Identifying Semantic Induction Heads to Understand In-Context Learning

Jie Ren*[1,2], Qipeng Guo†[2], Hang Yan[2,4], Dongrui Liu[1], Quanshi Zhang[1], Xipeng Qiu[3], Dahua Lin[2,4]

[1]Shanghai Jiao Tong University [2]Shanghai Artificial Intelligence Laboratory
[3]Fudan University [4]The Chinese University of Hong Kong

Abstract

Although large language models (LLMs) have demonstrated remarkable performance, the lack of transparency in their inference logic raises concerns about their trustworthiness. To gain a better understanding of LLMs, we conduct a detailed analysis of the operations of attention heads and aim to better understand the in-context learning of LLMs. Specifically, we investigate whether attention heads encode two types of relationships between tokens in natural languages: the syntactic dependency parsed from sentences and the relation within knowledge graphs. We find that certain attention heads exhibit a pattern where, when attending to head tokens, they recall tail tokens and increase the output logits of those tail tokens. More crucially, the formulation of such semantic induction heads has a close correlation with the emergence of the in-context learning ability of language models. The study of semantic attention heads advances our understanding of the intricate operations of attention heads in transformers, and further provides new insights into the in-context learning of LLMs.

1 Introduction

In recent years, the transformer-based large language models (LLMs) (Kaplan et al., 2020; Brown et al., 2020; Touvron et al., 2023; Bubeck et al., 2023) have rapidly emerged as one of the mainstreams in the field of natural language processing (NLP). While these models demonstrate emergent abilities as they scale (Brown et al., 2020; Wei et al., 2022), they become less interpretable due to the vast number of parameters and complex architectures, which emphasizes LLMs' safety and trustworthiness (Carlini et al., 2021; Manakul et al., 2023; Ren et al., 2024). Thus, beyond classical gradient-based explanations (Simonyan et al., 2013; Li et al., 2015), and perturbation-based explanations (Ribeiro et al., 2016; Lundberg and Lee, 2017; Sundararajan et al., 2017), recent studies in mechanistic interpretability (Cammarata et al., 2020; Elhage et al., 2021) attempt to reverse engineer the computations in transformers (particularly attention layers).

The mechanistic interpretability on transformer language models was first performed by Elhage et al. (2021). They disentangle two circuits from the operation of each attention head in transformers: Query-Key circuit (determines which token the head prefers to attend to) and Output-Value circuit (determines how the head affects the output logits of the next token). Then, Elhage et al. (2021) discover that some attention heads prefer to search for a previous occurrence of the current token in context and copy the next token associated with that occurrence, as shown in Figure 1. The attention heads performing such operations are termed *induction heads*. Taking a step further, Olsson et al. (2022); Bansal et al. (2023) have discovered that the presence of induction heads has a close correlation with the in-context learning (ICL) ability of LLMs. This finding highlights the importance of understanding the behavior of attention heads to the overall learning capabilities of LLMs.

On the other hand, semantic relationships have a vital importance on natural language understanding and processing. However, Elhage et al. (2021) only focus on whether the attention heads copy the attended token, without studying semantic relationships between tokens. Another major limitation of previous studies is that Olsson et al. (2022) does not explain the popular few-shot in-context learning schema. Instead, they study the loss decreasing along with the increase of token indices. This setting does not fully capture the complete ability of LLMs to learn from the context.

In this work, beyond simple copying, we delve deeper into high-level relationships encoded in at-

*This work is done during internship at Shanghai Artificial Intelligence Laboratory.
†Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 6916–6932

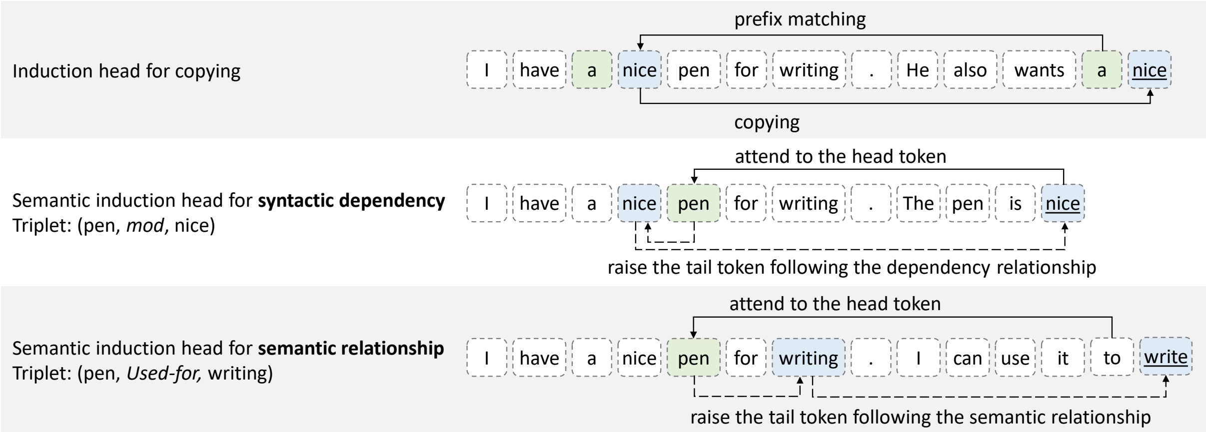

Figure 1: Induction heads and semantic induction heads. For the sequence "... a nice ... a", an induction head finds a place where the current token "a" occurred, attends to its next token "nice" (*prefix matching*), and then copies "nice" to the output *(copying*). In contrast, the semantic induction head raises the output logits of tail tokens ("nice" in the *mod* dependency and "writing" in the *Used-for* relation) when attending to the head token "pen".

tention heads. We focus on two types of relationships: (1) syntactic dependencies in the sentence and (2) semantic relationships between entities. Please refer to Figure 1 for examples. Each relation is represented as a triplet: (*head*, *relation*, *tail*). We find that when attending to head tokens, some attention heads prefer to raise the output logits of tail tokens associated with specific relations. Such attention heads encoding semantic relationships are termed *semantic induction heads*. Unlike conventional induction heads, semantic induction heads learn and leverage the semantic relationships between words to infer the output, thereby providing a better understanding of the behavior of networks.

Inspired by the study of induction heads and in-context learning, we further explore the correlation between semantic induction heads and in-context learning. We first categorize the in-context learning ability into three basic levels: loss reduction, format compliance, and pattern discovery. These three levels progressively increase in difficulty, with each subsequent level building upon the achievements of the previous one. The experimental results are consistent with our hypothesis, demonstrating the emergence of three levels of ICL in a sequential manner. Specifically, we observe the emergence of loss reduction from the beginning of the training, followed by the emergence of format compliance at around 1.6B tokens, and finally, the emergence of pattern discovery after training on approximately 4B tokens. Moreover, we find semantic induction heads mainly emerge around the same time as pattern discovery. Based on this finding, we infer that the emergence of semantic induction heads plays a crucial role in facilitating the ICL of LLMs.

Our contributions can be summarized as follows.

- We unveil the existence of semantic induction heads in LLMs that extract semantic relationships within the context. This discovery deepens the study of mechanistic interpretability and enhances our understanding of transformer-based models.
- To study the ICL in LLMs, we categorize it into three different levels and observe the gradual emergence of different levels of ICL during the early training stage of LLMs.
- Through a meticulous analysis of early checkpoints in the training of LLMs, we establish a close correlation between semantic induction heads and the occurrence of ICL.

2 Related Works

In this section, we provide an overview of recent advancements in the interpretability of neural networks, particularly mechanistic interpretability. On the other hand, previous studies (Petroni et al., 2019; Zhang et al., 2022) have also explored the topic of semantic relationships in models. However, our study distinguishes itself by focusing on mechanistic interpretability.

Previous studies in interpretability can be roughly categorized into the following four types: estimating the attribution of input features to the network output (Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg and Lee, 2017; Yang et al., 2023; Modarressi et al., 2023), discovering interaction patterns between input features (Ren et al., 2021, 2023; Liu et al., 2024; Zhou et al., 2024), extracting concepts from intermediate-layer features (Kim et al., 2018; Thomas et al., 2023; Qian et al., 2024), and designing self-explainable architectures (Li et al., 2018; Das et al., 2022). As transformer-based models become mainstream, recent works focus on understanding the attention mechanism. The most direct approach is to visualize the attention using bipartite graphs (Vig, 2019; Yeh et al., 2024) or heatmaps (Park et al., 2019). Another line of research aims to reverse engineer the operation of attention heads, called mechanistic interpretability (Cammarata et al., 2020; Elhage et al., 2021).

Elhage et al. (2021) proposed the circuit analysis (introduced in Section 3) to examine the operation of attention heads, and they found induction heads in attention-only models. Olsson et al. (2022) further investigated the correlation between the formation of induction heads and ICL. Bansal et al. (2023) observed an overlap between the set of induction heads and the set of important attention heads for ICL. Using circuit analysis, Wang et al. (2023) also found some attention heads performing the function of identifying/removing names in the indirect object identification task. Other studies (Lieberum et al., 2023; Geva et al., 2023; Mohebbi et al., 2023) intervened the attention or FFN layers to study their functions. In this paper, we leverage the circuit analysis to investigate semantic relationships in attention heads.

3 Semantic Induction Head

Preliminary. Elhage et al. (2021) rewrite the operation of a multi-head attention (MHA) layer containing h attention heads as follows.

$$\begin{aligned}&\sum\nolimits_{h=1}^{H} \mathit{softmax}\left(\boldsymbol{x}W_q^h(\boldsymbol{x}W_k^h)^T/\sqrt{d_h}\right)\boldsymbol{x}W_v^hW_o^h\\ =&\sum\nolimits_{h=1}^{H} \mathit{softmax}\left(\boldsymbol{x}W_{QK}^h\boldsymbol{x}^T/\sqrt{d_h}\right)\boldsymbol{x}W_{OV}^h\end{aligned} \tag{1}$$

where $\boldsymbol{x} = [x_1^T, x_2^T, \ldots, x_N^T]^T \in \mathbb{R}^{N\times d}$ denotes the embedding sequence, and $x_i = t_iW_e \in \mathbb{R}^{1\times d}$ is the embedding of the i-th input token t_i. $W_e \in \mathbb{R}^{|\mathcal{V}|\times d}$ denotes the embedding layer over a vocabulary $\mathcal{V}$. $W_q^h, W_k^h, W_v^h \in \mathbb{R}^{d\times d_h}$ denote the query, key, and value transformations in the h-th attention head. The output transformation W_o can be decomposed as $W_o = [(W_o^1)^T\ (W_o^2)^T\ \ldots\ (W_o^H)^T]^T$, where $W_o^h \in \mathbb{R}^{d_h\times d}$.

In Equation (1), $W_{QK}^h = W_q^h(W_k^h)^T$, termed the Query-Key (QK) circuit, is responsible for computing the attention pattern of the head, thus determining the head prefers to attend to which token. On the other hand, the matrix $W_{OV}^h = W_v^hW_o^h$, termed the Output-Value (OV) circuit, computes the independent output of each head at the current token regardless of the attention pattern. The output of the OV circuit can be projected back to the vocabulary as $\boldsymbol{x}W_{OV}^hW_u$ by the unembedding transformation $W_u \in \mathbb{R}^{d\times|\mathcal{V}|}$. The projected vector represents the influence of the attention head on the output. Importantly, according to (Elhage et al., 2021), both the QK circuit and OV circuit are directly performed on input embeddings, facilitating the understanding of operations in attention heads.

Based on the above decomposition, Elhage et al. (2021) identify a specific behavior in attention heads, which they refer to as *induction heads*. They observe this behavior in attention heads when presented with sequences like "[A] [B] $\cdots$ [A]". In these induction heads, the QK circuit causes the attention head to attend to the token [B], which appears next to the previous occurrence of the current token [A]. This behavior is termed prefix matching. Then, the OV circuit increases the output logit of the attended token [B], termed *copying*. This mechanism is shown in Figure 1.

Main experimental setup. We use the open-sourced InternLM2-1.8B[1], which contains 24 layers and each layer consists of 16 attention heads. We use the Abstract GENeration DAtaset (AGENDA)[2] (Koncel-Kedziorski et al., 2019) for testing because it contains well-annotated relations between entities. The test set of the AGENDA dataset has a total of 1,000 samples, each consisting of a knowledge graph and a corresponding paragraph that describes relations in the knowledge graph. For syntactic dependencies, we split paragraphs in the AGENDA dataset into individual sentences, and use spaCy (Honnibal et al., 2020) to extract dependencies between tokens.

3.1 Syntactic Dependency in Attention Heads

In this section, we explore whether attention heads encode more complex knowledge beyond copying. We first study a basic pairwise relation inherent in natural language, syntactic dependency, representing the grammatical structure of a sentence.

We focus on three frequent types of syntactic dependencies: subject-predicate (*subj*), predicate-object (*obj*), and modifier-noun/verb (*mod*). Each relation is represented as a triplet: $T =$

[1] https://huggingface.co/internlm
[2] https://github.com/rikdz/GraphWriter

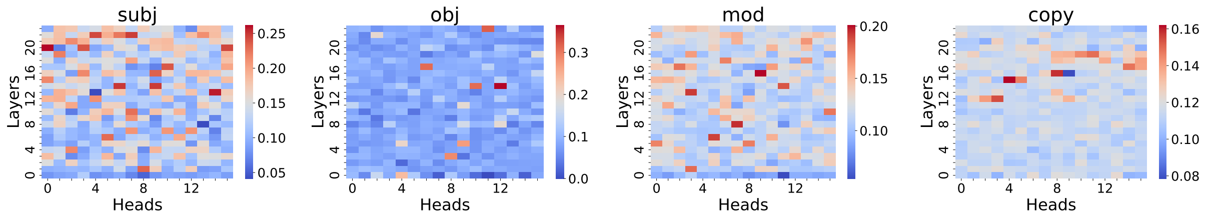

Figure 2: Heatmaps of the average relation index of attention heads for syntactic dependency between tokens and the heatmap of the average copying score (Bansal et al., 2023) of attention heads.

$(t_s, relation, t_o)$, $relation \in \{subj, obj, mod\}$. t_s denotes the token in the head node, termed the *head token*, where s denotes its index in the input sequence. t_o represents the token in the child node, termed the *tail token*, where o represents its index in the sequence.

To examine whether attention heads encode dependency relationships between tokens, we use the OV circuit to analyze the influence of attention heads on output logits of tail tokens when attending to head tokens. Given an input sequence $[t_1, \dots, t_n]$ and the triplet $T = (t_s, relation, t_o)$, we measure how much each head h raises the tail token t_o when attending to the head token t_s.

First, we look for attention heads that attend to the head token via the QK circuit. Given the current token t_j, let $A_j^h = softmax(x_j W_{QK}^h \boldsymbol{x}^T)$ denote the attention probability of the h-th attention head over all tokens. If the head h attends from t_j to the head token t_s with a high probability, *i.e.,* $s = \arg\max_{1 \le k \le j} A_{j,k}^h$ and $A_{j,s}^h / \max_{k \ne s}\{A_{j,k}^h\} > \tau$, we consider this head as a potential candidate for representing the triplet associated with the head token t_s. Otherwise, we skip this head on this triplet. We set $\tau = 2.2$ in experiments[3].

Second, we examine whether these heads raise the tail token t_o by computing the projection of the OV output on the vocabulary, $x_j W_{OV}^h W_u \in \mathbb{R}^{|\mathcal{V}|}$. Similar to (Bansal et al., 2023), we first compute the output probabilities of tokens as $\boldsymbol{p}^{h,j} = softmax(x_j W_{OV}^h W_u) \in \mathbb{R}^{|\mathcal{V}|}$. Then, we extract the probability $p_{t_k}^{h,j} (k \le j)$ of each token t_k before the token t_j . Here we suppose all tokens t_k before t_j are unique for simplification, and if several positions share the same token (*e.g.,* $t_{k_1} = t_{k_2}$), we only consider it once. These probabilities are further transformed by subtracting their mean value and ruling out values smaller than zero, as follows.

$$q_{t_k}^{h,j} = \max(0, p_{t_k}^{h,j} - \mathbb{E}_{1 \le k' \le j}[p_{t_{k'}}^{h,j}]) \tag{2}$$

[3]Please refer to Appendix A for discussions about the setting of τ.

This transformation helps to focus on tokens whose output probabilities are raised. Then, we compute the following ratio $a_T^{h,j}$ to measure the significance of raising the tail token t_o relative to all tokens before t_j.

$$a_T^{h,j} = q_{t_o}^{h,j} / \sum_{k=1}^{j} q_{t_k}^{h,j} \tag{3}$$

Finally, for each head h, we average the relation index $a_T^{h,j}$ across all current tokens t_j and across all triplets T. Note that we only consider current tokens after the head and tail tokens, *i.e.,* $j \ge \max(s, o)$.

Model's ability in understanding dependencies. Before examining whether attention heads encode dependencies between tokens, we first test the model's overall proficiency in learning and understanding dependency relationships. We follow Clark et al. (2019) to train an attention-and-words probing classifier, which takes the word embeddings and the attention weights extracted from InternLM2-1.8B as input and fits the probability of each token being the syntactic head of another token. We train the classifier on 200 sequences from the AGENDA dataset, each with a length of less than 32. Then, we evaluate the accuracy of the predicted head positions on another 100 sequences. For 52% tokens in the input sequence, the classifier can identify the position of their head tokens based on attention weights extracted from InternLM2-1.8B. This accuracy is significantly higher than the random guess, indicating that InternLM2-1.8B can well understand syntactic dependencies. Therefore, we are motivated to further study dependencies in its attention heads.

The *subj* and *obj* dependencies are encoded in attention heads. Figure 2 shows heatmaps of the relation index of each attention head *w.r.t.* three types of dependency relationships. As a baseline, we also compute the relation index when setting tail tokens in all triplets to the 10th token. Such triplets do not represent any relationships, and the relation

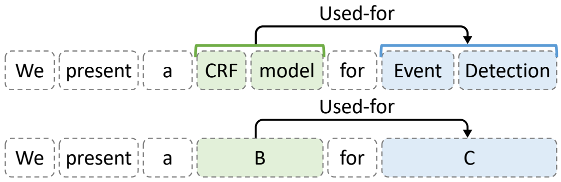

Figure 3: We replace the entities in the sentence (the first row) with capital English letters (the second row).

indexes of attention heads are lower than 0.1. In comparison, for the *subj* and *obj* dependencies, the attention heads exhibit relation indexes close to 0.3, and the relation index *w.r.t.* the *mod* dependency is a bit lower. This suggests the model may have better learned the *subj* and *obj* dependencies than *mod*. In Section 4.1, it is also observed that the model exhibits better performance on the justification of the *subj* and *obj* dependencies than the *mod* dependency, coinciding with this discovery. Furthermore, relation indexes for the *obj* dependency are more sparsely distributed than those for *subj* dependency. There are about ten attention heads that exhibit salient values for the *obj* dependency. This indicates that the model may store the *obj* dependency using a few attention heads, while the *subj* dependency is widely encoded in more attention heads. These observations highlight the varying degrees of the model's understanding and representation of different dependency relationships.

3.2 Semantic Relationship in Attention Heads

In addition to syntactic dependencies, we also study semantic relationships between entities in knowledge graphs. There are seven types of relations between entities in the AGENDA dataset: Part-of, Compare, Used-for, Feature-of, Hyponym-of, Evaluate-for, and Conjunction.

Similar to dependencies in Section 3.1, we also expect to represent each relation between entities as a triplet $T = (t_s, \textit{relation}, t_o)$. Because the entities in sentences often consist of multiple tokens, we replace them with capital English letters[4] as shown in Figure 3. By representing each entity with a single letter, we can directly adopt the metric in Equation (3). Besides, we remove the most frequent function words annotated by spaCy in the sentence. This step helps to reduce noise and focus on the more informative content words.

Various semantic relationships are encoded in attention heads. Figure 4 illustrates relation indexes of attention heads *w.r.t.* seven types of semantic relationships. In contrast to syntactic dependencies, semantic relationships exhibit clearer patterns within attention heads. Each type of relationship is represented by a range of 5 to 15 attention heads. Interestingly, certain relationships, such as "Used-for," "Hyponym-of," and "Conjunction," appear to be more clustered in specific attention heads. These findings suggest that the model possesses a capacity to represent various semantic relationships in attention heads. Furthermore, considering these semantic relationships are bidirectional, we also analyze the reverse relation triplet ($\tilde{T} = (t_o, \textit{relation}, t_s)$) in attention heads in Appendix B. Results in Figure 16 show that some attention heads store both directions of the relationship, reflecting the model's ability to understand the reciprocal nature of these semantic relationships.

4 In-Context Learning and Semantic Induction Heads

In this section, we investigate the correlation between in-context learning and semantic induction heads. We first categorize the ICL ability into three levels and observe the gradual emergence of different levels of ICL. Then, we investigate the formation of semantic induction heads in the training process to understand the emergence of ICL.

4.1 In-Context Learning of Different Levels

In-context learning refers to the ability to learn from the context to perform an unseen task. However, there is no standard measurement for the ICL ability of LLMs. Kaplan et al. (2020); Olsson et al. (2022) consider ICL as the ability to better predict later tokens in the context than earlier tokens. Another more widely adopted definition of ICL follows a few-shot setting. In this setting, language models are provided with a few examples within the context of the prompt, and the model can better perform the task with more examples given. This definition emphasizes the model's ability to extract and generalize the information in the context.

We rethink the ICL ability from the perspective of what the model has learned from the context. The loss reduction considered in (Olsson et al., 2022) only demonstrates that the context does help models make predictions, but it is unclear what the model has learned. Chomsky (1957) has proposed that when humans learn new languages, they initially grasp the surface structure of the language

[4]We exclude special letters like A,I,N,S,W, and E, which often appear alone and are meaningful alone.

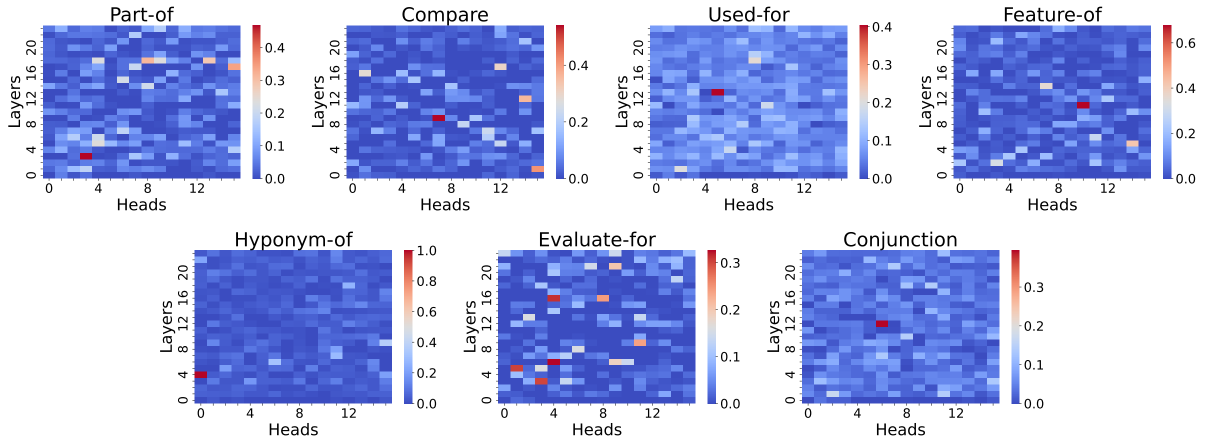

Figure 4: Heatmaps of the average relation index of attention heads for semantic relationships in knowledge graphs.

Task	Entity set	Template
Binary classification	(fruit, month), (furniture, profession)	<E1, E2>: 0; <E2, E1>: 1
Four-class classification	(fruit, month)	<E1, E1>: 0; <E1, E2>: 1; <E2, E1>: 2; <E2, E2>: 3
Nine-class classification	(fruit, animal, month)	<E1, E1>: 0; <E1, E2>: 1; <E1, E3>: 2; <E2, E1>: 3; $\cdots$
Relation justification	(*subj*, *verb*), (*verb*, *obj*), (*mod*, *obj*), (part, whole)	<E1, E2>: true ; <animal, month>: false

Table 1: We construct toy tasks for evaluating the ICL ability of models. E1, E2, and E3 in the template refer to instances belonging to the 1st, 2nd, and 3rd categories in each pair of entity sets. For example, binary classification contains inputs like "apple, January: 0" and "April, orange:1".

before delving into the deep structure. Inspired by this, we hypothesize that LLMs also first learn the surface format of the context, and then gradually comprehend the deep patterns or rules within the context. Based on this hypothesis, we categorize the ICL ability into three levels:

- Loss reduction: This level of ICL is characterized by a reduction in the loss of tokens as the model predicts later tokens in the context. Olsson et al. (2022) demonstrates ICL at this initial level.
- Format compliance (few-shot): At this level, the model learns the format of examples in the prompt (*e.g.,* numbers and symbols), and generates outputs following the same format. Although the outputs have the correct format, the predictions may be incorrect.
- Pattern discovery (few-shot): This level expects the model to recognize and comprehend the underlying pattern within the examples, and apply it consistently to generate the correct prediction.

By categorizing ICL into these levels, we can systematically assess the progression and development of the model's ICL abilities.

Model. To study the ICL ability of the model during the training process, we train a model from scratch using the InternLM framework (Team, 2023). The model contains 20 transformer layers, and each layer consists of $H = 16$ attention heads.

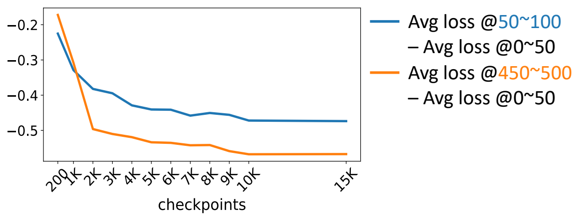

Figure 5: The average difference in loss between later tokens and early tokens decreases from the very beginning of the training.

The hidden size of the model is $d = 2048$, thus each head has a dimension of $d_h = 128$. The model is trained using the SlimPajama dataset (Soboleva et al., 2023) on 32 GPUs, and the batch size on each GPU is 128K tokens. We train the model for 40k steps, with checkpoints saved every 200 steps to monitor the model's progress in relationship representing and ICL during training.

Measurements and results. We assess the ICL ability at each level using the following methods.

For loss reduction, we follow Olsson et al. (2022) but with a minor modification that improves the statistical stability. Specifically, we adjust the formula from the loss at the j-th token minus the loss at the i-th token ($i \leq j$), to the averaged loss over the interval from the $j \sim (i + j)$-th tokens minus the averaged loss over the $0 \sim i$-th tokens.

We sample 100 sentences from the SlimPajama

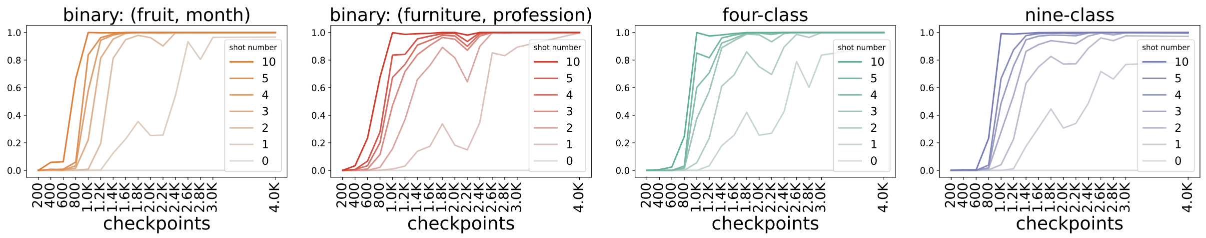

Figure 6: Format accuracy of different tasks at different checkpoints. Each line represents the format accuracy with different numbers of shots (examples) in the prompt.

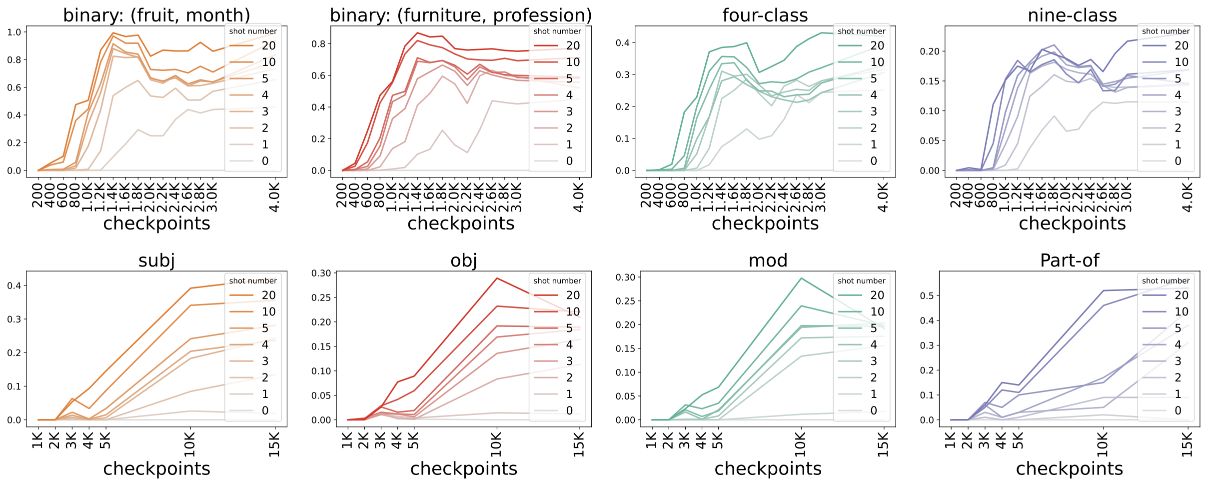

Figure 7: Prediction accuracy of different tasks at different checkpoints. Each line represents the prediction accuracy with different numbers of shots (examples) in the prompt. It is worth noting that these tasks cannot be simply considered as binary classification tasks, which is discussed in Appendix D.

dataset and we set $(i, j) \in \{(50, 50), (50, 450)\}$ to measure the loss reduction at different training checkpoints. Figure 5 shows that the loss difference between later and early tokens decreases quickly from the very beginning of the training. This suggests that from the beginning of the training, the model progressively improves its ability to leverage longer contexts for better predictions.

For format compliance, we construct classification tasks in Table 1. We adopt the few-shot setting of ICL, and the prompt is designed to include several examples followed by a query. Here we ensure that when the number of examples exceeds the number of classes, there is at least one example of each class in the prompt. To test the format compliance ability of the model, we force the model to generate only one token. If the generated token is also a number, matching the format presented in the examples, we consider it to have a correct format. We compute the accuracy of the format to measure the format compliance ability of models.

Figure 6 reports the format accuracy given different numbers of shots at different training checkpoints. For two binary classification tasks, we observe that the model's format accuracy progressively improves as the number of shots increases, starting from the 400th step. This suggests that the model's format compliance ability emerges at the early training stage, and it is independent of the entities involved in the task. For the four-class and nine-class classifications, the model gains improvement with an increasing number of shots at later stages of training (the 600th step and the 800th step, respectively). This indicates that the format compliance to more difficult tasks tends to appear at later training stages. Despite that, the model consistently achieves around 100% format accuracy when using 20 shots at the 1k-th step, and achieves a high accuracy with only one shot after 3k steps. We also measure the format compliance ability from the perspective of the minimum number of shots required to achieve a format accuracy of over 80%. Please refer to Appendix C for details.

For pattern discovery, we still use the above classification tasks, but the difference is that we compute the accuracy of the predicted label. If the model can generate correct labels, we consider it to have successfully discovered and applied the underlying pattern in the prompt. Besides, we also construct four relation justification tasks in Table 1,

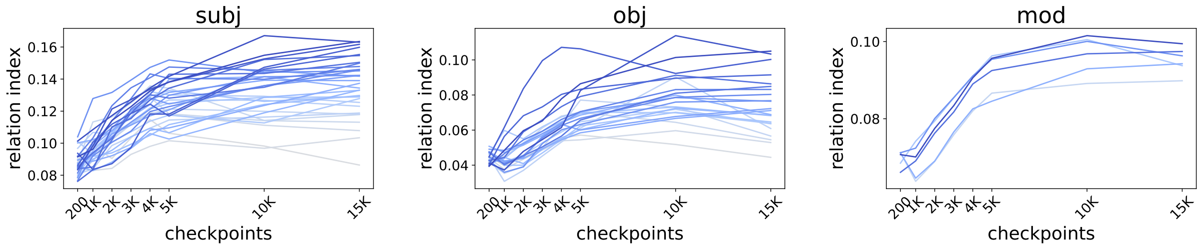

Figure 8: The change curve of average relation indexes of attention heads for syntactic dependency. Each line in the figure represents the relation index of an attention head over training time, and lines are colored according to the value at the 15k-th step.

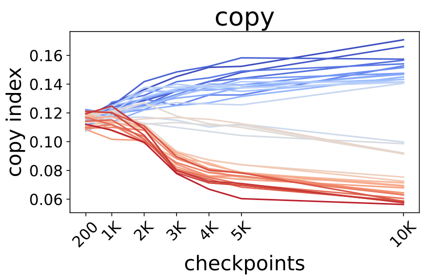

Figure 9: The change curve of copying scores of attention heads. Each line in the figure represents the copying score of an attention head over training time, and lines are colored according to the value at the 10k-th step.

which are related to the syntactic dependency and semantic relationship studied in this paper.

Figure 7 reports the prediction accuracy of the model at different checkpoints. For classification tasks in the first row, the model achieves considerable accuracy with 20 shots at around 1400 steps, indicating that the pattern discovery ability is mastered later than the format compliance. Furthermore, simple binary classification tasks are learned earlier than complex four-class and nine-class classification tasks. The relation justification tasks in the second row, which are more difficult than classification tasks, are learned at later stages, typically starting from around the 2k-th step. After approximately 10k steps, the prediction accuracy tends to saturate. Figure 13 in Appendix E shows that till the end of the training, even with 100 examples, the model cannot fully learn the pattern in the prompt.

Progressive learning of ICL of different levels. From the above results, we can observe a progressive learning process for the different levels of ICL. The loss reduction happens from the beginning of the training, followed by the emergence of format compliance (after 400 steps), and pattern discovery is mastered in the last (after 1k or 2k steps). This discovery aligns with our hypothesis that three levels of ICL have increasing difficulties. Moreover, within the format compliance and pattern discovery, we observe that the model typically learns more challenging tasks at later training stages.

4.2 Correlation Between Semantic Induction Heads and ICL

In this section, we investigate the formation of semantic induction heads during the training process and discover their correlation with ICL. We compute the average relation index of attention heads over all triplets for each syntactic dependency and each semantic relationship in different checkpoints. Here we only ensure $s = \arg\max_{1\leq k\leq N} A^h_{j,k}$, and do not require $A^h_{j,s} / \max_{k\neq s}\{A^h_{j,k}\} > \tau$ any more, because in the early stage of the training, it is too challenging to find attention heads having an extremely high attention probability on head tokens.

We find that the relation index of some attention heads increases during the same stage as the emergence of the ICL ability. Specifically, Figure 8 shows the change in the relation index for syntactic dependencies of attention heads. We sampled attention heads with an increasing relation index for visualization. It can be observed that relation indexes of some attention heads begin increasing from the beginning of the training, aligning with the emergence of loss reduction. On the other hand, relation indexes of other attention heads begin to increase after around 1k or 2k steps, which coincides with the emergence of the pattern discovery ability. Thus, we infer that the formation of semantic induction heads plays a crucial role in the development of the ICL ability. These semantic induction heads likely contribute to capturing and representing relationships between tokens, which are essential for the ICL ability.

On the other hand, Figure 9 shows the change in copying scores (Olsson et al., 2022; Bansal et al., 2023) of attention heads. Copying scores of some attention heads start increasing from 200 steps, so we infer the copying mechanism is responsible for the loss reduction and the format compliance. It is reasonable because the task used for evaluating

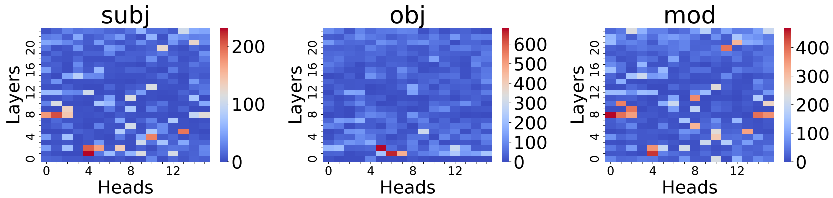

Figure 10: Occurrence of each attention head having the largest value of $\mathbb{E}_j[a_T^{h,j}]$ with $\mathbb{E}_j[a_T^{h,j}] > 0$ over all triplets T of each type of dependency.

the format compliance can be simply achieved by copying the token (“0” or “1”) after the colon in the preceding context to the output. More interestingly, copying scores of other heads begin to drop from the 1K step, where the pattern discovery ability emerges. Therefore, we hypothesize that the copying behavior is not always good for ICL, because sometimes direct copying may cause incorrect predictions.

5 More Discussions about Relationships Encoded in Attention Heads

A notable distinction between copying and relationship is that the relationship between tokens depends on the input context, while copying is context-agnostic. Therefore, different inputs may utilize different attention heads. Thus, we propose to find a common group of attention heads that represent specific relationships in different inputs. For each triplet T, we identify the attention head that has the largest value of $\mathbb{E}_j[a_T^{h,j}]$ (larger than 0). Then, we count the occurrence of each head having the largest value among all triplets T for each dependency relationship.

Figure 10 shows the number of occurrences of each attention head having the largest relation index for syntactic dependencies. Please refer to Appendix G for results on semantic relationships. We find that for each type of dependency/relationship, there are around 5$\sim$15 attention heads frequently activated by different inputs. Besides, different relationships tend to share some common attention heads (*e.g.,* layer2, head4 for syntactic dependencies). This may indicate that these human-defined relationships are not mutually exclusive from the LLMs’ point of view. In other words, there exists a many-to-many mapping between human-defined relationships and attention heads in LLMs.

6 Conclusion

Previous studies (Elhage et al., 2021; Olsson et al., 2022) in mechanistic interpretability only studied the simple functions in very specific tasks (Wang et al., 2023; Lieberum et al., 2023). In this study, we extended the conventional induction heads to analyze high-level relationships between words/entities in natural languages. Our experiments revealed that specific attention heads encode syntactic dependencies and semantic relationships in natural languages. Furthermore, we identified three levels of the in-context learning ability of LLMs, and experimental results showed they are progressively learned during the training process. Finally, we observed a close correlation between the formation of semantic induction heads and in-context learning ability, strengthening our understanding of in-context learning.

Limitations

Limitations of this paper lie in the following three perspectives. (1) While the proposed relation index has the potential to be adapted to different relationships in various languages, this paper only focuses on syntactic dependency and semantic relations in English. We think it is a promising direction to examine the representation of relationships in different languages and leave it to future work. (2) Although we have extended the simple copying operation to complex semantic relationships, the proposed method is limited to relationships between two tokens/entities. (3) Due to limitations in computational resources, we only conduct experiments on $\sim$1B models.

Acknowledgement

This work is supported by Shanghai Artificial Intelligence Laboratory.

References

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2023. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11833–11856, Toronto, Canada. 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, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. *Distill*. Https://distill.pub/2020/circuits.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *USENIX Security Symposium*, pages 2633–2650. USENIX Association.

Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Millon Das, Archit Mangrulkar, Ishan Manchanda, Manav Kapadnis, and Sohan Patnaik. 2022. Enolp musk@SMM4H'22 : Leveraging pre-trained language models for stance and premise classification. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 156–159, Gyeongju, Republic of Korea. Association for Computational Linguistics.

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, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2668–2677. PMLR.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Tom Lieberum, Matthew Rahtz, János Kramár, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

Dongrui Liu, Huiqi Deng, Xu Cheng, Qihan Ren, Kangrui Wang, and Quanshi Zhang. 2024. Towards the difficulty for a deep neural network to learn concepts of different complexities. *Advances in Neural Information Processing Systems*, 36.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing*, pages 4765–4774.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. DecompX: Explaining transformers decisions by propagating token decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2649–2664, Toronto, Canada. Association for Computational Linguistics.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. 2023. Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400, Dubrovnik, Croatia. Association for Computational Linguistics.

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, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Cheonbok Park, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, Yeonsoo Lee, and Jaegul Choo. 2019. Sanvis: Visual analytics for understanding self-attention networks. In *2019 IEEE Visualization Conference (VIS)*, pages 146–150.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Chen Qian, Jie Zhang, Wei Yao, Dongrui Liu, Zhenfei Yin, Yu Qiao, Yong Liu, and Jing Shao. 2024. Towards tracing trustworthiness dynamics: Revisiting pre-training period of large language models. *arXiv preprint arXiv:2402.19465*.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. 2023. Defining and quantifying the emergence of sparse concepts in dnns. In *CVPR*, pages 20280–20289. IEEE.

Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, and Quanshi Zhang. 2021. Towards a unified game-theoretic view of adversarial perturbations and robustness. In *NeurIPS*, volume 34, pages 3797–3810.

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024. Exploring safety generalization challenges of large language models via code. *arxiv preprint arxiv:2403.07865*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70, pages 3319–3328. PMLR.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM`.

Fel Thomas, Picard Agustin, Bethune Louis, Boissin Thibaut, Vigouroux David, Colin Julien, Cadène Rémi, and Serre Thomas. 2023. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

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*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Sen Yang, Shujian Huang, Wei Zou, Jianbing Zhang, Xinyu Dai, and Jiajun Chen. 2023. Local interpretation of transformer based on linear decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10270–10287, Toronto, Canada. Association for Computational Linguistics.

Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. 2024. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):262–272.

Lining Zhang, Mengchen Wang, Liben Chen, and Wenxin Zhang. 2022. Probing GPT-3's linguistic knowledge on semantic tasks. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 297–304, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. 2024. Explaining generalization power of a dnn using interactive concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17105–17113.

A Setting of τ in the relation index.

The setting of the threshold $\tau = 2.2$ for the value of $\frac{A^h_{j,s}}{\max_{k \neq s}\{A^h_{j,k}\}}$ is based on our observation in the distribution of values of $\frac{A^h_{j,s}}{\max_{k \neq s}\{A^h_{j,k}\}}$. Using input sentences and corresponding triplets in the AGENDA test set, we computed the value of $\frac{A^h_{j,s}}{\max_{k \neq s}\{A^h_{j,k}\}}$ at all heads h and all current tokens t_j that satisfy $s = \arg\max_k\{A^h_{j,k}\}$. The distribution of this value is shown in Figure 11. The frequency of values larger than 2.2 dropped significantly to less than 5%. Thus, we empirically set the threshold $\tau = 2.2$ to only focus on attention heads that *exclusively attended to the head token*.

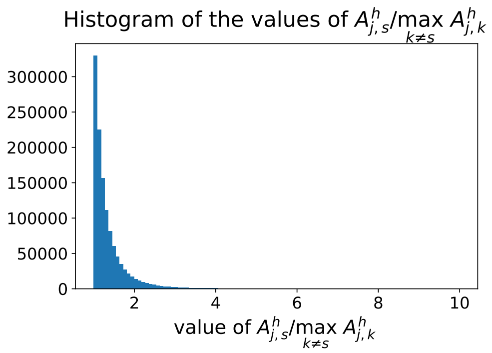

Figure 11: Distribution of $\frac{A^h_{j,s}}{\max_{k \neq s}\{A^h_{j,k}\}}$ in InternLM2-1.8B. For better visualization, we just show the distribution of $\frac{A^h_{j,s}}{\max_{k \neq s}\{A^h_{j,k}\}}$ within the range of 0 10.

Moreover, we also conduct ablation experiments with different values of τ. Specifically, we compute the relation index for syntactic dependencies on InternLM2-1.8B with a smaller value $\tau = 2.0$ and a larger value $\tau = 2.5$, respectively. Heatmaps in Figure 15 show that the setting of τ does not significantly affect the distribution of the relation index. Different settings of τ yield a similar set of semantic induction heads that have a high relation index.

B Relation index for the reverse semantic relationships

The semantic relationships in knowledge graphs are bidirectional, thus we also compute the relation index of attention heads for the reverse semantic relationships. Figure 16 shows the results in InternLM2-1.8B. Comparing Figure 16 and Figure 4, we find that some attention heads represent both directions of the relation.

C Format compliance ability

Besides the format accuracy in Figure 6, we also measure the format compliance ability from another perspective: the minimum number of shots required to achieve a format accuracy of over 80%. A lower minimum number of shots indicates a better format compliance ability. We set a maximum limit of 20 shots. If the model fails to achieve an accuracy of 80% even with 20 shots, we record the result as 20. Figure 12 consistently shows that the model learns format compliance on simple tasks earlier than on complex tasks, but all achieve a good performance at 1000 steps.

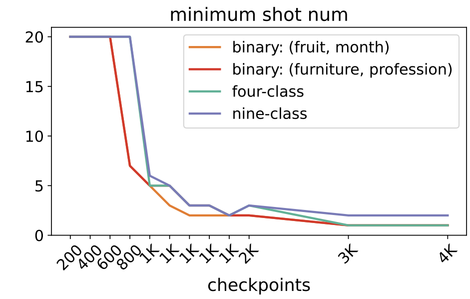

Figure 12: The minimum number of shots (examples in the prompt) required to achieve over 80% format accuracy.

Moreover, considering that the format of generating "true" or "false" in relation justification tasks is different from the simpler format in classification tasks in Figure 6, we additionally examine the format accuracy on relation justification tasks in Figure 17. The format compliance in these tasks emerges from about 2K steps, later than that in simpler tasks in Figure 6, and achieves about 50%-70% at 15K steps.

D Pattern discovery tasks are not binary classification tasks

Unlike binary classification tasks, pattern discovery tasks are actually more difficult for generative models.

First, the model generates the next token as the prediction, which is different from the classic binary task. When generating the next token, there are a total of $|\mathcal{V}| = 92544$ candidates in the vocabulary $\mathcal{V}$. If the model could perfectly comply with the format, the problem is simplified as a binary classification task. However, models in the early stages of training do not have such ideal capabilities yet. For example, Figure 17 shows that the format compliance in relation justification tasks emerges from about 2K steps, later than that in simpler tasks in Figure 6, and achieves about 50%-70% at 15K steps. Thus, the exact prediction accuracy of the model will be lower. Second, the model might inherit certain biases from the training data. Thus, the probability of generating either the token "true" or "false" is not 0.5 vs 0.5.

E Pattern discovery ability of a well-trained model

Although we have observed the development of the pattern discovery ability in Figure 7, we find that it is hard to be fully mastered by the model. Figure 13 reports the prediction accuracy of the well-trained InternLM2-1.8B on classification tasks. Even with 100 examples in the prompt, the model still cannot perfectly recognize and utilize the pattern in the prompt. On the other hand, Figure 14 shows that InternLM2-1.8B achieves higher accuracy on relation justification tasks.

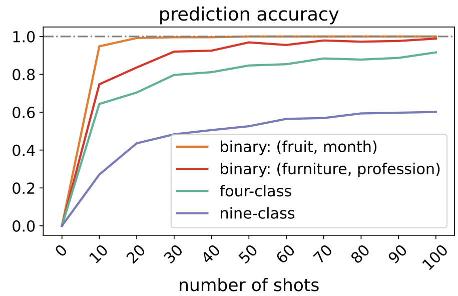

Figure 13: Prediction accuracy of InternLM2-1.8B on classification tasks.

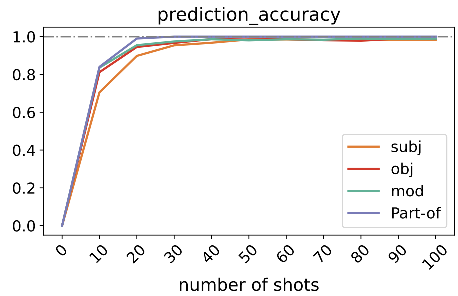

Figure 14: Prediction accuracy of InternLM2-1.8B on relation justification tasks.

F Change of relation index for semantic relationships over training time

This section provides results of the change of relation indexes of attention heads for semantic relationships in knowledge graphs. We consider both directions of the semantic relationship, and results are shown in Figure 18 and Figure 19.

G Grouping attention heads for relations

As discussed in Section 5, the semantic relationships are dependent on the context, so they may be stored in different attention heads given different contexts. In this section, we perform the grouping analysis on semantic relations in knowledge graphs.

Figure 20 and Figure 21 show the occurrence times of each attention head having the largest value of $\mathbb{E}_j[a_T^{h,j}]$ for triplets in semantic relations and the reverse triplets. We observe that some attention heads are commonly highlighted in different types of relationships.

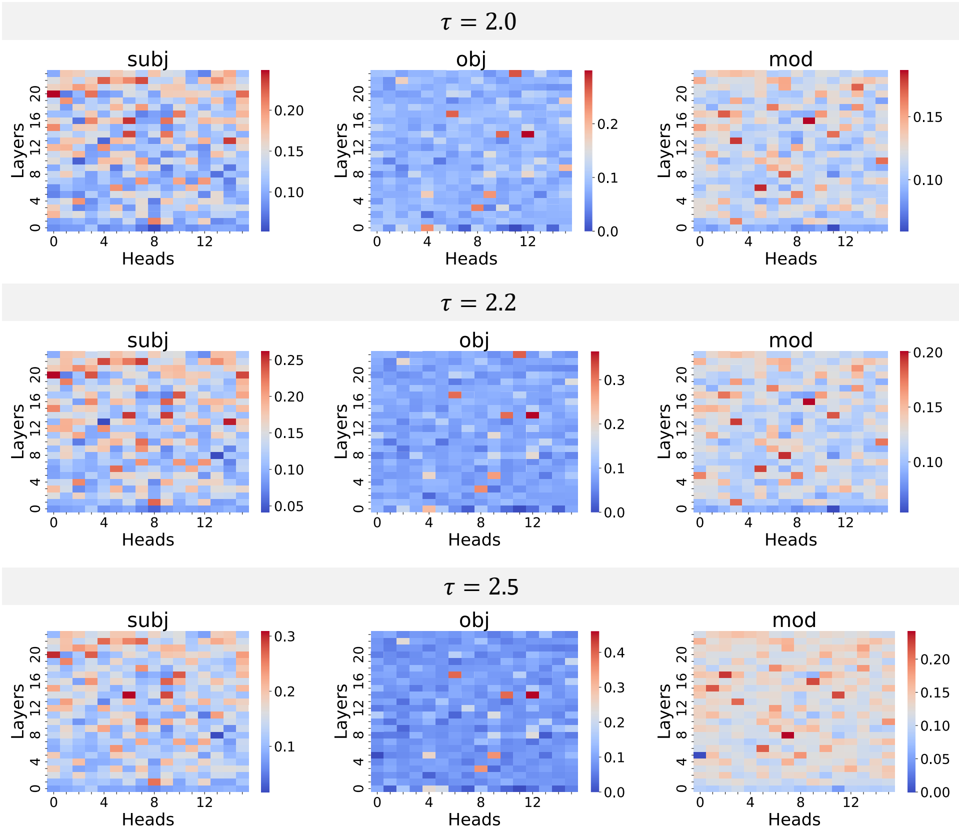

Figure 15: Heatmaps of relation indexes with different settings of τ. Different settings of τ yield a similar set of semantic induction heads that have a high relation index.

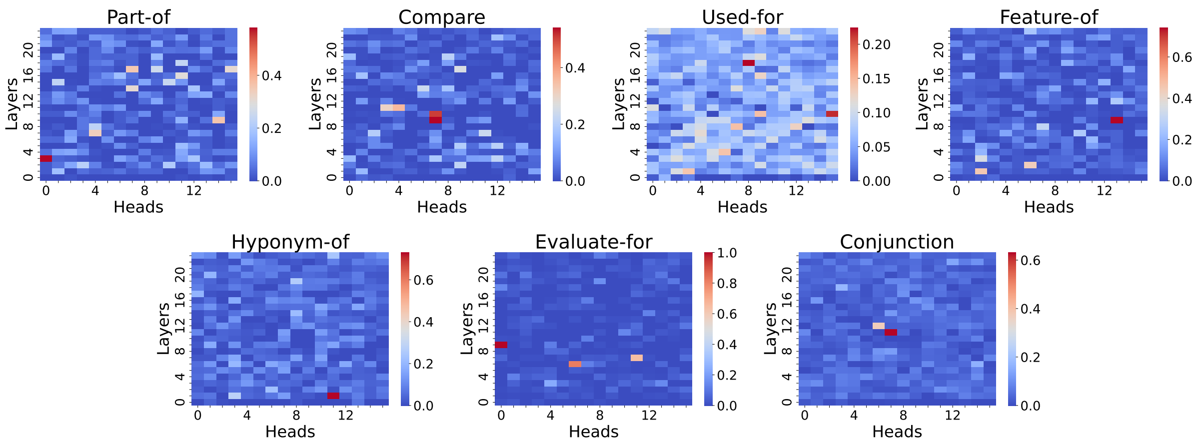

Figure 16: Heatmaps of the average relation index of attention heads for the reverse semantic relationships in knowledge graphs.

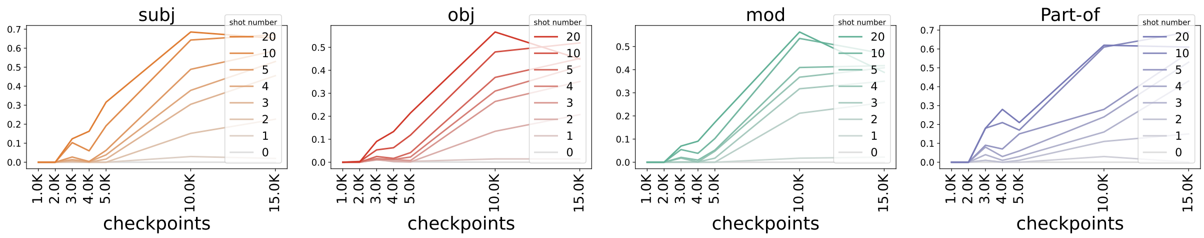

Figure 17: Format accuracy of different relation justification tasks at different checkpoints. Each line represents the format accuracy with different numbers of shots (examples) in the prompt.

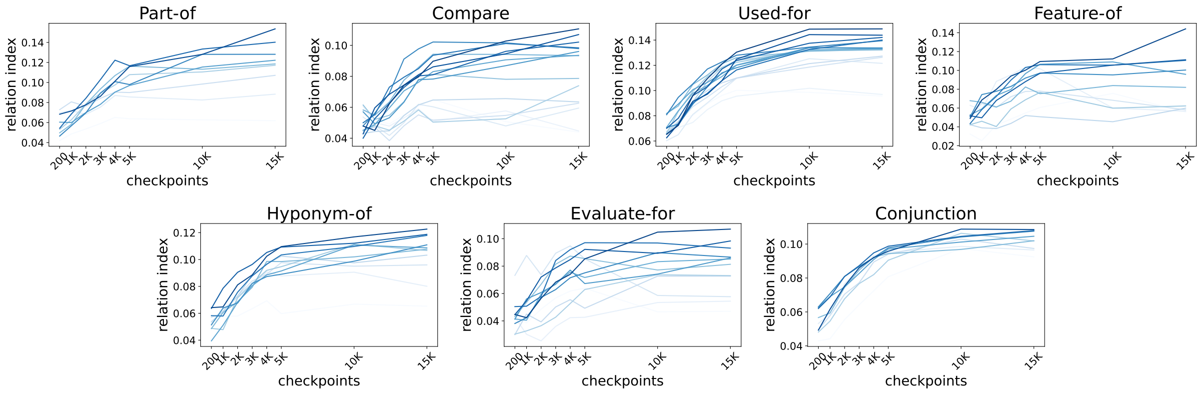

Figure 18: The change curve of relation indexes of attention heads for semantic relationships.

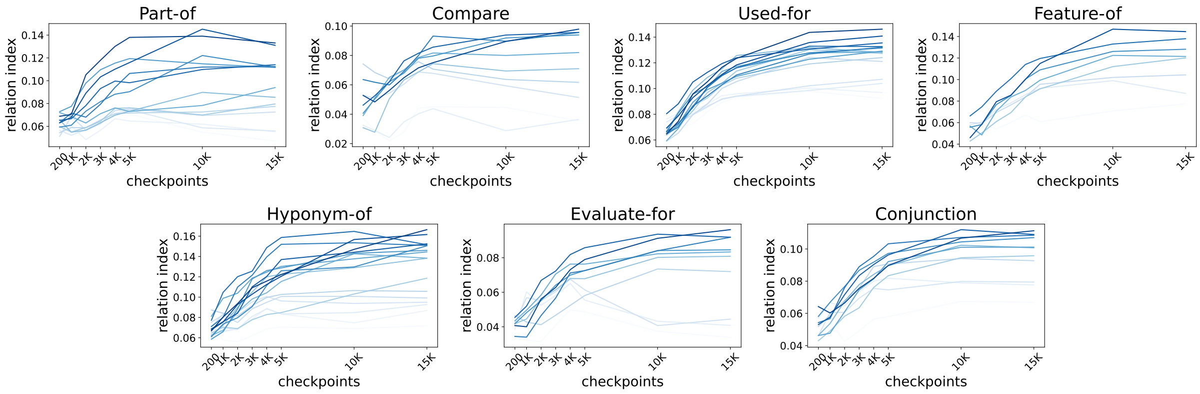

Figure 19: The change curve of relation indexes of attention heads for reverse semantic relationships.

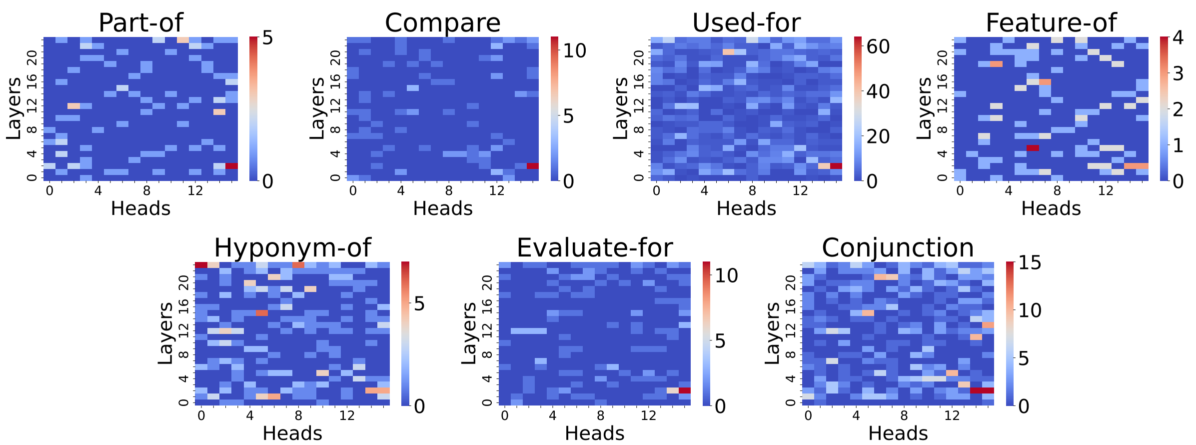

Figure 20: Heatmaps of occurrence of attention heads having the largest value of $\mathbb{E}_j[a_T^{h,j}]$ for each triplet in semantic relations.

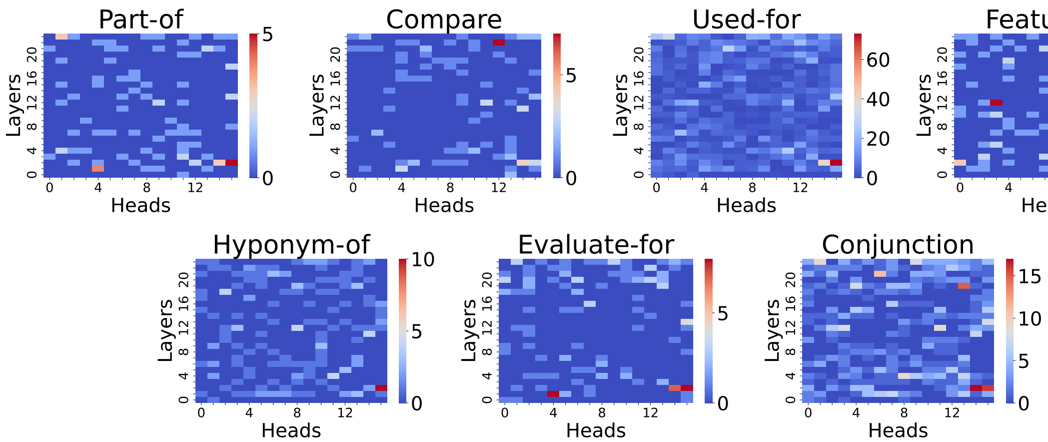

Figure 21: Heatmaps of occurrence of attention heads having the largest value of $\mathbb{E}_j[a_T^{h,j}]$ for each triplet in reverse semantic relations.