Beyond the Induction Circuit: A Mechanistic Prototype for Out-of-domain In-context Learning

趙羽風 [1]　井之上直也 [1,2]
[1] 北陸先端科学技術大学院大学　[2] 理化学研究所　yfzhao@jaist.ac.jp

Abstract

In-**c**ontext **L**earning (ICL) is a promising few-shot learning paradigm with unclear mechanisms. Existing explanations heavily rely on Induction Heads, which fail to account for out-of-domain ICL, where query labels are absent in demonstrations. To address this, we model ICL as attribute resolution, where queries are mixtures of some attributes, and ICL identifies and resolves relevant attributes for predictions. In this paper, we propose a mechanistic prototype using toy models trained on synthetic data, and observe: (1) even 1-layer Transformers achieve non-trivial accuracy, with limited benefit from additional demonstrations, (2) scaling models effectively improve accuracy, and (3) inference operations can be decomposed into label space identification and generalized induction, warranting further exploration.

1 Introduction

In-**C**ontext **L**earning (ICL) [1, 2] is an emerging few-shot learning paradigm: given an input sequence formed like $[x_1, y_1, \ldots, x_k, y_k, x_q]$, where x_is are demonstrations, y_is are label token corresponding to its preceding x_i, and x_q is a query, **L**anguage **M**odels (LMs) predict a label for the x_q by causal language modeling operation, with only parameters pre-trained on wild language dataset. ICL has aroused widespread interest with an unclear mechanism.

Current works on the mechanisms of ICL are largely related to circuit studies based on Induction Heads [3, 4, 5, 6, 7]. As shown in Fig. 1 (left, A), these studies propose that Transformers explicitly *retrieve* demonstration features similar to the query from the context through specialized attention behaviors, subsequently copying these features into the output of the attention layer. While such studies have advanced significantly, they face a critical limitation: when features that can be explicitly retrieved are absent from the context, specifically when the ground-truth label for the query does not appear in the context, this induction head-based methodology loses its explanatory power: in such a scenario (named **O**ut-**O**f-**D**omain, OOD), induction head-based explanation predicts an ICL accuracy of 0, which is obviously not the case.

To address the aforementioned OOD issue, we consider the following: in scenarios where similarity-based retrieval fails, it becomes essential for LMs to *resolve* the query into its required attributes specified by the contextual demonstrations, rather than merely retrieving a similar demonstration and copying its label to produce a correct answer. As shown in Fig. 1 (left, B) for an example, the LM catches the specified attribute "Occupation" and resolves the query on such an attribute. A good beginning in such a direction is *task vectors* [8] in ICL scenario, but more discussion is still beneficial to reveal the detailed operating dynamics.

Therefore, in this paper, we investigate the capacity and operational dynamics of Transformers on the "query resolution" operations. Specifically, we simulate a scenario where multiple attributes of input texts are encoded into feature vectors (as shown in [3, 9]) and resolved into prediction using contextual information. To achieve this, we train toy Transformers on synthetic data as a mechanistic prototype, where: the input feature x_i is represented as a mixture of **Attribute** vectors, with each attribute vector sampled from a Gaussian mixture comprising several clusters, and each cluster corresponds to an **Attribute Value**. A **Task** is then defined as querying the attribute value of a specific attribute. Using this setup, we train toy Transformers to derive preliminary prototypical observations.

Our experiments and subsequent analysis find that: (1) Even a 1-layer Transformer produces a non-trivial result, (2) scaling models effectively improves accuracy, (3) inference operations can be decomposed into label space identification and generalized induction.

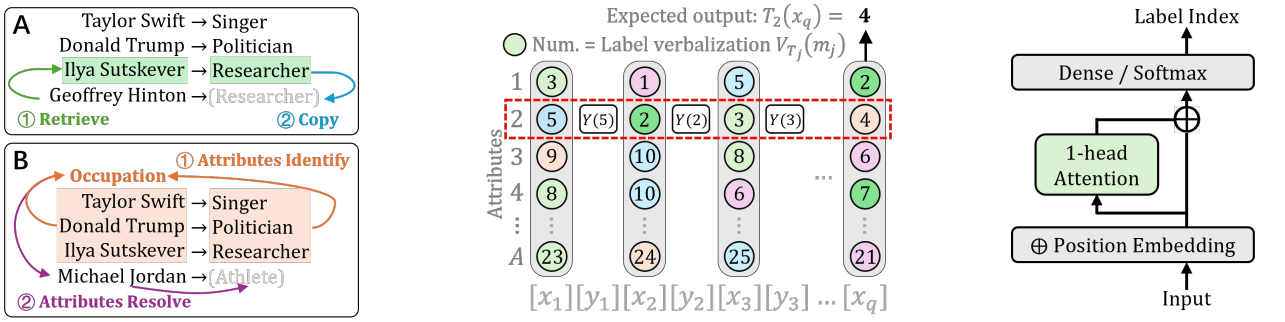

Figure 1 **Left:** (A) An induction-style explanation of ICL processing. LMs first search the same demonstration as the query "Geoffrey Hinton" and copy the subsequent label to the output of the query. When the label "Researcher" is not presented in the context, ICL can not work in this style. (B) A resolve-style explanation of ICL processing investigated in this paper. LMs first identify the attribute ("Occupation") specified by the demonstrations, and resolve the query into this attribute. **Middle:** A diagrammatic sketch of the data synthesis. Each train/test data is an ICL-formed sequence with input feature x_is and labels y_is. Each x_i is a mixture of several attributes, and y_i specifies the attribute to be resolved. **Right:** Model structure used in this paper.

2 Experiment Settings

As mentioned before, we train toy Transformers on synthetic data. Now, we introduce the experiment details.

2.1 Data Synthesis

Our experiments are conducted on synthetic sequence data formed like $[x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_q]$, with single time-step vectors synthesized following these rules:

Input feature and query x_i. Each x_i is a d-dimentional mixture of A attribute vectors, and each attribute vector a_j is sampled from a Gaussian mixture with C clusters in a d_a dimensional space defined by an orthonormal up-projection metrix $U_j \in \mathbb{R}^{d \times d_a}$:

$$x_i = \sum_{i=0}^{A} U_j a_j, a_j \sim \frac{1}{C} \sum_{k=0}^{C} \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \qquad (1)$$

where: any two U_js are orthogonal (which requires $Ad_a \leqslant d$), each centroid $\boldsymbol{\mu}_k$ is sampled in a d_a-dimensional Gaussian distribution with mean of $\mathbf{0}$ and covarience of $3\mathbf{I}$, and the sampling covarience $\boldsymbol{\Sigma}_k$ is fixed to $0.1\mathbf{I}$.

Task T_j. Intuitively, given a sampled vector x_i from the afore-defined process, for each attribute a_j that constitutes it, we can determine the maximum likelihood Gaussian cluster index m_j (called Attribute Value of attribute a_j). Repeat this process for every a_j, we can sequentially producing a vector $[m_1, m_2, \dots, m_A]$ composed of A indices. A task T_j is defined as an inquiry on the x_i to output the j-th attribute's attribute value m_j, that is, $T_j(x_i) = m_j$. For a vector composed of A attributes, we can define A tasks, each corresponding to a specific attribute, collectively forming a task family $\mathcal{T}_A$.

Label vector y_i. We define the *label verbalization* as a discrete representation of $T_j(x_i)$ as follows: (1) For each of the C possible attribute value (denoted as $m_j \in \{1, 2, \dots, C\}$) of a task T_j, we generate an index as the label verbalization $V_{T_j}(m_j)$ to represent it, which span a C-dimensional label space. (2) To prevent shortcut learning (discussed further below), we divide the task family into $\lceil A/B \rceil$ groups, each consisting of up to B tasks. Within each group, all tasks share the same label space. For example, if $A = 4$ and $B = 2$, the tasks can be divided into 2 groups: $\{T_1, T_2\}$, and $\{T_3, T_4\}$, then, if given the $T_1(x) = T_2(x)$, we have $V_{T_1}(T_1(x)) = V_{T_2}(T_2(x))$. So, for a task family of A tasks, we can have a total label space $\mathbb{Y}$ of size $\lceil A/B \rceil C$. Then, for each label l in $\mathbb{Y}$, we sample a vector from d-dimensional normal distribution as the *Label Vector* $y_i = Y(l)$ as the dense representation of the label.

Input sequence. To build one input sequence, we randomly[1] sample $(k+1)$ input features as $\{x_1, x_2, \dots, x_k, x_q\}$, and a task T_j. As shown in Fig. 1 (Middle) for a $B = 2$ scenario, for each input feature, we build label vectors $y_i = Y\left(V_{T_j}(x_i)\right)$, and combine them with formation $[x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_q]$ as an input sequence (where $x_{1:k}$s are the demonstrations, and x_q is the query), and $V_{T_j}(x_q)$ as the expected label. We train Transformer models (§2.2) on such input-label pairs.

Default parameters. Unless specified, we use $k = 4$, $A = 16$, $B = 4$, $C = 16$, so that a label space of size $|\mathbb{Y}| = 64$; and $d = 256$, $d_a = 16$. We use standard unit vectors to span the up-projection U_js.

1) Notice we do not force an OOD condition since it can lead models to learn to only output labels absent from the context.

2.2 Model and Training

Model. Unless specified, we use 1-layer Transformer-formed attention with 1 head (Fig. 1 (Right)). A 32-dimensional one-hot position embedding is concatenated to the input, so the final dimensionality d_m is 288. In some experiments, we increase the number of attention heads, but they always divide the 288 dimensions equally without additional parameters.

Training. We generate a total of $n = 819200$ input data instances by the aforementioned pipeline. We use a standard SGD optimizer, with a learning rate 0.01 and batch size 128 to conduct full-precision training. No learning rate decay, regularization, or momentum are used. Validation data is sampled under the same distribution as training data.

3 Results

One layer Transformer resolves query to the specified attribute. We plot the validation accuracy along the training processing as shown in Fig. 2, where non-trivial accuracy can be observed in 1-layer Transformers. In detail, compared to the random baselines and ablation experiments, where (1) demonstrations are ablated ($k = 0$) to block the model from identifying the task information; (2) input features and queries are ablated ($x = \mathbf{0}$) to block the model from resolving the inputs: when the demonstrations specify the $x \mapsto y$ correlation (Standard $k = 4$), the model predicts the label relatively accurate. Moreover, as shown in Fig. 3, increasing the number of demonstrations does not significantly improve accuracy. Such a sign gives a conclusion, as even one demonstration nearly specifies the attribute to be resolved, with additional demonstrations only marginally reducing minor ambiguities.

Capacities of various model scales. In Fig. 2, we observed a non-trivial accuracy while not ideal, therefore, we are curious about whether a larger or more complex model can act better, so we repeat the experiments on 2-layer and 8-head settings with more training epochs. The orthogonal experiment results are shown in Fig. 4, where the 2-layer 8-head result significantly outperforms with an obvious phase transition (discussed later), and the remaining results are almost equal, which suggests that: (1) 2-layer Transformer may conduct different inference operations, (2) multi-head attention is an essential component for ICL.

Operations of attribute resolution. Then, we attempt

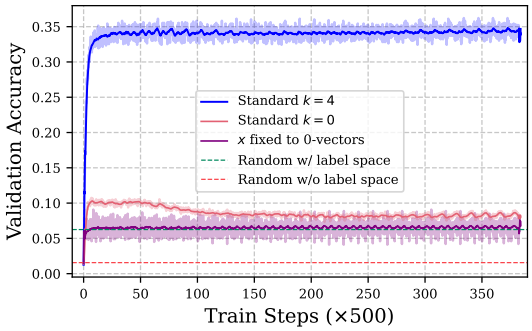

Figure 2 The training dynamics of the **standard experiments** and some reference experiments: (1) **Standard $k = 0$**: trained/tested on sequence without demonstrations. (2) **x fixed to 0 vectors**: trained/tested on sequence where x_is are fixed to 0. (3) **Random w/ label space**: Random prediction inside the label space, i.e., 1/16. (4) **Random w/o label space**: Random prediction inside the label space, i.e., 1/64.

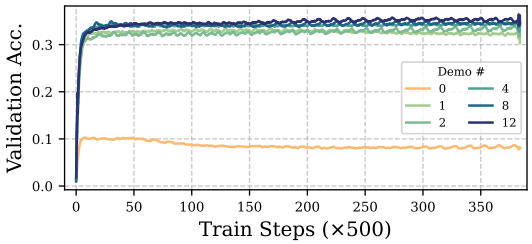

Figure 3 The training dynamics with various k.

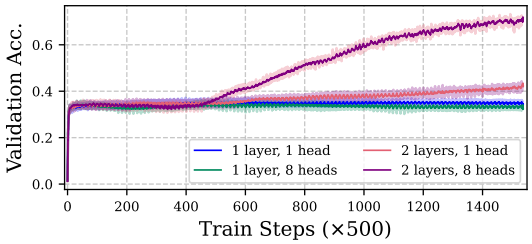

Figure 4 The training dynamics on 4 model specifications.

to investigate the mechanism for the attribute resolution, and preliminarily observe 2 key operations. **(1) Label space indentification.** As a necessary condition for an accurate inference, the model should identify the candidate labels w.r.t. the given context. Shown in Fig. 5 for some cases, when the labels are given in context, even if the information of x is absent, the model can correctly identify the label space to be outputted. Moreover, as a closer observation, we conduct principal component analysis on line vectors of the output dense layer (see Fig. 1, each line vector corresponds to an output un-embedding) as shown in Fig. 6, clear clusters are observed within the same label group (notice that we have 4 16-label groups), suggesting that the model learns the label space information in the training processing, which is aligned with previous

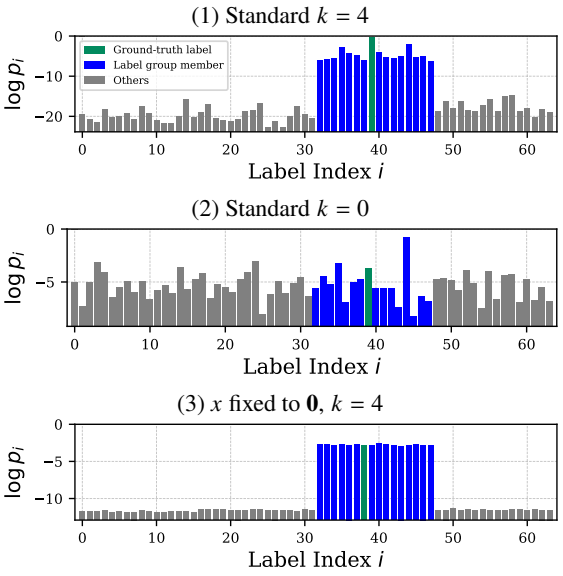

Figure 5 Output logits visualized on 3 settings aligned with Fig. 2, each for a input case. The model can utilize the contextual label information to identify the correct output label space (1, 3), and while no label information is given, the model can not significantly identify the label space (2).

work [10]. **(2) Generalized induction.** We visualize the attention behavior of (A) the end checkpoint of the default model and (B) two checkpoints before and after the phase transition of the 2-layer and 8-head model, as shown in Fig. 7, where: in the 1-layer model and the 2-layer model before the phase transition, the information flow from the x_is to the query dominates, and after the phase transition of the 2-layer model, the information flow from a y_i to the query dominates. This clearly indicates two different mechanisms, and the one that focuses on label y can achieve better accuracy. Since no ground-truth labels are presented in the context in this case, we believe that such an induction-like operation is an essentially novel, or *generalized induction*, which is worthy of further exploration.

Novel attributes cannot be resolved. To simulate scenarios where the tasks specified by the demonstrations are unseen during training, we resample $\boldsymbol{\mu}_k$ to generate novel validation samples. As shown in Fig. 8, the model achieves near-random accuracy on these novel samples, which is expected given the difficulty of responding to unknown attributes. This highlights the In-weight Learning [5, 6, 11] characteristic of OOD ICL.

4 Discussion

Summary. This paper introduces the attribute-resolving explanation of ICL with prototypical observations.

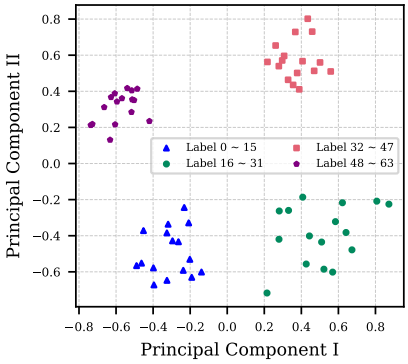

Figure 6 Output embeddings visualized.

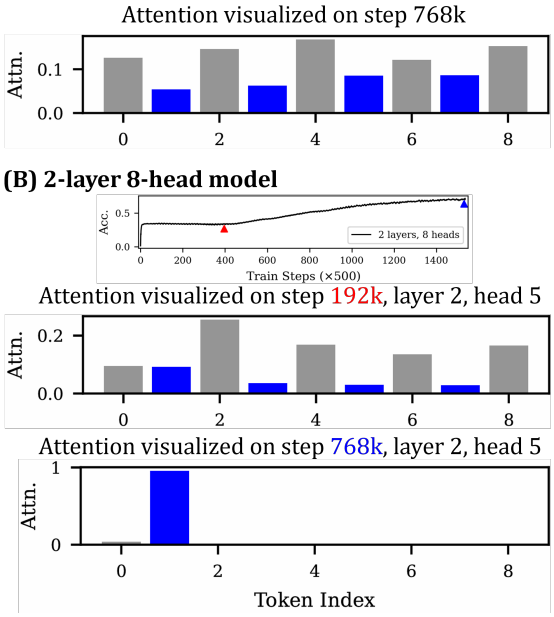

Figure 7 Attention scores from the last token (as the attention query) on an OOD input case, label tokens are in blue.

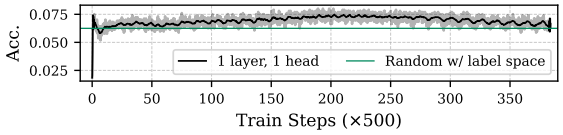

Figure 8 The training dynamics evaluated on distribution-shifted data.

Clues for future works. Future research should look closer at the proposed mechanism in real-world language models and explore connections with other theoretical prototypes for OOD ICL, such as in-context regression [12, 13]. Additionally, it would be valuable to investigate how more complex structures in actual language models, such as FFN blocks within standard Transformer layers, contribute to or interfere with the proposed operations. By laying a foundation, this paper opens up possibilities for such exciting and impactful research.

謝辞

本研究は，JST 創発的研究支援事業 JPMJFR232K，および JSPS 科研費 19K20332 の助成を受けたものです．

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[2] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. **arXiv preprint arXiv:2301.00234**, 2022.

[3] Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. **arXiv preprint arXiv:2410.04468**, 2024.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. **Transformer Circuits Thread**, Vol. 1, No. 1, p. 12, 2021.

[5] Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie CY Chan, and Andrew M Saxe. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. **arXiv preprint arXiv:2404.07129**, 2024.

[6] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In **The Twelfth International Conference on Learning Representations**, 2024.

[7] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 9840–9855, 2023.

[8] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 9318–9333, 2023.

[9] Hakaze Cho, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, and Naoya Inoue. Token-based decision criteria are suboptimal in in-context learning. **arXiv preprint arXiv:2406.16535**, 2024.

[10] Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In **International Conference on Artificial Intelligence and Statistics**, pp. 685–693. PMLR, 2024.

[11] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. **Advances in Neural Information Processing Systems**, Vol. 35, pp. 18878–18891, 2022.

[12] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. **Advances in Neural Information Processing Systems**, Vol. 35, pp. 30583–30598, 2022.

[13] Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Explaining emergent in-context learning as kernel regression. **arXiv preprint arXiv:2305.12766**, 2023.