for two conditions: (1) predicting only the final label token (“Last Label Only,” blue) and (2) predicting every label token in the context (“All Labels,” red). In both settings, we still observe learning phases. Because the task requires contextual information for correct label predictions, the “All Labels” setting never reaches 100% accuracy. The attention patterns (right) show that when predicting all label tokens, the model still learns the same final circuit (FCC), combining chunk example in Layer 1 and label attention in Layer 2.

## N. Prompt Examples of LLM Experiment

Review: hide new secretions from the parental units  
 Sentiment: Negative  
 Review: that loves its characters and communicates something rather beautiful about human nature  
 Sentiment: Positive  
 Review: remains utterly satisfied to remain the same throughout  
 Sentiment:

## O. Accuracy and Attention Patterns across Model Depths

Figure 22 shows accuracy curves of attention-only Transformers with 2 to 5 layers. Clear multiple learning phases are observed in the 2- and 3-layer models, whereas the 4- and 5-layer models exhibit smoother transitions without distinct phases. Figure 23 presents attention maps from models achieving 100% accuracy. Regardless of the total number of layers, the core circuit (FCC) consistently emerges in the first two layers: the first layer captures chunk-related information, and the second layer focuses attention on labels.

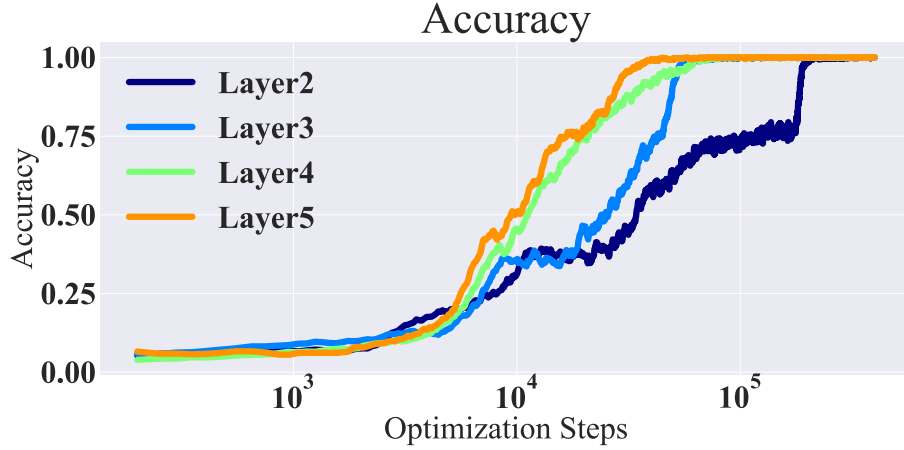


Figure 22. Accuracy curves for attention-only Transformers with 2, 3, 4, and 5 layers. The 2-layer model shows three clear learning phases, and the 3-layer model also exhibits multiple transitions. For 4- and 5-layer models, no distinct multiple learning phases are visible in the accuracy curves.

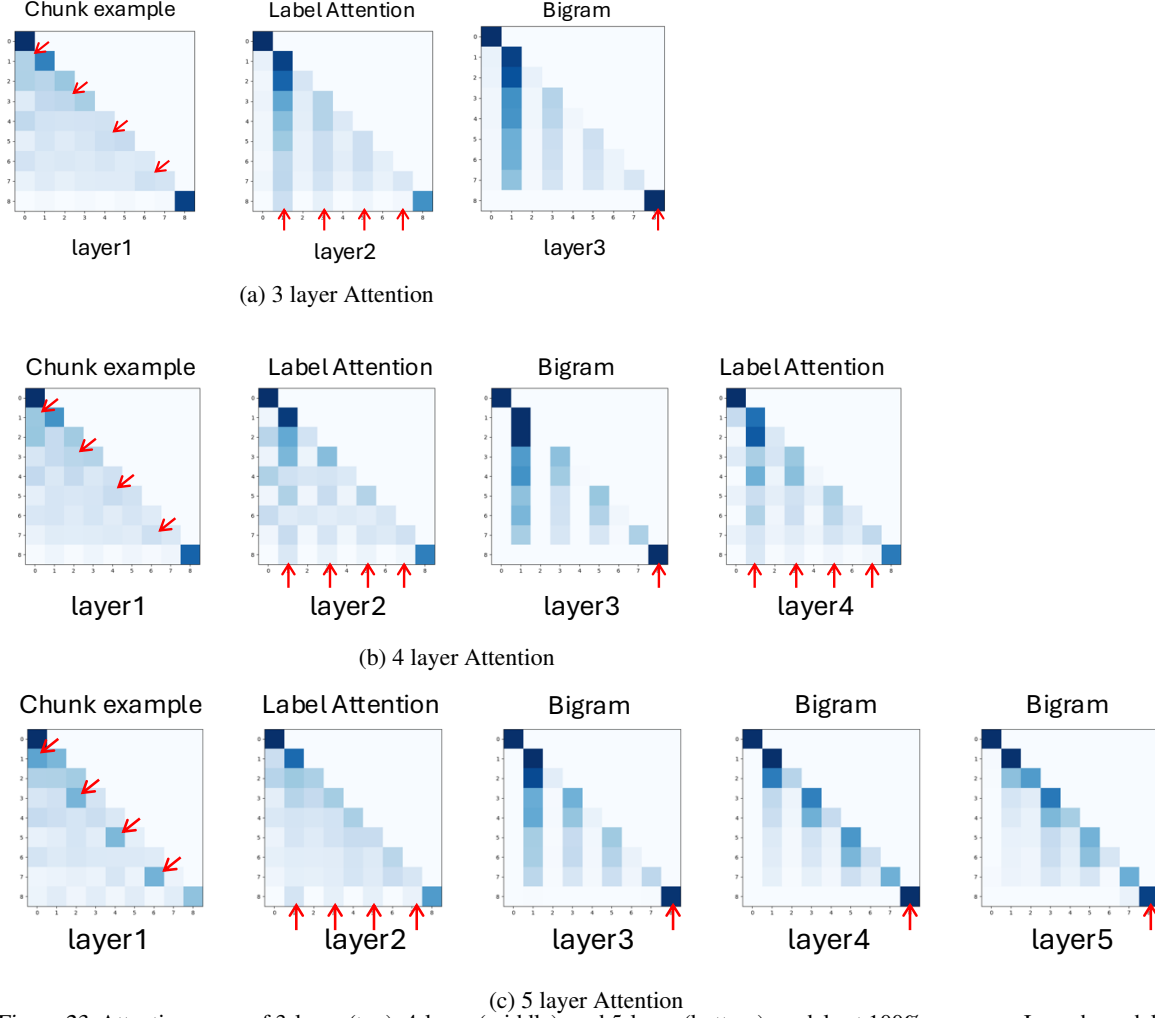


Figure 23. Attention maps of 3-layer (top), 4-layer (middle), and 5-layer (bottom) models at 100% accuracy. In each model, the first layer displays a chunk example pattern, and the second layer exhibits label attention. This suggests that the core circuit (FCC) for achieving 100% accuracy is formed in the first two layers, regardless of the total number of layers.