JAIST Repository
https://dspace.jaist.ac.jp/

Title	文脈内学習のメカニズム的基盤
Author(s)	趙, 羽風
Citation	
Issue Date	2026-03
Type	Thesis or Dissertation
Text version	ETD
URL	https://hdl.handle.net/10119/20578
Rights	
Description	Supervisor: 井之上 直也, 先端科学技術研究科　, 博士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

The Mechanistic Basis of In-context Learning

Yufeng Zhao

Supervisor Naoya Inoue

Division of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

March 2026

Dissertation Committee

Associate Professor	Naoya Inoue
Professor	Kiyoaki Shirai
Professor	Le-Minh Nguyen
Associate Professor	Natthawut Kertkeidkachorn
Associate Professor	Benjamin Heinzerling (RIKEN / Tohoku Univ.)

Abstract

In-**c**ontext **L**earning (ICL) has recently emerged as a powerful paradigm enabling **L**anguage **M**odels (LMs) to perform few-shot learning without parameter updates. However, despite the success in application, the underlying mechanism of ICL, i.e., why and how LMs perform ICL operation, remains poorly understood. Therefore, this dissertation, which consists of 4 major works, aims to present a systematic investigation into the inner dynamics of LMs under the ICL scenario, and also utilize the gained insights to develop practical applications to improve ICL performance.

(Work 1) Motivated by the gap between the current interpretation of ICL and the observed ICL inference phenomena, first, we establish a comprehensive inference circuit based on the previous works of induction circuits, revealing how the induction circuits work in ICL on real data samples and real LMs. Through careful measurements and causal analysis, we identify three essential operations: (1) Input Text Encode: LMs encode every input text (in the demonstrations and queries) into a linear representation in the hidden states with sufficient information to solve ICL tasks. (2) Semantics Merge: LMs merge the encoded representations of demonstrations with their corresponding label tokens to produce joint representations of labels and demonstrations. (3) Feature Retrieval and Copy: LMs search the joint representations of demonstrations similar to the query representation on a task subspace, and copy the searched representations into the query. Then, language model heads capture these copied label representations to a certain extent and decode them into predicted labels. This decomposition unifies previously fragmented observations of ICL and is empirically validated via ablation, which shows substantial degradation when any step is disabled. Additionally, we confirm and list some bypass mechanisms that solve ICL tasks in parallel with the proposed circuit.

(Work 2) However, operation 3 above requires the ground-truth label in the demonstrations as the source for copying; otherwise, according to our proposed theoretical framework, LMs should fail to make correct predictions in the unseen-label scenario, where the ground-truth labels are absent from the demonstrations. Surprisingly, we find that LMs can still achieve significantly higher accuracy even under this unseen-label scenario, which suggests the existence of another significant mechanism in the aforementioned bypasses. Therefore, we hypothesize another mechanism, called *Task-oriented Information Removal*, which runs parallel with induction circuits. Specifically, we demonstrate that in the zero-shot scenario, LMs encode queries into non-selective representations in hidden states containing information for all possible tasks, leading to arbitrary outputs without focusing on the intended task, resulting in low accuracy. Meanwhile, we find that selectively removing specific information from hidden states by a low-rank filter effectively steers LMs toward the intended task. Building on these findings, by measuring the hidden states on carefully designed metrics, we observe that few-shot ICL effectively simulates such task-oriented information removal processes, selectively removing the redundant information from entangled non-selective representations, and improving the output based on the demonstrations, which constitutes a key mechanism underlying ICL. Moreover, we identify essential attention heads inducing the removal operation, termed Denoising Heads, which enables the ablation experiments blocking the information removal operation from the inference, where the ICL accuracy significantly degrades, especially when the correct label is absent from the few-shot demonstrations, confirming both the critical role of the information removal mechanism and denoising heads. Notice that such an information removal mechanism does not require the ground-truth labels in the demonstrations, thus explaining the ICL performance in the unseen-label scenario.

(Work 3) Then, we leverage the mechanistic insights gained from the above studies to develop practical applications to enhance ICL performance. Based on the conclusion of "LMs encode every input text into a linear representation in the hidden states with sufficient information to solve ICL tasks", we develop a new output calibration method called *Hidden Calibration*, which renounces token probabilities and uses the nearest centroid classifier on the LM's last hidden states. In detail, we assign the label of the nearest centroid previously estimated from a calibration set to the test sample as the predicted label. Our experiments on 6 models and 10 classification datasets indicate that Hidden Calibration consistently outperforms current token-based baselines by about 20%$\sim$50%, achieving a strong state-of-the-art in ICL. Our further analysis demonstrates that Hidden Calibration finds better classification criteria with less inter-class overlap, and LMs provide linearly separable intra-class clusters with the help of demonstrations, which supports Hidden Calibration and gives new insights into the principle of ICL.

(Work 4) Moreover, based on the conclusion of "induction head copy label information to the output", we develop an ICL-oriented fine-tuning method called *Attention Behavior Fine-Tuning*, building training objectives on the attention scores instead of the final outputs, to force the attention scores of induction heads to focus on the correct label tokens presented in the context and mitigate attention scores from the wrong label tokens. Our experiments on 9 modern LMs and 8 datasets empirically find that ABFT outperforms in performance, robustness, unbiasedness, and efficiency, with only around 0.01% data cost compared to the previous methods. Moreover, our subsequent analysis finds that the end-to-end training objective contains the ABFT objective, suggesting the implicit bias of ICL-style data to the emergence of induction heads. Also, ABFT demonstrates the possibility of controlling specific module sequences within LMs to improve their behavior, opening up the future application of mechanistic interpretability.

In summary, this dissertation improves the mechanistic interpretation of ICL by fitting induction circuits onto the previous observations, then proposing a new mechanism to supply the unseen label scenario. Moreover, based on such a mechanistic understanding, we propose some applicable methods to improve the performance of ICL, and also open up a new research direction, Mechanistic Controllability by the proposed prototypes.

Keywords: Mechanistic Interpretability, In-context Learning, Language Models, Transformer, Low-resource Model Controllability, Representation Learning

Acknowledgment

First and foremost, I owe my sincere and profound thanks to my supervisor **Professor Naoya Inoue**, who kindly accepted me as his doctoral student in 2023 and has since offered continuous encouragement, insightful advice, unwavering patience, and financial support. Without his mentorship, this dissertation could not have come to fruition. I am sincerely grateful to **Professor Kentaro Inui** (MBZUAI / Tohoku University / RIKEN) for his generous support and for granting me a postdoctoral position at RIKEN, and to **Professor Jianbing Shi** (Beijing Institute of Technology), who supervised my undergraduate thesis and first taught me the rigor and devotion required in scientific research. I also thank my high school teacher, **Mrs. Lyu**, for her inspiring and elegant teaching in mathematics and for encouraging independent thought.

This dissertation has benefited from the criticisms, suggestions, and collaboration of my excellent colleagues, **Mr. Haolin Yang** (University of Chicago), **Mr. Gouki Minegishi** (University of Tokyo), and **Mr. Peng Luo** (Beijing Institute of Technology). I am sincerely grateful for their intellectual companionship, which has been both inspiring and beneficial throughout my research career.

This work has been supported by the Nakajima Foundation and the JST FOREST Program (Grant Number JPMJFR232K, Japan). Parts of the research presented in this dissertation have also benefited from the thoughtful feedback of reviewers and organizers of international conferences, including ICLR 2025, ICLR 2026, and NAACL 2025.

Knowledge knows no bounds, and we are ceaseless explorers in the journey of life. Almost unconsciously, we discover, record, and narrate under the banner of curiosity; this natural process of discovery becomes not only an essential part of human existence but also a noble mission entrusted to us as beings endowed with spirit. In this sense, the course of human life is, in essence, a journey of knowledge: we chronicle the discoveries of the past and, in doing so, glimpse our aspirations for tomorrow. Thus, thousands of wisdom slumber with us at dusk and flow with our dreams toward the dawn. In this way, the river of knowledge is perpetuated, sustained by the vision of all those who weave the shape of tomorrow and carve the path to the future. Accordingly, I must first express the highest respect to the pioneers who have fearlessly walked the path toward truth. It is their burning curiosity, keen intellects, solid work, and rigorous scrutiny that have created the world we inhabit today. They have drafted blueprints for new worlds, and so their stories are celebrated as the mythology of materialism. Whenever I recall them, I feel a sense of kinship and am often moved to tears. In the same spirit, I also fervently hope that my own knowledge, discoveries, and reflections may one day inspire others, becoming solid foundations and nourishment for those who seek to uncover a broader tomorrow. To that end, and to take a step forward on the path toward answers, I am willing to devote my life to becoming a scholar worthy of the name.

I would like to thank Mr. Maurice Ravel, whose music I truly love from the bottom of my heart, and which has also enriched the process of writing this dissertation. Thanks to the Aniplex.exe Group: One of their outstanding works, *ATRI -My Dear Moments-*, was the original inspiration for my deep learning research, while another masterpiece, *Seedsow Lullaby*, gave me strength when I was exhausted.

Special thanks to my beloved maternal grandfather, **Mr. Yan**, who has kindly provided me with most of the mental care essential to the growth of any child. I believe he will be proud of my doctoral degree. I wish him good health, longevity, and happiness.

Finally, thank you for reading this dissertation. The author's proficiency is limited, and any comments, guidance, or corrections for possible errors or weaknesses are welcome.

Yufeng Zhao (Hakaze Cho)
March 2026, Japan

Table of Contents

Abstract III

Acknowledgment V

Table of Contents VI

List of Figures X

List of Tables XVII

I Introduction 1

1 Introduction 2
1.1 The Structure and Logic Flow of This Dissertation . . . 2

2 Preliminary & Previous Works 4
2.1 Formalization of In-context Learning . . . 4
2.2 Previous Works . . . 4
2.2.1 Previous Works on Improving In-context Learning Performance . . . 4
2.2.2 Previous Works on the Inner Mechanism of In-context Learning . . . 7
2.3 Preliminary Knowledge . . . 10
2.3.1 The Feed-forward Calculation of Transformer . . . 10
2.3.2 Mechanistic Interpretability . . . 12

II Theory 13

3 Mechanistic Basis for In-context Learning 14
3.1 Introduction . . . 14
3.2 Preparation . . . 15
3.2.1 Background & Related Works . . . 15
3.2.2 Experiment Settings . . . 16
3.3 Step 1, Input Text Encode: Semantics Encoding as Linear Representations in Hidden States 16
3.3.1 LLMs Encode Input Text on Forerunner Tokens in Hidden States . . . 16
3.3.2 Input Text Encoding is Linear and Task-Relevant but Position-biased . . . 17
3.4 Induction Circuits in Large Language Models . . . 18
3.4.1 Step 2, Forerunner Token Head: Copy from Text Feature to Label Token . . . 18
3.4.2 Step 3, Induction Head: Feature Retrieval on Task Subspace . . . 19
3.5 Putting Things Together . . . 21
3.5.1 Ablation Analysis . . . 21
3.5.2 Bypass Mechanism . . . 21
3.5.3 Explanation towards Observed ICL Phenomena . . . 22
3.6 Conclusion and Discussion . . . 22

4 Supplementary Mechanism Basis for ICL in Unseen Label Scenario 24
4.1 Introduction . . . 24
4.2 Background . . . 26
4.3 Method: Tracing the Task-oriented Information Removal . . . 26

4.4 Task-oriented Information Removal in In-context Learning . . . 27
4.4.1 Experiment Settings . . . 27
4.4.2 Step 1: Finding the Explicit TVS in Zero-shot Hidden States . . . 27
4.4.3 Step 2: Tracing the Implicit Task-oriented Information Removal . . . 28
4.4.4 Step 3: Causality Test by Denoising Heads . . . 30
4.5 Conclusion and Discussion . . . 32

III Application 34

5 Hidden Calibration for Improving the Decision Boundary of ICL 35
5.1 Introduction . . . 35
5.2 Background . . . 37
5.2.1 In-context Learning . . . 37
5.2.2 Token-probability Calibration for ICL . . . 37
5.3 Methodology . . . 37
5.3.1 Token Probabilities Are Not Good Classification Criteria . . . 37
5.3.2 Hidden Calibration . . . 38
5.4 Experiments & Main Results . . . 39
5.4.1 Experimental Settings . . . 39
5.4.2 Main Results: Hidden Calibration is A New State-of-the-art of ICL . . . 40
5.4.3 Efficiency: Low Complexity towards Time, Space, Data, and Prompting . . . 40
5.5 Analysis . . . 41
5.5.1 Effectiveness: Hidden Calibration Finds Criteria with Lower Overlap . . . 42
5.5.2 Principle: The Inner Linear-separability . . . 42
5.6 Related Works . . . 43
5.7 Discussion . . . 44
5.8 Limitations . . . 44

6 Low-resource Attention Behavior Fine-tuning for ICL 46
6.1 Introduction . . . 46
6.2 Background & Related Works . . . 47
6.3 Attention Behavior Fine-tuning . . . 48
6.4 Main Experiments . . . 49
6.4.1 Experiment Settings . . . 49
6.4.2 Main Results . . . 50
6.4.3 Comparison against End-to-end Fine-tuning . . . 51
6.5 Analysis . . . 51
6.5.1 Attention Visualization after ABFT . . . 51
6.5.2 Ablation Analysis . . . 52
6.5.3 Balance the Loss Factor A and B . . . 52
6.6 Consistency of Training Objective: ABFT and End-to-end Fine-tuning . . . 53
6.7 Discussion . . . 53
6.8 Limitations . . . 54

IV Discussion 55

7 Discussion 56
7.1 Summarization . . . 56
7.2 Still Open Questions and Future Directions . . . 56

References 58

Appendices

A Supplementary Materials for Chapter 3 **71**
A.1 Experiment Details and Settings . . . 71
A.1.1 Detailed Overall Experimental Settings . . . 71
A.1.2 Calculation of Mutual Nearest-neighbor Kernel Alignment . . . 71
A.1.3 Training and Inference of Centroid Classifier . . . 73
A.1.4 Cartography Details of Fig. 3.6 (Middle) . . . 73
A.1.5 Cartography Details of Fig. 3.7 . . . 74
A.1.6 Cartography Details of Fig. 3.8 . . . 74
A.1.7 Experiment Setting of Ablation Experiments in Table 3.1 . . . 75
A.2 LM Pre-training Dynamics Measured by ICL Circuit . . . 75
A.2.1 Data Distribution Requirement Explained by Hidden State Multiplex . . . 75
A.3 Can LMs Segment ICL Inputs? . . . 77
A.4 Semantics Merge is Non-selective on Location . . . 77
A.5 Measurement on Direct Decoding . . . 79
A.6 Demonstrations Enhance the Inference of Perplexed Queries . . . 80
A.7 Degradation Analysis: In-weight Learning without Ground-truth Label in Context . . . 80
A.7.1 How Accurate can Induction Heads Explain IWL Inference? . . . 80
A.7.2 Do Induction Heads Contribute to IWL Processing? . . . 81
A.7.3 The Counterexample . . . 82
A.7.4 Discussions: The Possibility of Different Inference Dynamics in IWL Scenarios . . . 83
A.8 Augmented Experiment Results . . . 84
A.8.1 Attention Head Statistics . . . 84
A.8.2 Other LMs' Experiment Results . . . 84
A.8.3 More Results of Fig. 3.7 . . . 85

B Supplementary Materials for Chapter 4 **96**
B.1 Experiment Details . . . 96
B.1.1 Prompt Template . . . 96
B.1.2 Training Details of $W_{\text{enc}}W_{\text{dec}}$ (§4.4.2) . . . 97
B.1.3 Details for Eccentricity / Covariance Flux and Experiments of §4.4.3 . . . 97
B.1.4 Details for Attention Visualization in Fig. 4.10 . . . 97
B.2 The Information Removing Magnitude of Low-rank Filter . . . 97
B.3 Visualization: the Direction of Information Reduction . . . 98
B.4 Case Analysis for DHs' Mechanism . . . 100
B.5 Discussion on the Selection of Ablation Threshold . . . 100
B.6 "Implicit Steering Heads" Discovered in §4.4.4 . . . 100
B.7 Grounding the Metrics to Information Measurement . . . 101
B.8 Prototype of Information Removal on Generative Scenario . . . 103
B.9 Supplementary Materials Outline . . . 104

C Supplementary Materials for Chapter 5 **169**
C.1 Experimental Details . . . 169
C.1.1 Datasets . . . 169
C.1.2 Baselines . . . 169
C.1.3 Prompts . . . 170
C.1.4 Details of Visualization in §5.3.1 . . . 170
C.1.5 Details of Experiment in §5.5.1 . . . 171
C.1.6 Details of Experiment in §5.5.2 . . . 173
C.1.7 Experimental Details for Fig. 5.6 . . . 173
C.2 Detailed Results . . . 174
C.2.1 Details of Main Results . . . 174
C.2.2 Details of Averaged Overlaps Results . . . 174
C.3 Additional Discussion . . . 174
C.3.1 The Similarity Measures Used in Hidden Calibration . . . 174
C.3.2 Transferability of the Centroid . . . 174
C.3.3 A Demonstration towards ICL Principles . . . 175
C.3.4 Applicability on More-way Classification and Biased Dataset . . . 175

C.4 Statements . . . 175
C.4.1 Author Contributions Statement . . . 175
C.4.2 License for Artifacts . . . 176
C.4.3 AI Agent Usage . . . 176

D Supplementary Materials for Chapter 6 **183**
D.1 Detailed Experiment Implementation . . . 183
D.1.1 Model & Dataset Details . . . 183
D.1.2 Prompt Templates . . . 183
D.1.3 Details of PID Algorithm . . . 183
D.1.4 Experiment Protocol of Unseen Label . . . 184
D.1.5 Experiment Protocol of Stability against Prompting . . . 184
D.2 Parameter Shift after ABFT against Layers . . . 184
D.3 Augmentation Results . . . 185
D.3.1 Augmentation Results for Data Efficiency (Fig. 6.2) . . . 185
D.3.2 Augmentation Results for Number of Induction Heads against Training Processing (Fig. 6.5) . . . 185
D.3.3 Augmentation Results for Attention Visualization (Fig. 6.4 and 6.8) . . . 185
D.4 Statements . . . 186

Authorship, Contribution, and AI Usage Statement **200**

Data Release & Reproducibility Statement **201**

Publications During Ph.D. Program **202**

List of Figures

1.1 The logical structure of this dissertation. . . . 3

2.1 Diagram of In-context Learning with $k = 3$. A few-shot demonstration sequence and one query are concatenated to form an ICL input, and language models predict the next token as the label of the query. . . . 4

2.2 Taxonomy of previous works for in-context learning. The author's principal contributions to ICL area are on **bold and green**. . . . 5

2.3 Schematic diagram of the Transformer inference process. . . . 10

3.1 The 3-phase inference diagram of ICL. **Step 1**: LMs encode every input text into representations, **Step 2**: LMs merge the encoded text representations of demonstrations with their corresponding label semantics, **Step 3**: LMs retrieve merged label-text representations similar to the encoded query, and copy the retrieved representations into the query representation. . . . 15

3.3 Input text encoding magnitudes (metricized by kernel alignment with feature encoded by an encoder-structured model) of hidden states in various layers in ICL scenario (The controlled experiments are results between current 6 datasets and TEE (Mohammad et al., 2018)). **Left**: Encoding magnitudes on hidden states from various types of token. **Middle**[1]: Encoding magnitudes with different k on the forerunner tokens. **Right**: Encoding magnitudes in layer 24 of Llama 3-70B against the causal language modeling loss of the input text with (upper) $k = 0$ and (lower) $k = 8$. . . . 17

3.2 Test results of centroid classifier trained on ICL hidden states. **Solid**: Centroid classification accuracy, **Dotted**: Kernel alignment. . . . 17

3.4 The similarities of ICL hidden states in different positions on layer 24 between **Left**: the same queries, **Right**: two different queries (on SST-2). . . . 18

3.5 Hidden states copy magnitude from forerunner tokens to label tokens against layers. **Left**: Kernel alignment between the forerunner token (the copy source) and the abstract label token of the next layer (the copy target). **Middle**: **Curves**: The count of marked forerunner token heads with correct and wrong labels; **Colored Areas**: The maximum attention scores from forerunner token to query (*copy magnitude*) with correct and wrong labels (detailed attention head statistical data is in Appendix A.8.1). **Right**: Centroid classifier results predicted on the hidden states of correct and wrong *label tokens*, on SST-2 and MR. **Solid**: Predicted by classifiers $\mathcal{C}_s$ trained on hidden states of *forerunner tokens*. **Dotted**: Predicted by classifiers $\mathcal{C}_y$ trained on hidden states of *label tokens*. . . . 19

3.6 Measurements for induction heads. **Left**: The count of marked induction heads and correct induction heads against layers. **Middle**: The correctness of attention assignment (the sum of attention scores from query's forerunner token towards correct label tokens normalized by attention scores towards all the label tokens) extracted from the following attention calculations: **Vanilla Attention:** attention scores directly calculated on full dimensionality. **Head Average:** the averaged attention scores among all the heads. **Best Ind. Head:** the scores of attention head with the most correctness. **Right**: The correct induction heads overlap of all dataset pairs. 20

3.7 Label representations of k demonstrations visualized on **Left 4**: correct and **Right 4**: wrong induction head, on one sample of SST-2 (see Appendix A.8.3). ○: "positive" label, ×: "negative" label, ▲: zero vector. Color: Radial density of each categories, to estimate the attention assigned to query, negative to positive (cartography: Appendix A.1.5). . . . 20

3.8 Dynamics and deserialization of magnitudes of proposed 3 inference steps (cartography details: Appendix A.1.6). . . . 21

4.1 **(A)** A zero-shot query is encoded into a non-selective semantic representation containing all possible label information on various subspaces, making the output arbitrary among these labels. **(B)** Demonstrations help LM filter the label information, saving only the task-related one on the specific subspace (termed **T**ask-**V**erbalization **S**ubspace (TVS)), leading to the task-specific output. **(C)** We explicitly find the TVS by injecting a low-rank filter into the residual stream of zero-shot inputs, and train only the filter to drive the final outputs towards the ground-truth labels. . . . 25

4.2 Logits distribution (selected) of zero-shot input "Donald Trump, label:". . . . 26

4.3 Filter-injection evaluations on various layers and ranks. **Marker size**: the remaining covariance out of the top-r principal components of the point cloud of h_q^l. Refer to Appendix B.9 for numerical results. **Encoding Magnitude**: the quality of the hidden states in the current layer as a sentence encoding of the query, emerging simultaneously with the test accuracy, calculated follows Cho et al. (2025b). . . . 28

4.4 Test results of **(left)** open-end accuracy against demonstration numbers k, **(middle)** eccentricity of hidden states point cloud against demonstration numbers k on various layers, **(right)** covariance flux through the catched $W_{\text{enc}}W_{\text{dec}}$ against demonstration numbers k on various layers. . . . 29

4.5 Eccentricity and covariance flux w.r.t. layer on MR, with instruction texts, demonstrations, or plain zero-shot inference. . . . 29

4.6 Eccentricity and covariance flux w.r.t. layer on MR, with random label, unseen label, and normal settings. . . . 29

4.7 Ablation results for the two metrics and output accuracy of every head in Llama3.2-1B Layer 9 on 8-shot SST-2 (relative to the clean run). A smaller value indicates a more severe drop with ablation, i.e., this attention head contributes more significantly to the metric. . . . 30

4.8 Counting of DHs and anti-DHs w.r.t. layers. . . . 30

4.9 Head-wise (**scatter size**) induction magnitude and (**color**) covariance flux relative change after ablation. . . . 31

4.10 Attention map of a common DH (Layer 9, Head 15). White dotted lines are the position of label tokens. . . . 31

4.11 Overlap of DH over all dataset pairs. . . . 31

4.12 Filter-injection results on country-capitals task. In **cross-entropy loss**. . . . 32

4.13 Filter-injection results on the people-profession task. In **cross-entropy loss**. . . . 32

4.14 DH amplification results on Llama 3.2-1B and SST-2. . . . 33

5.1 In an ICL diagram, **A.** The prompt of ICL consists of a concatenation of demonstrations and a query. LMs encode the prompt into the last hidden state h, then **B.** (previous works) use the un-embedding vectors of the label tokens (E_+^U, E_-^U) to decode the h to prediction $\hat{y}$, then calibrations are used to adjust the predicted logits. **C.** Our work uses the calibration dataset to calculate centroids ($\bar{h}_+$, $\bar{h}_-$) to decode the h. . . . 36

5.2 Token probability-based decision boundaries (original & batch calibrated) are suboptimal comparing to centroid-based boundary. Points and contour lines are ICL's last hidden states and kernel densities mapped by Principal Component Analysis. Oblique coordinate axis is the direction of the un-embedding difference of label tokens $\left(E_+^U - E_-^U\right)$, where the kernel densities of mapped data points are plotted. The rotating calibration by $A \neq \mathbf{1}$ (e.g. Contextual Calibration, Domain Calibration) has a limited feasible mapping direction[17] . . . 38

5.3 The diagram of Hidden Calibration. **Step 1:** Calculating the hidden state centroid of each label. **Step 2:** Find the label of the nearest centroid of the text sample to be the prediction. . . 38

5.4 The classification performance (Macro F1(%)) of 6 models averaged on 10 datasets. Hidden Calibration (Hidd.C) is a new state-of-the-art of ICL, where demonstrations consistently improve the performance. . . . 39

5.5 Sensitivities on (**left**) prompt template, (**middle**) demonstration label distribution, and (**right**) demonstration order on Llama 2-6.9B and `Rotten_Tomatoes`. Legend is consistent with Fig. 5.4, omitted. . . . 40

5.7 Diagrammatic sketch of the overlap calculation with GPT 2-XL on `SemEval 2014-Task 4 Restaurant`, $k = 4$. **Curves:** The kernel density of probability difference of l_1 ="positive" and l_2 ="negative". **Heatmaps:** The overlap of 2-combinations (we plot the combination with the same label with overlap 1, but omit them in averaging). . . . 41

5.6 Classification performance against the number of training examples (m) of calibrations. As a comparison, we plotted the results of vanilla ICL using equivalent demonstrations, due to the quadratic overhead against the context length, we can test up to $k = 4|\mathcal{Y}|$. 41
5.8 The Averaged Overlap of 4 inference methods on GPT 2-XL and 5 datasets. 42
5.9 The Averaged Overlap on OPT-2.7B and 5 datasets against the demonstrations number. 42
5.10 Hidden state clustering w.r.t. k of Fig. 5.12 visualized on the direction of principal component II. 42
5.12 ICL hidden states clustering dynamics visualized on OPT-2.7B and `SemEval 2014-Task 4 Laptops` (Pontiki et al., 2014). The densities of data points appear in clusters responding to their query labels originally when no demonstrations are given, and gradually converge to the centroid w.r.t. the demonstrations number (k). 43
5.11 The averaged intra-class standard error of data points and the inter-class averaged centroid distance against k. **Solid curves**: means on 5 datasets; **Dashed and pale curves**: Individual results for each dataset. 43

6.1 Diagram of ABFT framework. (A) **An example of ICL-style inputs.** We build datasets from such examples to fine-tune models. (B) **Feed-forward inference of ICL.** We collect the attention scores of every attention head in every layer to calculate the training objective. and we only enable the gradient of the W_Q and W_K matrices. (C) **The criterion for induction head.** Only attention heads producing attention scores with a significant focus on the label tokens can be identified as induction heads. (D) **Loss calculation of ABFT.** Only induction heads return a non-zero loss, and such loss contains a punishment on "wrong" attention scores to wrong label tokens, and a reward on "correct" attention scores to correct label tokens. . . . 47
6.2 Data efficiency: accuracy against data size on DeepSeek-R1 Distill Qwen-14B / 8 datasets. . . 51
6.3 Accuracy on unseen label inputs and randomly sampled inputs, w/ and w/o ABFT. 51
6.4 Induction attention scores and correct (on the correct labels) induction attention score averaged among heads on each layer. **ABFT enhances induction attention, and improves the correctness**. 51
6.5 Number of attention heads that are judged as induction heads on 4 settings, against the training processing (the number of seen data samples). **PID effectively stabilizes the induction head numbers**. 51
6.6 Accuracies with various settings on hyperparameter A and B, with and without PID algorithm. **PID weakens the sensitivity to initial parameters**. 52
6.7 **Upper**: A better accuracy (lower loss) in the interpolation path suggests the same basin, and vice versa. **Lower**: Contour map of accuracies against the coefficient α_{E} and α_{A} in Eq. 6.2. **E2E fine-tuned and ABFT models are located in the same low-loss area.** 53
6.8 Attention score visualization of the pre-trained model, ABFT model, and E2E fine-tuned model, on the same input as Fig. D.7, and the same models as Fig. 6.7 (Refer to Appendix D.3.3 for details and more cases). 53
6.9 Contour map of accuracies against the α_{E} and α_{A} on GPT 2-L, and the θ_{E} is set as MetaICL model. 54

A.1 Distributions (*clusters*) of representations generated by: **Left**: encoder model (BGE), clustering w.r.t. the label. **Middle**: ICL on the forerunner token, where representations gather into one point ($e([\text{": "}])$) when no Transformer operation is conducted. **Right**: ICL on the label token, where representations gather into points w.r.t. label (using a 2-way example) when no Transformer operation is conducted, causing a high background value. 72
A.2 Visualization of non-trivial attention connection defined and disconnected in ablation analysis in §3.5.1 and Table 3.1. Notations and line numbers are same as Table 3.1. 74
A.4 The 3 operating magnitudes on Pythia-6.9B with various pre-training steps. **Left**: Step 1, Input Text Encode. **Middle**: Step 2, Semantics Merge. **Right**: Step 3, Feature Retrieval and Copy, measured by the correct induction head numbers. 75
A.3 Operating magnitude for each inference step (normalized) and ICL accuracy against pre-training steps on Pythia-6.9B and SST-2. 75
A.5 Attention score visualized with the forerunner token of the ICL query as the attention query, from (**Left**) layer 21 and (**Right**) layer 2 in Llama 3-70B from an input case. 76
A.6 Attention matrix visualized on layer 2, head 53 in Llama 3-70B from an input case. 77

A.7 Copy magnitude normalized by the sequence length from the previous token to the non-label token and label token, for every token and head in each layer. Significant statistical differences cannot be observed. . . . 78
A.9 Predicting distributions on different forerunner tokens with various direct decoding logits. Inference process used: **Upper**: Vanilla ICL, **Lower**: Biased removed inference by Contextual Calibration proposed by Zhao et al. (2021). . . . 79
A.8 Direct decoding accuracies on various layers. . . . 79
A.10 The correlations between language modeling loss and ICL prediction accuracies. **Upper**: 0-shot results; **Lower**: 4-shot results. . . . 80
A.11 The JS divergence between the predicted output by the induction attention score (Eq. A.15) and real output on ICL and IWL settings. Since not all attention heads are induction heads, the lowest 5 data of each layer is selected. . . . 81
A.12 A counterexample when the induction head's behavior cannot predict the LM's inference behavior. Given the demonstration shown in the figure, the attention scores from the label's forerunner token are visualized on layer 31 of Llama 3-70B. The **left** part is a standard ICL scenario where the ground-truth label of the query can be accessed in the demonstrations. The **right** part is the IWL scenario where the ground-truth label of the query is not presented in the demonstrations. A clear induction pattern can not be observed in the IWL scenario. . . . 83
A.13 Supplemental experiment result on more samples for Fig. 3.7. . . . 84
A.14 Forerunner Token Head marked on Llama 3-70B. . . . 85
A.15 Forerunner Token Head marked on Llama 3-8B. . . . 86
A.16 Forerunner Token Head marked on Falcon-40B. . . . 88
A.17 Forerunner Token Head marked on Falcon-7B. . . . 89
A.18 Correct Induction Head marked on Llama 3-70B. . . . 90
A.19 Correct Induction Head marked on Llama 3-8B. . . . 90
A.20 Correct Induction Head marked on Falcon-40B. . . . 91
A.21 Correct Induction Head marked on Falcon-7B. . . . 92
A.22 Augmentated results towards Fig. 3.3 on Llama 3-8B. . . . 92
A.23 Augmentated results towards Fig. 3.3 on Falcon-40B. . . . 93
A.24 Augmentated results towards Fig. 3.3 on Falcon-7B. . . . 93
A.25 Augmentated results towards Fig. 3.2 and 3.4 (on Layer 16) on Llama 3-8B. . . . 93
A.26 Augmentated results towards Fig. 3.2 and 3.4 (on Layer 24) on Falcon-40B. . . . 93
A.27 Augmentated results towards Fig. 3.2 and 3.4 (on Layer 16) on Falcon-7B. . . . 94
A.28 Augmentated results towards Fig. 3.5 on Llama 3-8B. . . . 94
A.29 Augmentated results towards Fig. 3.5 on Falcon-40B. . . . 94
A.30 Augmentated results towards Fig. 3.5 on Falcon-7B. . . . 94
A.31 Augmentated results towards Fig. 3.6 on Llama 3-8B. . . . 95
A.32 Augmentated results towards Fig. 3.6 on Falcon-40B. . . . 95
A.33 Augmentated results towards Fig. 3.6 on Falcon-7B. . . . 95

B.1 Examples of seen label demonstration and unseen label demonstration. . . . 96
B.2 The input for attention visualization. . . . 98
B.3 An illustration of task-oriented information removal with nonmonotonic eccentricities against the number of demonstrations. The ellipsoids refer to the hidden state point cloud with two ground-truth labels (distinguished in colors), which is separated even no demonstrations are given, according to previous works (Cho et al., 2025a; Yang et al., 2025c). . . . 98
B.4 Visualization of the last token's hidden states from Layer 13 of Llama 3.2-1B, (**upper**) on SST-2 with a nonmonotonic eccentricity, and (**lower**) on MR with a monotonic eccentricity. The colors refer to the queries' ground-truth label. . . . 99
B.5 (Upper) attention map; (lower) attention scores to the last token (":", as the attention query) of Layer 6 Head 11. . . . 99
B.6 Examples for test samples from (upper) SST-2 and (lower) MR. . . . 100
B.7 DH / RH ablation results against ablation threshold. . . . 100
B.8 Attention score visualization of DH (Llama 3.2-1B Layer 9 Head 15), where the last token serves the attention query. . . . 101
B.9 Eccentricity results of 3 non-classification tasks. **(Left)** people-profession; **(Middle)** opus-100 translation: English to Chinese; **(Right)** country-capital. . . . 103
B.10 Diagram for eccentricity and covariance flux metrics. The green plain refers to the TVS ($W_{\text{enc}}W_{\text{dec}}$) obtained in §4.4.2. . . . 103

B.11 Filter-injection evaluations on opus-100 English–Chinese translation task. **In cross-entropy loss.** . . . 104
B.12 The cosine similarity between all the eigenvectors (each for a scatter) of W_{enc} on specific layers and the subspace spanned by the top-64 principal components of the hidden state point cloud, measured by the ratio of vector norms before and after mapping to the principal subspace. On Llama 3.2-1B, SST-2, repeated on various inner ranks of TVS, i.e., the column dimensionality of W_{enc}. . . . 106
B.13 The similar result with Fig. B.12 on Llama 3.2-1B, MR. . . . 106
B.14 The similar result with Fig. B.12 on Llama 3.2-1B, FP. . . . 106
B.15 The similar result with Fig. B.12 on Llama 3.2-1B, SST-5. . . . 106
B.16 The similar result with Fig. B.12 on Llama 3.2-1B, AGNews. . . . 106
B.17 The similar result with Fig. B.12 on Llama 3.2-1B, Subjective. . . . 106
B.18 Augmentation results for Fig. 4.3 on Llama 3.2-1B. . . . 107
B.19 Augmentation results for Fig. 4.3 on Llama 3-8B. . . . 107
B.20 Augmentation results for Fig. 4.3 on Llama 3-13B. . . . 108
B.21 Augmentation results for Fig. 4.3 on Qwen 2.5-3B. . . . 108
B.22 Augmentation results for Fig. 4.3 on Qwen 2.5-3B Instruct. . . . 109
B.23 Augmentation results for Fig. 4.3 on Qwen 2.5-7B. . . . 109
B.24 Augmentation results for Fig. 4.12 on 3 models. . . . 110
B.25 Augmentation results for Fig. 4.13 on 3 models. . . . 110
B.26 Augmentation results for Table 4.1, (left) on Qwen 2.5-3B, (right) on Llama 3-8B. . . . 110
B.27 Augmentation results for Fig. 4.11, (left) on Qwen 2.5-3B, (right) on Llama 3-8B. . . . 110
B.28 Augmentation results for attention visualization on (left) Layer 9, Head 6, (right) Layer 9, Head 20, with the inputs shown in Fig. B.2. . . . 110
B.29 Numerical results of covariance out of rank on Llama 3.2-1B. . . . 111
B.30 Numerical results of covariance out of rank on Llama 3-8B2828[28]. . . . 111
B.31 Numerical results of covariance out of rank on Qwen 2.5-3B2828[28]. . . . 111
B.32 Numerical results of covariance out of rank on Qwen 2.5-3B Instruct2828[28]. . . . 111
B.33 Numerical results of covariance out of rank on Qwen 2.5-7B2828[28]. . . . 111
B.34 Augmentation results for Fig. 4.4 on Llama 3.2-1B. . . . 112
B.35 Augmentation results for Fig. 4.4 on Llama 3-8B. . . . 113
B.36 Augmentation results for Fig. 4.4 on Llama 3-13B. . . . 114
B.37 Augmentation results for Fig. 4.4 on Qwen 2.5-3B. . . . 115
B.38 Augmentation results for Fig. 4.4 on Qwen 2.5-3B Instruct. . . . 116
B.39 Augmentation results for Fig. 4.4 on Qwen 2.5-7B. . . . 117
B.40 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3.2-1B. . . . 118
B.41 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3-8B. . . . 119
B.42 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3-13B. . . . 120
B.43 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-3B. . . . 121
B.44 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-3B Instruct. . . . 122
B.45 (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-7B. . . . 123
B.46 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, SST-2. . . . 125
B.47 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, MR. . . . 127
B.48 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, FP. . . . 129
B.49 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, SST-5. . . . 131
B.50 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, AGNews. . . . 133
B.51 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, Subjective. . . . 135
B.52 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, SST-2. . . . 136
B.53 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, MR. . . . 137
B.54 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, FP. . . . 138

B.55 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, SST-5. . . . 139
B.56 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, AGNews. . . . 140
B.57 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, Subjective. . . . 141
B.58 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, SST-2. . . . 143
B.59 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, MR. . . . 145
B.60 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, FP. . . . 147
B.61 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, SST-5. . . . 149
B.62 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, AGNews. . . . 151
B.63 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, Subjective. . . . 153
B.64 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, SST-2. . . . 155
B.65 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, MR. . . . 157
B.66 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, FP. . . . 159
B.67 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, SST-5. . . . 161
B.68 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, AGNews. . . . 163
B.69 (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, Subjective. . . . 165
B.70 Augmentation results for Fig. 4.8 on Llama 3.2-1B. . . . 166
B.71 Augmentation results for Fig. 4.8 on Llama 3-8B. . . . 166
B.72 Augmentation results for Fig. 4.8 on Qwen 2.5-3B (the threshold is set to $\pm 2.5\%$). . . . 166
B.73 Augmentation results for Fig. 4.8 on Qwen 2.5-3B Instruct (the threshold is set to $\pm 2.5\%$). . . . 166
B.74 Augmentation results for Fig. 4.9 on Llama 3.2-1B. . . . 167
B.75 Augmentation results for Fig. 4.9 on Llama 3-8B. . . . 167
B.76 Augmentation results for Fig. 4.9 on Qwen 2.5-3B. . . . 168
B.77 Augmentation results for Fig. 4.9 on Qwen 2.5-3B Instruct. . . . 168

C.1 The classification performance (Accuracy(%)) of 3 models averaged on 10 datasets. . . . 172
C.2 The augmented results on 2 models of Fig. 5.8. . . . 173
C.3 The classification performance (Macro F1(%)) of Hidden Calibration with difference similarity measure. . . . 174

D.1 Visualization of induction attention scores on unseen label settings (Llama 3-8B, TREC). . . . 184
D.3 Augmentation results of Fig. 6.4 on GPT 2-L. . . . 185
D.4 Augmentation results of Fig. 6.4 on GPT 2-XL. . . . 185
D.5 Augmentation results of Fig. 6.4 on Falcon 3-7B. . . . 185
D.6 Augmentation results of Fig. 6.4 on Llama 3-42B. . . . 185
D.2 Accuracy against training set size as a metric of data efficiency, for Qwen 2.5-32B. . . . 185
D.7 Attention score visualization on the last token of ICL input of every attention head (vertical axis) towards each token. Label tokens and their contents are marked with dotted lines. Refer to Appendix D.3.3 for more examples and layers. **ABFT successfully focuses attention scores to correct labels**. . . . 186
D.8 Shifting distance before and after ABFT on the `c_attn` matrix of GPT 2-XL and SST-2. . . . 187
D.9 Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama 3-8B and SST-2. . . . 187
D.10 Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama 3-42B and SST-2. . . . 187

D.11 Induction head numbers along training dynamics on Llama 3-8B and all 8 datasets. 188
D.12 Induction head numbers along training dynamics on Falcon 3-7B and all 8 datasets. 189
D.13 Induction head numbers along training dynamics on GPT 2-Large and all 8 datasets. 190
D.14 Induction head numbers along training dynamics on GPT 2-XL and all 8 datasets. 191
D.15 Induction head numbers along training dynamics on DeepSeek-R1 and all 8 datasets. 192
D.16 Induction head numbers along training dynamics on Qwen 2.5-32B and all 8 datasets. 193
D.17 Induction head numbers along training dynamics on SimpleScaling s1.1 and all 8 datasets. . . 194
D.18 Induction head numbers along training dynamics on Llama 3-42B and all 8 datasets. 195
D.19 Induction head numbers along training dynamics on Llama 3-56B and all 8 datasets. 196
D.20 Augmentation results (4 cases) for attention visualization. **Upper**: results for Fig. D.7, **lower**: results for Fig. 6.8. 197
D.21 Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. D.7. 198
D.22 Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. 6.8. 199

List of Tables

3.1 Accuracy variation (%) with each inference step ablated on Llama 3-8B. Small numbers are **controlled results** (mean ± std) of randomly ablating equivalent amounts of connections. Ablations are applied from the bottom to the top layers, and results with various ablated layers are reported (detailed settings: Appendix A.1.7). . . . 21
3.2 Accuracy drop with shortcut ablated. . . . 22
3.3 Performance of full and layer-pruned ICL inference. . . . 23

4.1 Fine-tuning results with parts of the filter frozen, on Llama 3.2-1B and SST-2. . . . 28
4.2 Ablated inference accuracy (%) of DHs, averaged on all 6 datasets. **Layer**: the ratio of the layer scanned for DH / total layers. **w/o DH**: accuracy with DHs ablated. **w/o RH**: accuracy with ablation of randomly sampled attention heads of equal amounts to DHs on each layer. **P-value**: the probability for "w/o RH" to sample a value less than "w/o DH" (i.e., the likelihood of DHs being randomly sampled). (Due to computational constraints, we only evaluate Llama 3-13B on SST-2.) . . . 32

5.1 The **additional** (compare to vanilla ICL) time and space on calibration and inference cost of various methods. Hidden Calibration has a similar cost upper bound to other calibrations. $|\mathbb{V}|$ is the vocabulary size. . . . 41

6.1 Accuracies (%) of ABFT and baselines on 9 LMs and 8 datasets. The best results are in **bold**. 49
6.2 A comparison between ABFT and End-to-end Fine-tuning (E2E FT). Param.*: Parameters which are required in `FP16/32` and with gradient on. . . . 50
6.3 Prediction consistency metrics (%) on each models averaged among 8 datasets. . . . 50
6.4 Ablation analysis of removing some components from ABFT. Notice that the PID algorithm is to stabilize the induction head number by adjusting the factor A, so when we disable the head filter or fix the A or B, the PID algorithm naturally loses its function. . . . 52

A.1 Prompt templates used in this work. . . . 71
A.2 Accuracy drop with delimiters removed/modified from prompts on SST-2. . . . 77
A.3 Accuracy variation (%) with each inference step ablated on **Left**: Llama 3-8B, **Right**: Falcon-7B **in IWL scenarios**. Notations are the same with Table 3.1, significant comparisons are highlighted with arrows. . . . 82
A.4 Label probabilities from the model predictions of the ICL (query: "Geoffrey Hinton" (Researcher)) and IWL (query: "Michael Jordan" (Athlete)) scenario shown in Fig. A.12. In this case, IWL predictions also benefit from demonstrations. . . . 82
A.5 Results of Table 3.1 on Falcon-7B. . . . 84

B.1 Prompt templates used in this work. . . . 96
B.2 Instruction used in Fig. 4.5. . . . 96
B.3 Models and corresponding checkpoint names used in this work. . . . 96
B.4 Accuracies in all configurations shown in Fig. 4.5 and 4.6. **Ins.**: with basic instruction; **Ins. (LS)**: with instructions and label space prompt; **Unseen**: with unseen label settings; **Rand** y: with randomly sampled demonstrations label tokens in the original label space. . . . 97
B.5 Fine-grained data for Table 4.2. . . . 102

C.1 Prompt templates used in this work. . . . 169
C.2 Datasets and Abbreviations used in this work. . . . 170
C.3 The 7 prompt templates used in the experiment (on `Rotten_Tomatoes`) of Fig. 5.5 (left). . 171
C.4 Performance of all 6 models on TREC for a more-way classification, and on Hate_Speech18 for a biased dataset. $k = 4$, top-2 results are in **bold**. . . . 174

C.5 Transferability of centroid among various datasets with the same label space. Big numbers are the average improvement (MF1) compared to vanilla ICL, small numbers are standard error. Statistically significant results ($p < 0.1$) are in **bold**. 175
C.6 Transferability of centroid among various k on the same dataset. $k_1 \rightarrow k_2$ is to use centroids estimated by k_1 demonstrations for inference on test examples with k_2 demonstrations. Other annotations are the same as Table. C.5 175
C.7 Classification performance (Macro F1(%)) on GPT 2-XL. mean_{std}, top-2 results are in **bold**. . 177
C.8 Classification performance (Macro F1(%)) on OPT-2.7B. mean_{std}, top-2 results are in **bold**. . 178
C.9 Classification performance (Macro F1(%)) on Llama 2-6.9B. mean_{std}, top-2 results are in **bold**. 179
C.10 Classification performance (Macro F1(%)) on Llama 3-8B. mean_{std}, top-2 results are in **bold**. 180
C.11 Classification performance (Macro F1(%)) on Llama 2-13B. mean_{std}, top-2 results are in **bold**. 181
C.12 Classification performance (Macro F1(%)) on Llama 2-34B. mean_{std}, top-2 results are in **bold**. 182

D.2 `Huggingface` repository name for models used in this work. 183
D.1 Prompt templates used in this work. 184

Part I
Introduction

I Introduction **1**

1 Introduction **2**
1.1 The Structure and Logic Flow of This Dissertation 2

2 Preliminary & Previous Works **4**
2.1 Formalization of In-context Learning 4
2.2 Previous Works 4
2.2.1 Previous Works on Improving In-context Learning Performance 4
2.2.1.1 Pre-training & Model Tuning 4
2.2.1.2 Output Calibration and Logits Re-calculation 6
2.2.1.3 Prompt Engineering 6
2.2.1.4 Implicit ICL 7
2.2.2 Previous Works on the Inner Mechanism of In-context Learning 7
2.2.2.1 Mechanistic Interpretability 7
2.2.2.2 Attribution Analysis 9
2.2.2.3 Analogy to Simple Algorithms 9
2.2.2.4 Error Bounds and Numerical Simulation 9
2.3 Preliminary Knowledge 10
2.3.1 The Feed-forward Calculation of Transformer 10
2.3.2 Mechanistic Interpretability 12

Chapter 1

Introduction

In-**C**ontext **L**earning (ICL) (Radford et al., 2019; Dong et al., 2022) is an emerging few-shot learning paradigm: given the *demonstrations* $\{(x_i, y_i)\}_{i=1}^{k}$ consisting of [input text]-[label token] pairs and a *query* x_q, **L**anguage **M**odels (LMs) take the sequence $[x_1, y_1, \ldots, x_k, y_k, x_q]$ (shown as an example in Fig. 2.1 and detailed defination in §2.1) as input and then predicts the label for x_q by causal language modeling operation. ICL has aroused widespread interest, but its underlying mechanism is still unclear.

Some studies attempt to reconstruct or model the ICL process as human-understandable algorithms and associate them with actual model operations. This methodology is known as Mechanistic Interpretability (Rai et al., 2024). Classical examples include viewing ICL as a Bayesian inference (Xie et al., 2022), a kernel regression (Han et al., 2023a), an implicit gradient descent (Dai et al., 2023), or a two-step copy-retrieval process that is known as the induction circuit (Olsson et al., 2022), which will be detailed in Chapter 2.2.2. These works before this dissertation can interpret ICL to some extent. Other empirical works observe interesting behaviors of ICL. For example, compared with traditional gradient-based learning (Zhang et al., 2021), ICL is less sensitive to the correctness of label y (Min et al., 2022c). Moreover, a simpler demonstration can induce better accuracy (Gonen et al., 2023), and the model often becomes biased to repeat the last label or the most frequent label (Zhao et al., 2021). These observations are intriguing, but scattered, and lack a unified theoretical explanation.

Notice that, we mentioned that "these (interpretation) works before this dissertation can interpret ICL to some extent", the reason is that: the current simplified models fail to explain the empirical phenomena before. For instance, treating ICL as implicit gradient descent cannot explain why it remains stable under label noise. Also, treating it as a kernel regression cannot explain the existence of strong prediction biases, which is unsatisfactory.

Therefore, the initial motivation of this dissertation is to propose a modeling framework that can capture all the above-observed phenomena. In Chapter 3, based on the theory of induction circuits, we conduct extensive and carefully designed measurements on real large language models and extend this theory into a three-step process that explains all the noted inference behaviors. This represents a significant advancement in mechanistic interpretability within the context of ICL.

However, the above framework works well, except in an important scenario: when the ground truth label of x_q does not appear in the demonstrations, which we call the *unseen label scenario*, ICL is expected to fail. In reality, its performance drops substantially, as shown in Table A.3, yet it remains significantly above random chance. This indicates that another mechanism besides induction is also contributing to the ICL process. In Chapter 4, we identify this mechanism and demonstrate that it cooperates with the induction mechanism to support ICL accuracy.

Finally, based on the theoretical insights obtained, we propose two applications that improve ICL accuracy in Chapter 5 and Chapter 6.

1.1 The Structure and Logic Flow of This Dissertation

Motivation & Objective for This Dissertation. As mentioned before, current mechanistic interpretations for ICL can not catch all the observed phenomena. Therefore, the main motivation and objective of this dissertation is to provide a more comprehensive and unified mechanistic interpretation for ICL, which can well explain the scattered observations, driving the §3. The flaw of induction-head-based interpretation of unseen label scenarios found in §3 drives the §4, which further improves the mechanistic interpretation.

The Logic Flow of This Dissertation. As shown in Fig. 1.1, given the objectives, we propose two major theories in §3 and §4. To further validate our theories and demonstrate their empirical applicability, in §5 and §6, we present two applications based on the aforementioned theoretical framework. These applications

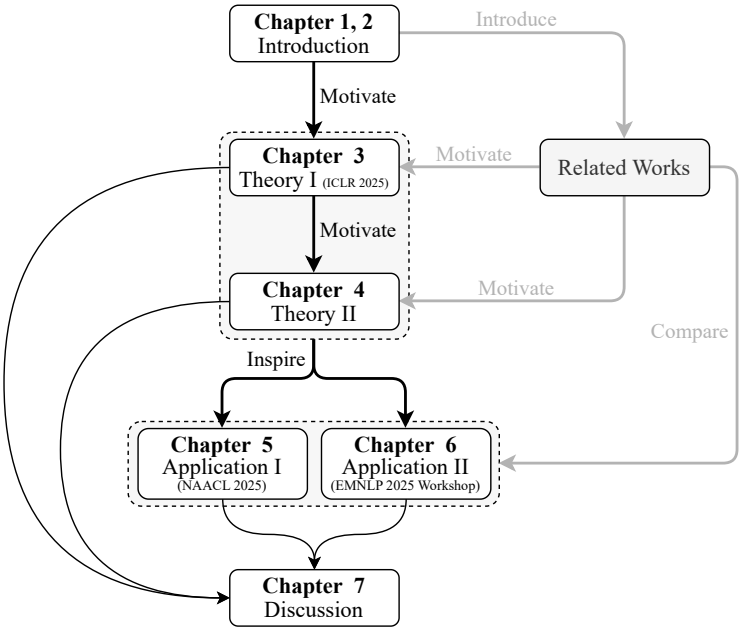


Figure 1.1: The logical structure of this dissertation.

are not only highly practical but also confirm the reliability of our theories.

Significance. The research presented in this doctoral dissertation is based on the author's papers accepted or submitted to top-tier AI conferences. Specifically, §3 is derived from ICLR 2025, §4 is derived from ICLR 2026, §5 from NAACL 2025, and §6 from the EMNLP 2025 BlackBox Workshop. As of October 2025, all chapters have been peer-reviewed and accepted. According to the statistics from (Eickhoff, 2023), the cumulative impact factor of these chapters reaches 114.

Appendices. As mentioned before, most of the content is a reiteration of (some of) the author's first-author top-tier conference papers. Therefore, to ensure a compact and smooth narrative, and align with the general conventions of top-tier conference papers, a substantial amount of content has been moved to the appendix. This does not imply that the appendix is less valuable or of lower quality: these contents are typically necessary details but may hinder the fluency of our main content. Examples include repeated experiments conducted on different datasets/models, or under different conditions, detailed explanations of experimental settings, raw data for descriptive statistics, and supporting evidence for secondary conclusions. Skipping these sections does not affect the **understanding** of the main content of the dissertation, while they serve as **essential support** for the dissertation, dry but make the argument here solid. Therefore, they are not merely redundant content added to "make the dissertation **appear** more lengthy".

Chapter 2

Preliminary & Previous Works

In this chapter, we will formalize ICL, and comprehensively review existing related research, especially the inner mechanism of ICL (§2.2.2), to motivate our subsequent investigation in detail. Then, we review some preliminary knowledge as essential basics and also agree on some terms.

2.1 Formalization of In-context Learning

In-**C**ontext **L**earning (ICL) (Radford et al., 2019; Dong et al., 2022) is an emerging few-shot learning paradigm: given a k-shot *demonstrations* $\{(x_i, y_i)\}_{i=1}^{k}$ consisting of [input text]-[label token] pairs and a *query* x_q, LMs take the sequence $S = [x_1, y_1, x_2, y_2, \ldots, x_k, y_k, x_q]$ as input and then predicts the label $\hat{y}_q$ for x_q by causal language modeling operation, as shown for a case in Fig. 2.1. To be formalized, a language model, parameterized by θ receives such input S, and output it onto $|\mathbb{V}|$-dimensional probability simplex as o, where the $|\mathbb{V}|$ is the vocabulary size of the LM (we review the calculation of LMs in §2.3.1):

$$o = \mathrm{LM}_\theta([x_1, y_1, x_2, y_2, \ldots, x_k, y_k, x_q]). \tag{2.1}$$

Restricted Decoding. A typical ICL processing only filters some specified tokens in a *label space* $\mathbb{Y}$ to judge the prediction. For example, with a label space of {"positive", "negative"}, output probabilities in o for other tokens will be discarded, with only $\mathrm{LM}_\theta(\text{"positive"}|S)$ and $\mathrm{LM}_\theta(\text{"negative"}|S)$ remained, then the label token with the most probability will be assigned as the final prediction. Such a decoding strategy is called "Restricted Decoding".

Due to the convenience (inference-only) and low-resource characteristics (only need few-shot examples), ICL has raised widespread research interest. Therefore, in this dissertation, we focus on ICL in **L**arge **L**anguage **M**odels (LLMs), and introduce a series of top-conference works conducted by the author (see §1.1 for the overall structure).

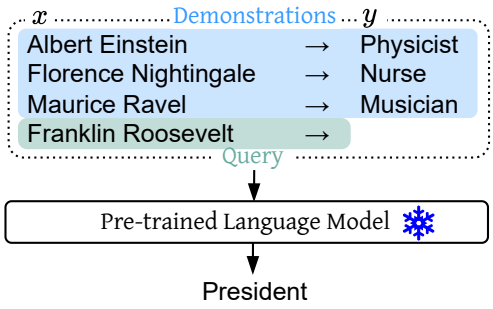


Figure 2.1: Diagram of In-context Learning with $k = 3$. A few-shot demonstration sequence and one query are concatenated to form an ICL input, and language models predict the next token as the label of the query.

2.2 Previous Works

We start by reviewing some related works for ICL. ICL is a young area, and we do not care about the downstream applications, e.g., applying ICL in the medical area. Also, to select the outstanding works, we primarily focus on the top conferences (ICLR, NeurIPS, ICML, ACL, NAACL, EMNLP).

As shown in Fig. 2.2, we mainly divide these works into 2 categories: efforts to improve the performance of ICL, and efforts to explain why ICL works in Transformer-based LMs. Notice that such categorization is not strict, since some studies primarily focus on mechanism analysis but also propose methods to improve ICL performance, like Yang et al. (2025c). Therefore, we classify these works into both categories.

2.2.1 Previous Works on Improving In-context Learning Performance

2.2.1.1 Pre-training & Model Tuning

Since LMs are typically pre-trained on plain natural language data (*wild data*) instead of ICL-styled data, it can be expected that a distribution gap between the pre-training and ICL testing occurs to prevent optimal performance. Therefore, some works focus on tuning LMs on the ICL data.

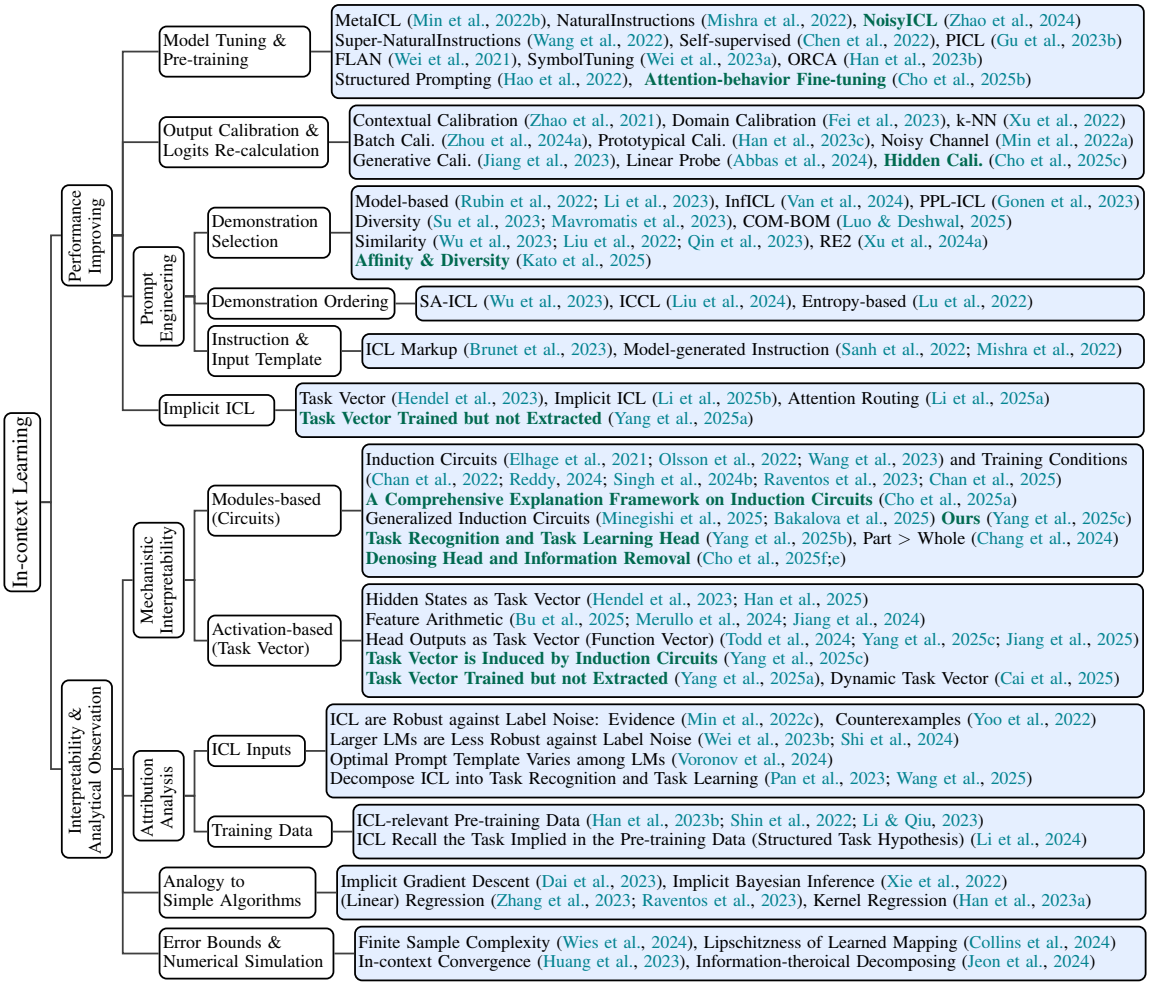


Figure 2.2: Taxonomy of previous works for in-context learning. The author's principal contributions to ICL area are on **bold and green**.

In detail, MetaICL (Min et al., 2022b) builds a dataset with 3.55M data samples from 142 raw NLP downstream task datasets, and fine-tunes GPT-2 Large to end-to-end causal language objectives on ICL-styled inputs constructed from these data. NaturalInstructions (Mishra et al., 2022) collects 193K data samples from 61 tasks, and fine-tunes BART-base (Lewis et al., 2020) on them. Super-NaturalInstructions (Wang et al., 2022) builds a dataset with 5M data samples and fine-tunes T5-11B with causal language objectives. Chen et al. (2022) fine-tunes the model in 4 self-supervised objectives on 1M data samples. Also, FLAN (Wei et al., 2021) instruct-tunes GPT3 on 2.49T tokens from 62 raw NLP datasets. Interestingly, SymbolTuning (Wei et al., 2023a) fine-tunes LMs on ICL inputs with *symbolic labels*, i.e., labels without effective semantics (e.g., "A", "B", "C", instead of "Good", "Normal", "Bad"). PICL (Gu et al., 2023a) re-pre-trains LMs with 80M data samples. Even though these methods are effective, they require gradient-based whole-model end-to-end training on enormous datasets, making them impractical for practical applications and contrary to ICL's original intention of low-resource learning.

For better efficiency, more works avoid conducting full-model tuning on large-scale datasets: Hao et al. (2022) directly modifies the position encoding and attention calculation to adapt LMs to more-shot ICL. ORCA-ICL (Han et al., 2023b) selects some critical data samples (*supportive data*) for ICL tasks from the pre-training datasets, and fine-tunes LMs on these supportive data for better efficiency. One of our works (Zhao et al., 2024) directly adds Gaussian noise to LMs' parameters for better performance and output calibration. Also, in this dissertation (§6), we will propose a novel method named **A**ttention **B**ehavior **F**ine-**T**uning (ABFT) (Cho et al., 2025b) to directly control the LM's behavior for better performance on less than 10% of the LM's parameters and only hundreds of data samples. Compared to all the methods before, this is a highly efficient method.

2.2.1.2 Output Calibration and Logits Re-calculation

Another perspective focuses on recalculating the output of LMs, toward an improvement or revolution for the restricted decoding. Min et al. (2022b) propose Noise Channel method, reversing the input templates into $[y_1, x_1, y_2, x_2, \ldots, y_*]$, where y_* is a possible label, then the generation probability of x_q is calculated as the logits for label y_*. Traversing all the possible labels, and the one with the largest logits is seen as the predicted label. (*Label token-based Calibration*) One type of previous work focuses on correcting the probability value from the restricted decoding, they estimate some correction factor A_l and B_l for each term of the restricted decoding results $\hat{o} = \{o_l\}_{l \in \mathbb{Y}}$, towards calibrated output probabilities $\hat{o}' = \{A_l o_l + B_l\}_{l \in \mathbb{Y}}$. In detail, Zhao et al. (2021) uses the reciprocal of predicted probabilities averaged from inputs with empty pseudo queries as the A_l, and sets $B_l = 0$. Fei et al. (2023) utilizes a similar methodology but on randomly sampled token sequences on the token distribution of the test dataset as the pseudo queries, instead of empty queries. Also, Jiang et al. (2023) uses generated pseudo queries by the LM to be calibrated, on a set of ICL-styled prefixes. Zhou et al. (2024a) catch the averaged predicted probabilities in a window during the inference processing as $-B_l$. Han et al. (2023c) estimate Gaussian clusters on the predicted probabilities distribution to form a Gaussian mixture model on some training samples, and re-calculate the logits on the trained model.

Breaking out of the framework of label token-based calibration, some works also suggest that only probabilities of label tokens from the restricted decoding are not informative enough. They propose *Vocabulary-based Calibration*, which trains some classifiers on the whole-vocabulary probabilities as the classification feature, instead of the label-token-only probabilities. Xu et al. (2022) utilizes a KNN classifier, and Abbas et al. (2024) utilizes a linear classifier. However, as shown in §5, we suggest that token probability-based classification methods, whether from the whole vocabulary or selected label tokens, are unable to achieve optimal classification performance, due to information loss and a limited degree of freedom for the decision boundary. Therefore, we propose a novel method named Hidden Calibration (Cho et al., 2025c), which directly recalculates on the last hidden states, instead of the token/vocabulary-based logits, to address such a problem, and reaches the state-of-the-art.

2.2.1.3 Prompt Engineering

Some works focus on improving the inputs (i.e., demonstrations, prompt templates, etc.) of ICL. The majority of such methods are heuristic-based, such as selecting better demonstrations (Xu et al., 2024a; Wu et al., 2023; Liu et al., 2022), re-ordering the demonstration examples (Lu et al., 2022; Liu et al., 2022; 2024), or adding task instructions into the inputs (Mishra et al., 2022; Sanh et al., 2022). In detail:

Selection of Demonstrations. Xu et al. (2024a) demonstrates that repeating only the query without a ground-truth label can improve the output of zero-shot inference. Rubin et al. (2022) trains a retriever model to select demonstrations from a demonstration pool, and Li et al. (2023) integrates training signals for multiple tasks for a unified retriever. Gonen et al. (2023) simply utilizes the "simplest" demonstrations, which produce the least perplexity from the LMs, as the selected demonstrations. Wu et al. (2023); Liu et al. (2022); Qin et al. (2023) utilize similarity-based methods to select demonstrations, i.e., selecting the most similar demonstrations to the query. In these methods, a serious concern is that they may work in classification tasks by exploiting the LM's prediction bias: that is, demonstrations similar to the query may share the same label as the query, and the LM's prediction may naturally be biased toward labels that appear more frequently in the context (Zhao et al., 2021). Mavromatis et al. (2023); Su et al. (2023) introduces diversity into demonstration selection practice, arguing that semantically diverse demonstrations may be more beneficial for ICL prediction. One of our work (Kato et al., 2025) combines the proposed similarity and diversity, resulting in a unified demonstration selection metric. Van et al. (2024) evaluate the saliency of each demonstration from Influence Function (Koh & Liang, 2017) to select the most important demonstration samples. Nevertheless, the author argues that research in this direction lacks generalizability (Voronov et al., 2024) and a systematic guiding theory, and that, since the task involves discrete combinatorial optimization, devising a non-heuristic method for demonstration selection optimization may be highly challenging. While, Luo & Deshwal (2025) proposes an optimization method for selecting demonstrations in ICL based on a semi-adversarial objective that combines Expected Calibration Error (Guo et al., 2017) and accuracy. In detail, to address several challenges: (1) the combinatorial nature of the selection problem and (2) the high computational cost of estimating accuracy and ECE for test instances via LM instance, the authors fit and improve Bayesian Optimization to this task.

Order of Demonstrations. Wu et al. (2023) searches for an ordering within a filtered demonstrations set that maximizes the LM's prediction confidence, i.e., the output probability of the finally predicted label. Liu et al. (2024) builds demonstration sequences from easy to hard, following the curriculum learning (Wang et al., 2021) framework. Lu et al. (2022) utilizes an entropy-based training objective to train an ordering model for sorting the selected demonstrations.

Instruction & Input Template. Mishra et al. (2022) indicates that textual instructions before demonstrations can adapt LM to tasks, and Sanh et al. (2022) builds a system to generate such textual instructions. Some works focus on the connector between parts of the input sequence, such as the "Label: ": Brunet et al. (2023) replace fixed connection tokens with trainable soft tokens optimized via gradient descent on ICL inputs.

2.2.1.4 Implicit ICL

Since a typical ICL requires a relatively long demonstration sequence, causing a quadratic cost against the token length, some works focus on reconstructing the influence of demonstrations to zero-shot queries by some inner steering technique on the LMs, whose token lengths are significantly reduced for better inference efficiency. Such works are quite overlapped with the Task Vector perspective for interpretability (mentioned below), in detail: Brunet et al. (2023) extracts the hidden state of the last token in few-shot inference on a certain layer and adds the extracted hidden state to the zero-shot last-token hidden states can partly reconstruct the accuracy of few-shot inference. Li et al. (2025b) conducts a linear combination of label tokens' hidden states, and injects such a combined vector into zero-shot queries. Li et al. (2025a) focuses on reconstructing the attention pattern presented in few-shot inputs on some special attention heads in zero-shot inputs. Moreover, one of our work Yang et al. (2025a) directly trains a bias term on any location (layer or token location) in the LMs, and observes a significant accuracy improvement.

2.2.2 Previous Works on the Inner Mechanism of In-context Learning

2.2.2.1 Mechanistic Interpretability

Some works aim to identify the components within LMs (can be specific modules or activations/hidden states) that make significant contributions to ICL to reconstruct the ICL behavior by a pipeline consisting of such components. This paradigm is referred to as *Mechanistic Interpretability*. Based on the taxonomy of the aforementioned modules or activations, we list the representative works in this area as follows:

Subnetworks-based. Based on identifying the significant modules, some works are dedicated to reconstructing LMs into minimal sub-networks containing human-understandable algorithms. Such decomposition is also named "*circuit*" in the context of Mechanistic Interpretability. In the ICL domain, such works are inclined toward *Induction Circuits* (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2023). In detail, the induction circuits consist of two kinds of attention heads: the previous-token heads and the induction heads, perform an "in-context matching and continuation" task: for an input sequence like [C, A, B, C, A], when generating the next token after the second A, the induction circuits attends to the previous occurrence of A in the context and steers the output towards its following token B. Such an operation is close to what should be revoked by the input format of ICL: concretely, if we analogize C to the input text, A to a repeated structured template token (e.g., "Label: "), and B to the label token, the entire sequence corresponds to an ICL-style input. Therefore, such induction circuits are widely employed to interpret the mechanism of ICL. In detail:

Elhage et al. (2021) first demonstrates the existence of induction heads, and shows that ablating these heads significantly degrades ICL performance. They also find the 2-stage nature of induction circuits, i.e., the first stage is to copy the previous token's information to the next token, and the second stage is to match the current token with the copied previous tokens' information, corresponding to the two attention patterns of heads termed as previous-token heads and induction heads, respectively. Olsson et al. (2022) further demonstrates the training dynamics of ICL, and suggests that induction heads develop at the same point as a sudden increase in ICL performance, serving as a strong causal evidence that induction heads are crucial for ICL. Wang et al. (2023) reveals the layer distribution of induction heads, demonstrating that induction heads are primarily concentrated in the middle and late layers of the model, while early layers contain few induction heads, which conducting the semantic encoding of the input text.

Chan et al. (2022); Reddy (2024); Singh et al. (2024b); Raventos et al. (2023); Chan et al. (2025) investigate the pre-training conditions for the emergence of induction heads. They demonstrate similar conclusions that *contextual inference dependency* (i.e., the conditions where contextual information is necessary for predicting the next token) in the training data is a crucial factor for the emergence of induction heads. In practice,

these conditions with dependency contain the label diversity (i.e., the size of label space), the burstiness of labels (i.e., the existence of rare labels in the demonstrations), the diversity of inputs x (i.e., the variance of the embedded x), etc. More diverse and complex data distributions tend to induce the emergence of ICL inference dynamics and induction heads, and less diverse and complex data induces the **I**n-**w**eight **L**earning (IWL). Chan et al. (2022) first demonstrates such conclusions by synthetic scenarios based on image embedding. Reddy (2024) further suggests that the reason of the emergence of ICL is the suppression of IWL by the complex pre-training data: IWL exhibits a fast convergence but fails to fit complex or noisy training data, whereas ICL shows a slower, two-stage convergence process but can fit complex data effectively. During training, both the IWL and ICL circuits are optimized simultaneously: on simple data, the IWL circuit quickly achieves zero loss, leading the training to stop, while on complex data, the opposite occurs. Singh et al. (2024b) conducts a fine-grained investigation, dividing the induction circuits into the aforementioned 2 sub-operations, and observes the training dynamics separately. Chan et al. (2025) provides a theoretical explanation for the emergence of ICL behavior against IWL behavior.

However, the current observations can not fit the mechanistic interpretation of ICL perfectly towards the observed phenomena (mentioned in §2.2.2.2). Therefore, in ICLR 2025, we propose a comprehensive and unified mechanistic interpretation for ICL based on induction circuits, which can well explain the scattered observations (Cho et al., 2025b). We will retell the story of this paper in §3. While, such an interpretation is not perfect: the flaw is rooted in the definition of induction heads, which defines a copy process requiring a *source token* to copy from, causing the interpretation to fail in *unseen-label* conditions, where the ground-truth label of the query, which should be copied, has never appeared in the demonstrations. In such scenarios, the theory predicts a failure of ICL, but in practice, LMs can still perform ICL to some extent under unseen-label conditions. To address this flaw, some works attempt to find some attention heads with highly similar attention patterns to the induction heads but semantically different behavior or function, which actually generalize the functionality of induction heads, such as: Minegishi et al. (2025), who find a similar 2-stage inference process under a synthetic unseen-label setting, and Bakalova et al. (2025) for real-world datasets and pre-trained LMs. Also, one of our works (Yang et al., 2025b) suggests that the induction heads do not mainly copy the correct label, but rather help the LMs to recognize the defined task from the context.

To address the aforementioned flaw, another of our works (Cho et al., 2025b) (§4) suggests that the induction heads are not the only crucial component for ICL, but rather a supplementary task-oriented information removal mechanism conducted by *denoising heads*, which remove the irrelevant information in the query to steer the prediction toward the task specified by the demonstrations. Such a mechanism dominates the ICL processing in unseen-label conditions, where the induction heads fail, and is also significant in typical (seen-label) conditions. Therefore, the ICL processing is a combination of induction heads and denoising heads, which can well explain observations. Moreover, Chang et al. (2024) suggests that removing some specific components can improve ICL performance, which is also found by us in §4 as a side production.

Model Activation-based. Some studies argue that certain specific numerical components within the LM's activation (hidden states) induce the ICL outputs, and therefore refer to these components that guide the LMs toward the task as *Task Vectors* (the term Task Vector originates from the difference in parameters between a fine-tuned model on a specific task and its pre-trained counterpart, typically constructed via Linear Mode Connectivity (Neyshabur et al., 2020; Ilharco et al., 2023), and generalized to ICL scenario). Hendel et al. (2023) demonstrates that extracting the last-token hidden states from few-shot inputs and injecting such hidden states into the last-token hidden states of zero-shot inputs improves the zero-shot accuracy; on this basis, they regard it as the task vector that guides ICL. Such usage of full hidden states as a task vector is also found in (Han et al., 2025). Moreover, some works demonstrates that such task vectors follows the feature arithmetic in the fact recall tasks (Bu et al., 2025; Merullo et al., 2024; Jiang et al., 2024), where the task vector conduct the transform action from one entity to another entity with a fixed relation, e.g., in "Tokyo" $-$ "Japan" $+$ "Germany" $\approx$ "Berlin", the ("Tokyo" $-$ "Japan") is the task vector.

Beyond the full hidden states, some works focus on specific subspaces of the hidden states as task vectors, which are typically extracted from the outputs of attention heads, termed *Function Vector* (Todd et al., 2024; Yang et al., 2025c; Jiang et al., 2025). They claim that such function vectors are better task representations, improving zero-shot performance more significantly. Moreover, Cai et al. (2025) combines the outputs of multiple attention heads to form a dynamic task vector, and one of our works (Yang et al., 2025c) finds the special geometric effects of the function vectors from induction heads, i.e., increases the probability of the output to the label spaces defined by the demonstrations. Also, another of our works (Yang et al., 2025a) actively trains task vectors instead of passively extracting components from hidden states for a more pure task representation, enabling further analysis.

2.2.2.2 *Attribution Analysis*

As a traditional interpretability perspective (i.e., *saliency* (Sun et al., 2021)), another line of work mainly explores how parts of the input or training data influence ICL performance. In detail:

Input Attribution. Min et al. (2022c) evaluates the saliency of each component in ICL inputs, demonstrating that the correctness of labels presented in demonstrations is less significant to ICL accuracy, i.e., randomly flipping the demonstration label tokens (e.g., "Positive"↦"Negative") minorly influences the prediction accuracy, which is a significant difference of ICL between the IWL, where the noisy training labels block the inference (Zhang et al., 2021). Moreover, they find that the more important factors of ICL accuracy are label spaces (i.e., the query's ground-truth label should be contained in the demonstrations' label space) and fixed format, which will be unified into one interpretation framework by our work in §3. However, Yoo et al. (2022) finds that the insensitivity against the label correctness is dataset-wise, i.e., on some datasets and models, the ICL accuracy is sensitive to label correctness. Moreover, Wei et al. (2023b); Shi et al. (2024) demonstrate and explain that larger LMs are more sensitive to label correctness, i.e., if the labels are flipped, larger LMs are more likely to produce flipped outputs. Voronov et al. (2024) shows that the optimal prompt templates are not generalizable among various LMs. Pan et al. (2023) decomposes the ICL inference into 2 phases: task recognition, where LMs extract the task information from the demonstration label space (Yang et al., 2025b), irrelevant to the label correctness; and task learning, where LMs extract the expression pattern of the task results from the demonstration-label pairs, helping some special scenario such as the symbolic/flipped label settings. Wang et al. (2025) investigates the emergence of task recognition and task learning during the pre-training, and Yang et al. (2025b) localizes the task recognition and task learning on the LM components.

Training Data Attribution. Besides the works on the data condition evoking the induction head mentioned before, some works focus on more empirical and global analysis on full ICL behavior and accuracy, instead of single attention heads, for example: Han et al. (2023b) find the most helpful fine-tuning data by identifying the data samples from a plain corpus producing the most similar gradient with the ICL inputs. Then, they characterize the properties of such selected data samples as: non-relative to the downstream task defined by the ICL inputs utilized in the identifying process, but they contain more bursty tokens (i.e., rarely occurring, long-tail frequency tokens, notice that this is the second time we mention this concept, therefore, it may be an essiential factor for ICL), and are more contextually related. Such results almost align with the conclusion from the label condition investigations. However, an earlier work suggests that the ICL performance is related to the domain of pre-training data (Shin et al., 2022). Li & Qiu (2023) proposes an improved pipeline to identify these significant data for ICL, but no further analysis is made.

2.2.2.3 *Analogy to Simple Algorithms*

Some studies attempt to analogize ICL to simpler algorithms, which is somewhat similar to mechanistic interpretability in that both aim to reduce ICL to simpler processing. However, these analogy works do not explicitly identify which model components induce such a process, which is the major difference from mechanistic interpretability. In detail, Dai et al. (2023) proposes an analogy between attention calculation and gradient descent, suggesting that the demonstrations serve as training samples, and the query represents the test sample. This analogy is elegant, but it fails to precisely capture the phenomena observed earlier: specifically, we know that gradient descent is highly sensitive to label noise (Zhang et al., 2021); therefore, if ICL were indeed a gradient descent, it should also be sensitive to label noise. However, ICL is stable against label noise (Min et al., 2022c), which is contradictory. Also, (Han et al., 2023a) suggests that ICL can be explained as a kernel regression between x and y, while such a kernel regression regards all the demonstrations equally, but is negatived by Zhao et al. (2021). Moreover, some works utilize linear regression (Zhang et al., 2023; Raventos et al., 2023) and Bayesian inference (Xie et al., 2022) to explain ICL, which serve as the foundation of more abstract theoretical analysis.

2.2.2.4 *Error Bounds and Numerical Simulation*

More abstractly, some works focus on the theoretical guarantee of ICL performance or emergence in the training dynamics. Wies et al. (2024) proposes a Probably Approximately Correct framework for in-context learnability, suggesting that when the pretraining distribution is a mixture of latent tasks (a model often considered for natural language pretraining), these tasks can be efficiently evoked via in-context learning. In other words, this work suggests that ICL is not a learning process on new tasks or behaviors, but recognizing or evoking the task specified in the demonstrations which is already learned in the pre-training, aligning with the related works (Pan et al., 2023; Yang et al., 2025b). Collins et al. (2024) examines the mechanistic

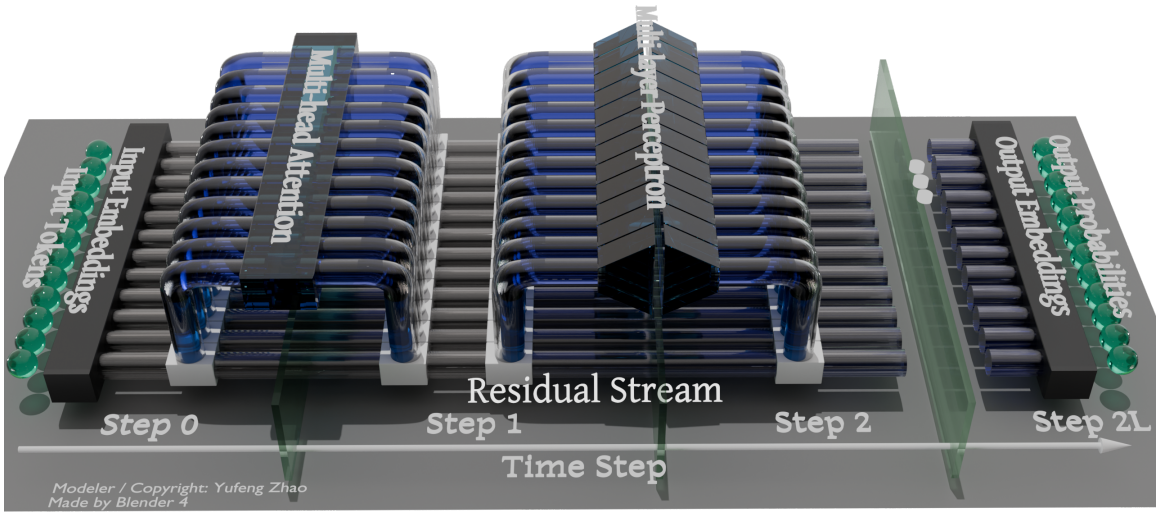


Figure 2.3: Schematic diagram of the Transformer inference process.

function of attention modules, revealing that attention calculation can implement a nearest-neighbor predictor, and dynamically adjust its receptive window based on task Lipschitzness and label noise. Huang et al. (2023) investigates the learning dynamics of transformers trained via gradient descent for in-context regression tasks. Jeon et al. (2024) introduces information-theoretic frameworks that unify ICL and meta-learning through a general decomposition of the prediction error bounds.

2.3 Preliminary Knowledge

The readers of this doctoral dissertation are expected to have basic knowledge in the following fields: Natural Language Processing and Deep Learning, particularly the computational processes of the attention mechanism and Transformer models (Vaswani, 2017). Due to space limitations, the author will not elaborate on all the fundamental knowledge: some relevant materials are cited here to supplement the foundational knowledge. The author believes these sources provide a more effective introduction to the subject than he does.

- **Multi-head Attention & Transformer Feed-forward Calculation**: the original work of Transformer (Vaswani, 2017), a new perspective of calculation of Transformer (Elhage et al., 2021).
- **In-context Learning**: the original work of ICL (Radford et al., 2019), a global survey on ICL (Dong et al., 2022), and a survey on the inner mechanism research of ICL (Zhou et al., 2024b).
- **Mechanistic Interpretability**: a survey on mechanistic interpretability can be found in (Rai et al., 2024).

Here, we briefly recall the feed-forward calculation of Transformer models and mechanistic interpretability to agree on some terms. If the reader is already familiar with this background, they can skip this section without loss of understanding of this dissertation.

2.3.1 The Feed-forward Calculation of Transformer

As shown in Fig. 2.3, the main body of a Transformer model is a *residual stream* that begins with the *Input Embedding* and ends with the *Output Embedding*. Then, Transformer modules, primarily containing *multi-head self-attention* and *multi-layer perceptron*, sequentially read and write information in the residual stream.

Residual Stream. Residual stream is a sequence of matrices $H^{(i)}$ in a single vector space, evolving through successive residual updates. Originating from the residual connection (He et al., 2016), i.e., in the Transformer model, every module is equipped with a skip connection, causing a shortcut path from the input embedding directly to the output embedding, which we call the residual stream. Every other module in the Transformer model, often called *residual branch*, acts on the residual stream, causing an update of the

contents in the residual stream from $H^{(i)}$ to the next time step $H^{(i+1)}$.

Input Embedding. An input embedding $\mathcal{E} : [0, |\mathbb{V}|] \subset \mathbb{N}^{n_t} \to \mathbb{R}^{n_t}$ transfers a token sequence with length n_t (represented by integer indices in the range of vocabulary size $|\mathbb{V}|$) into a high-dimensional real matrix, as the initial representations of the input text, and also the original content of the residual stream. In detail, for an input token sequence $X = [x_1, x_2, \ldots, x_{n_t}]$ tokenized from a natural language input, where $x_i \in [0, |\mathbb{V}|] \subset \mathbb{N}^{n_t}$ are the token index, the input embedding return a $H^{(0)} = \mathcal{E}(X) \in \mathbb{R}^{d \times n_t}$, where d is the dimension of the embedding space (alias: inner dimension; model's dimension). Typically, this process is context-independent, that is, each vector $H_i^{(0)}$ in the n_t dimension is independent of the $x_j, i \neq j$, or: $\forall i, H_i^{(0)} = \mathcal{E}([x_i])$. Also, the input embeddings are merged with the positional embedding, which is a fixed matrix truncated on the column n_t; for the sake of simplicity, we ignore it in this work.

Transformer Modules. As shown in Fig. 2.3, Transformer modules, composed of multi-head self-attention modules and multi-layer perceptron modules, read information from the residual stream of a certain time step $H^{(i)}$ and conduct some modification as $f(H^{(i)})$, then update the $H^{(i)}$ to $H^{(i+1)} = H^{(i)} + f(H^{(i)})$. Residual stream on one time step $H^{(i)}$ is only accessed and updated by one module. The multi-head self-attention modules access even-indexed $H^{(2t)}$ and multi-layer perceptron modules access odd-indexed $H^{(2t+1)}$, i.e., they are alternately stacked; a multi-head self-attention module and the following multi-layer perceptron module form a Transformer layer, i.e., the module accessing the $H^{(i)}$ is in the $\lfloor i/2 \rfloor$-th layer. We call the even-indexed $H^{(2t)}$ *the hidden states of the t-th layer*. Every module is followed by a layer normalization (Ba et al., 2016); for the sake of simplicity, we ignore it in the context of interpretability.

- **Multi-head Self-attention.** A multi-head self-attention is a sum of n_h (the n_h is a model hyperparameter) attention mechanism (termed *attention head*). The calculation j-th attention head in the i-th layer is:

$$\text{head}_{i,j}\left(H^{(2i)}\right) = W_{O,i,j} W_{V,i,j} H^{(2i)} \text{softmax}\left(\frac{H^{(2i),\top} W_{Q,i,j}^{\top} W_{K,i,j} H^{(2i)}}{\sqrt{d/n_h}}\right), \tag{2.2}$$

  where $W_{Q,i,j}, W_{K,i,j}, W_{V,i,j}, W_{O,i,j}^{\top} \in \mathbb{R}^{d/n_h, d}$, are model parameters. $W_{Q,i,j}, W_{K,i,j}, W_{V,i,j}$ defines the input subspace of this attention head, and $W_{O,i,j}$ defines the output subspace of this attention head. The final update of the multi-head self-attention module is calculated as $\sum_{j=1}^{n_h} \text{head}_{i,j}$.

  Causal Mask. For the decoder-type Transformer (which is the primary focus of this work), this operation requires an additional step: multiplying the attention logits (i.e., the term inner the softmax function) by a lower-triangular matrix Γ as shown below to prevent information of later tokens from flowing to earlier tokens. This is known as the *causal mask* shown below:

$$\Gamma = \begin{bmatrix} 1 & -\infty & -\infty & -\infty & \cdots & -\infty \\ 1 & 1 & -\infty & -\infty & \cdots & -\infty \\ 1 & 1 & 1 & -\infty & \cdots & -\infty \\ 1 & 1 & 1 & 1 & \cdots & -\infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}, \qquad \text{where } \Gamma_{ij} = \begin{cases} 1, & i \geqslant j, \\ -\infty, & i < j. \end{cases} \tag{2.3}$$

  The masked attention is then computed as:

$$\text{head}_{i,j}\left(H^{(2i)}\right) = W_{O,i,j} W_{V,i,j} H^{(2i)} \text{softmax}(\Gamma \cdots), \tag{2.4}$$

- **Multi-layer Perceptron.** A multi-layer perceptron handle $H^{(i)} \in \mathbb{R}^{d \times n_t}$ column-wisely, produce the update as:

$$\text{MLP}_i\left(H^{(2i+1)}\right) = W_{\text{out},i} \mathcal{F}\left(W_{\text{in},i} H^{(2i+1)}\right), \tag{2.5}$$

  where $W_{\text{out},i}^{\top}, W_{\text{in},i} \in \mathbb{R}^{d' \times d}$, are model parameters. $\mathcal{F}$ is the activation function, mainly ReLU and its variants. Typically, $d' > d$.

Output Embedding. An output embedding $\mathcal{E}^U : \mathbb{R}^d \to \Lambda^{|\mathbb{V}|}$ acts on the last time step of the residual stream, transferring each column vector of $H^{(2L)}$ into a $|\mathbb{V}|$-dimensional probability simplex, serving as the output probabilities of the current next token among the vocabulary. Typically, an output embedding is composed

of an output embedding matrix $E^U \in \mathbb{R}^{|\mathbb{V}| \times d}$ and a column-wise $\mathrm{softmax}$ function, i.e., $\mathcal{E}^U(H^{(2L)}) = \mathrm{softmax}(E^U H^{(2L)})$.

According to the description above, the output of a Transformer model with L layers can be written as:

$$o = \mathcal{E}^U \left(H^{(0)} + \sum_{i=0}^{L-1} \left(\sum_{j=1}^{n_h} \mathrm{head}_{i,j} \left(H^{(2i)} \right) + \mathrm{MLP}_i \left(H^{(2i+1)} \right) \right) \right). \tag{2.6}$$

2.3.2 Mechanistic Interpretability

The objective of mechanistic interpretability is to find a subset of Transformer modules that is mainly responsible for the outputs from specific inputs. Such a subset of Transformer modules naturally forms a network, called *circuit*, since any two modules that access the residual streams at different time steps are connected, provided that their read and write subspaces are not strictly orthogonal.

Modules in one circuit interact through certain semantical components in the residual stream, named *features*, and generate new features. Therefore, analyses of such circuits largely focus on decoding the semantics of these features, and observing how the features are transformed in the circuit processing, to localize the functionality of each part in the circuits. Typical methodologies include (1) directly decoding the semantics of features as tokens (nostalgebraist, 2020; Belrose et al., 2023; Ghandeharioun et al., 2024), (2) untangling the hidden states' superposition (Elhage et al., 2022) for atomized feature interpretation (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2024; Gao et al., 2025; Bussmann et al., 2024; Rajamanoharan et al., 2024a;b; Shi et al., 2025; Lindsey et al., 2024; Cho et al., 2025e) (also termed *dictionary learning* (Shu et al., 2025)), (3) unsupervised clustering on these features, such as manifold learning methods (McInnes et al., 2018), (4) measuring customized metrics on these features (Yang et al., 2025c; Cho et al., 2025f), etc.

Moreover, automatic circuit discovery methods (Conmy et al., 2023; Hsu et al., 2025) have been developed, and causal inference methods based on ablation (Ghorbani & Zou, 2020; Li & Janson, 2024) or steering (Meng et al., 2022) are generally applicable for assessing the robustness and causality of the outputs of the extracted circuits.

Part II
Theory

II Theory **13**

3 Mechanistic Basis for In-context Learning **14**
- 3.1 Introduction . . . 14
- 3.2 Preparation . . . 15
  - 3.2.1 Background & Related Works . . . 15
  - 3.2.2 Experiment Settings . . . 16
- 3.3 Step 1, Input Text Encode: Semantics Encoding as Linear Representations in Hidden States . . . 16
  - 3.3.1 LLMs Encode Input Text on Forerunner Tokens in Hidden States . . . 16
  - 3.3.2 Input Text Encoding is Linear and Task-Relevant but Position-biased . . . 17
- 3.4 Induction Circuits in Large Language Models . . . 18
  - 3.4.1 Step 2, Forerunner Token Head: Copy from Text Feature to Label Token . . . 18
  - 3.4.2 Step 3, Induction Head: Feature Retrieval on Task Subspace . . . 19
- 3.5 Putting Things Together . . . 21
  - 3.5.1 Ablation Analysis . . . 21
  - 3.5.2 Bypass Mechanism . . . 21
  - 3.5.3 Explanation towards Observed ICL Phenomena . . . 22
- 3.6 Conclusion and Discussion . . . 22

4 Supplementary Mechanism Basis for ICL in Unseen Label Scenario **24**
- 4.1 Introduction . . . 24
- 4.2 Background . . . 26
- 4.3 Method: Tracing the Task-oriented Information Removal . . . 26
- 4.4 Task-oriented Information Removal in In-context Learning . . . 27
  - 4.4.1 Experiment Settings . . . 27
  - 4.4.2 Step 1: Finding the Explicit TVS in Zero-shot Hidden States . . . 27
  - 4.4.3 Step 2: Tracing the Implicit Task-oriented Information Removal . . . 28
  - 4.4.4 Step 3: Causality Test by Denoising Heads . . . 30
- 4.5 Conclusion and Discussion . . . 32

Chapter 3

Mechanistic Basis for In-context Learning

Revisiting In-context Learning Inference Circuit in Large Language Models

This chapter mainly retells the paper "Revisiting In-context Learning Inference Circuit in Large Language Models" in ICLR 2025 by Yufeng Zhao (alias: Hakaze Cho) (refer to Publication List).

Abstract

In-**c**ontext **L**earning (ICL) is an emerging few-shot learning paradigm on **L**anguage **M**odels (LMs) with inner mechanisms un-explored. There are already existing works describing the inner processing of ICL, while they struggle to capture all the inference phenomena in large language models. Therefore, this chapter proposes a comprehensive circuit to model the inference dynamics and try to explain the observed phenomena of ICL. In detail, we divide ICL inference into 3 major operations: **(1) Input Text Encode**: LMs encode every input text (in the demonstrations and queries) into linear representation in the hidden states with sufficient information to solve ICL tasks. **(2) Semantics Merge**: LMs merge the encoded representations of demonstrations with their corresponding label tokens to produce joint representations of labels and demonstrations. **(3) Feature Retrieval and Copy**: LMs search the joint representations of demonstrations similar to the query representation on a task subspace, and copy the searched representations into the query. Then, language model heads capture these copied label representations to a certain extent and decode them into predicted labels. Through careful measurements, the proposed inference circuit successfully captures and unifies many fragmented phenomena observed during the ICL process, making it a comprehensive and practical explanation of the ICL inference process. Moreover, ablation analysis by disabling the proposed steps seriously damages the ICL performance, suggesting the proposed inference circuit is a dominating mechanism. Additionally, we confirm and list some bypass mechanisms that solve ICL tasks in parallel with the proposed circuit.

3.1 Introduction

In-**C**ontext **L**earning (ICL) (Radford et al., 2019; Dong et al., 2022) is an emerging few-shot learning paradigm: given the *demonstrations* $\{(x_i, y_i)\}_{i=1}^{k}$ consisting of [input text]-[label token] pairs and a *query* x_q, **L**anguage **M**odels (LMs) take the sequence $[x_1][s_1][y_1]\dots[x_k][s_k][y_k][x_q][s_q]$[2] (Fig. 3.1) as input and then predicts the label for x_q by causal language modeling operation. Typically, the label tokens y_i are preceded by and also predicted on *forerunner tokens* s_i (e.g., the colon in "Label: "). ICL has aroused widespread interest, but its underlying mechanism is still unclear.

There have been theoretical or empirical trials to characterize and explain the inference process of ICL (Xie et al., 2022; Dai et al., 2023; Wang et al., 2023; Han et al., 2023a; Jeon et al., 2024; Zheng et al., 2024). However, to capture all the operating dynamics and many fragmented observed interesting phenomena of ICL in **L**arge **L**anguage **M**odels[3] (LLMs), a more comprehensive characterization is still necessary. Therefore, this chapter tries to propose a unified inference circuit and measures various properties in LLMs for confirmation of the observed ICL phenomena.

[2] In this chapter, we denote *tokenization* as $[\cdot]$, and *token concatenating* as $[\cdot][\cdot]$.

[3] *Large* refers to scaled LMs trained by natural language data, such as Llama 3 (AI@Meta, 2024), contrast to simplified work that uses simple models trained and test on well-embedded input in toy models.

In detail, as shown in Fig. 3.1, we decompose ICL dynamics into 3 atomic operations on Transformer layers. **Step 1: INPUT TEXT ENCODE**: LMs encode each input text x_i into linear representations in the hidden state of its corresponding forerunner token s_i. **Step 2: SEMANTICS MERGE**: For demonstrations, LMs merge the encoded representations of s_i with the hidden state of the corresponding label tokens y_i. **Step 3: FEATURE RETRIEVAL AND COPY**: LMs retrieve the similar semantics-merged label representations $y_{1:k}$ (from Step 2) to the query representation s_q in a task-relevant subspace, then copy them into the query's forerunner token representation. Finally, LM heads predict the label for x_q using the label-attached query representation s_q. Steps 2 and 3 form a typical induction circuit, which is a key mechanism of ICL but only examined in synthetic scenarios (Elhage et al., 2021; Singh et al., 2024b; Reddy, 2024).

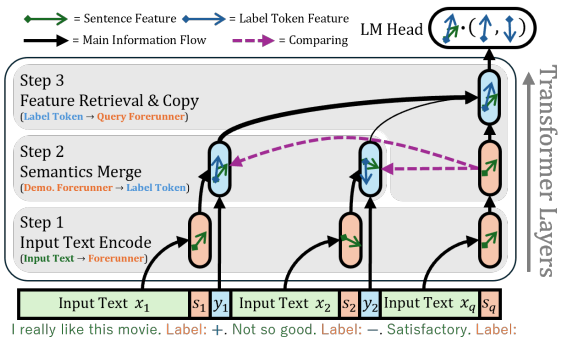


Figure 3.1: The 3-phase inference diagram of ICL. **Step 1**: LMs encode every input text into representations, **Step 2**: LMs merge the encoded text representations of demonstrations with their corresponding label semantics, **Step 3**: LMs retrieve merged label-text representations similar to the encoded query, and copy the retrieved representations into the query representation.

We empirically find evidence for the existence of each proposed step in LLMs, and conduct more fine-grained measurements to gain insights into some phenomena observed in ICL scenarios, such as (1) *positional bias*: the prediction is more influenced by the latter demonstration (Zhao et al., 2021), (2) *noise robustness*: the prediction is not easy to be affected by demonstrations with wrong (*noisy*) labels (Min et al., 2022c), while larger models are less robust to label noise (Wei et al., 2023b), and (3) *demonstration saturation*: the accuracy improvements plateau when sufficient demonstrations are given (Agarwal et al., 2024; Bertsch et al., 2024), etc. (discussed in §3.5.3). Moreover, we find multiple bypass mechanisms for ICL conducted by residual connections, while the 3-phase dynamics remain dominant.

Our contributions can be summarized as: (1) We propose a comprehensive 3-step inference circuit to characterize the inference process of ICL, and find empirical evidence of their existence in LLMs. (2) We conduct careful measurements for each inference step and successfully capture a large number of interesting phenomena observed in ICL, which enhances the practicality of the proposed circuit. (3) Our ablation analysis suggests that the proposed circuit dominates, but some bypass mechanisms exist in parallel to perform ICL, and we introduce some of these bypasses along with their empirical existence evidence.

3.2 Preparation

3.2.1 Background & Related Works

In-context Learning. Discovered by Radford et al. (2019), ICL is an emerging few-shot learning paradigm with only feed-forward calculation in LMs. Given demonstrations $\{(x_i, y_i)\}_{i=1}^{k}$ composed of input-label pairs and a query x_q, typical ICL creates an input $[x_1][s_1][y_1]\dots[x_k][s_k][y_k][x_q][s_q]$, with some structural connectors (e.g. "Label: ") including forerunner token s_i (e.g. ": "), as shown in Fig. 3.1. LMs receive such inputs and return the next token distribution, where the label token with the highest likelihood is chosen as the prediction. Explaining the principle of ICL is an open question, although there have been some efforts on **the relationship between ICL capacity and pre-training data** (Li & Qiu, 2023; Singh et al., 2024b;a; Gu et al., 2023b; Han et al., 2023b; Chan et al., 2022), the **feature attribution of inputs** (Min et al., 2022c; Yoo et al., 2022; Pan et al., 2023; Kossen et al., 2024), and **reduction to simpler processes** (Zhang et al., 2023; Dai et al., 2023; Xie et al., 2022; Han et al., 2023a). However, a comprehensive and unified explanation on real-world LMs is still needed to capture the operating dynamics of ICL.

Induction Circuit. Introduced by Elhage et al. (2021), an induction circuit is a pair of two cooperating attention heads from two transformer layers, where the "previous token head" writes information about the previous token to each token, and the "induction head" uses the wroten information to identify a token that should follow each token. Concretely, such a function is implemented by two atomic operations in attention calculations: (1) copy the representation of the previous token [A] to the next token [B], and (2) retrieve and copy similar representations on [B] (copied from [A]) to the current token [A′]. Concisely, it performs inference in the form of $[A][B]\dots[A'] \Rightarrow [B]$, which is similar to the ICL diagram. Therefore, the induction

circuit has been widely used to explain the inference dynamics of ICL (Wang et al., 2023) and the emergence of ICL during pre-training (Olsson et al., 2022; Reddy, 2024; Singh et al., 2024b). Despite their valuable insights, these experiments rely on a synthetic setting: using simplified models and well-embedded (linearly separable) inputs, which differs from the practical ICL scenario using real-world LMs with many layers and complicated inputs: we believe these synthetic works indicate the *potential* of Transformers or a locally optimal behavior, and can not demonstrate that their observation exists in scaled LLMs trained on wild data. So, in this chapter, we try to bridge the gap between the synthetic and real-world settings with more detailed observations for the inference of ICL on real-world LLMs.

3.2.2 Experiment Settings

Models. We mainly conduct experiments on 4 modern LLMs: Llama 3 (8B, 70B) (AI@Meta, 2024), and Falcon (7B, 40B) (Almazrouei et al., 2023). Unless specified, we report the results on Llama 3-70B, since its deep and narrow structure (80 layers, 64 heads) makes it easier to show hierarchical inference dynamics (discussed in §3.5.2). The results of other models can be found in Appendix A.8.2.

Datasets. We build ICL-formed test inputs from 6 real-world sentence classification datasets, and unless specified, we report the average results on them: SST-2 (Socher et al., 2013), MR (Pang & Lee, 2005), Financial Phrasebank (Malo et al., 2014), SST-5 (Socher et al., 2013), TREC (Li & Roth, 2002; Hovy et al., 2001), and AGNews (Zhang et al., 2015).

Others. Unless specified, we use $k = 4$ demonstrations in ICL inputs. For each dataset, we randomly sample 512 test data points and assign one fixed demonstration sequence for each test sample to form a test input. About the prompt templates, etc., please refer to Appendix A.1.1.

3.3 Step 1, Input Text Encode: Semantics Encoding as Linear Representations in Hidden States

This section mainly confirms that LMs construct task-relevant and linearly separable semantic representations for every input text (demonstrations and queries) in the hidden states. Such linear representations are an important foundation for explaining the dynamics of ICL based on induction heads, since attention-based feature retrieval, a key mechanism of induction heads, can be easily done on linear representations. Current successful studies on simplified models and inputs (Chan et al., 2022; Reddy, 2024; Singh et al., 2024b) also assume the existence of such linear representations. Moreover, we confirm some interesting properties of the encoded input text representations: (1) It is based on the capacity in the model weights and can be enhanced by demonstrations in context. (2) The similarity of representations is biased towards the encoding target's position.

3.3.1 LLMs Encode Input Text on Forerunner Tokens in Hidden States

We first study the existence of input text encoding in hidden states and then explain their linear separability and task relevance in §3.3.2. For each text-label pair (x_t, y_t) (*encoding target*) sampled from the datasets, we prepend them with k demonstrations, resulting in augmentated (by label $[y_t]$) ICL-styled inputs $[x_1][s_1][y_1]\dots[x_k][s_k][y_k][x_t][s_t][y_t]$. These inputs are then fed into an LM to extract the hidden states of a specific token in $[x_t][s_t][y_t]$ from each layer, serving as the ICL inner representations. To assess the quality of these representations as sentence representations, we use the sentence embedding of x_t encoded by BGE M3 (Chen et al., 2024), a SotA encoder-only Transformer, as a *reference representation* and then calculate the *mutual nearest-neighbor kernel alignment*[4] (Huh et al., 2024) between these representations. See Appendix A.1.1 and A.1.2 for details.

Forerunner Tokens Encode Input Text Representations. We plot the kernel alignment on the 3 types of tokens in Fig. 3.3 (Left), where the forerunner token, while often overlooked in previous work, produces the best input text encoding, emerging in the early phase (layers 0-28) of the inference process, and keeping a high level to the end of inference. Interestingly, hidden states of label words are not satisfactory input text representations even with a high background value (the result at layer 0, refer to Appendix A.1.2.1), which is a

[4]Intuitively, kernel alignment measures similarity between two representations toward the same datasets, and according to (Huh et al., 2024), a higher cross-model kernel alignment usually means a better representation.

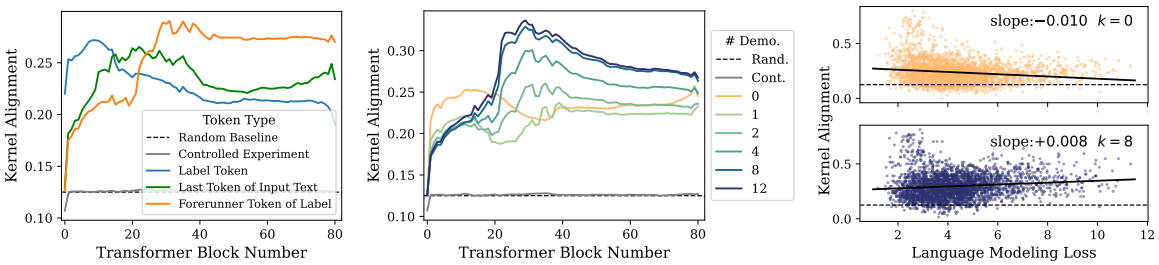


Figure 3.3: Input text encoding magnitudes (metricized by kernel alignment with feature encoded by an encoder-structured model) of hidden states in various layers in ICL scenario (The controlled experiments are results between current 6 datasets and TEE (Mohammad et al., 2018)). **Left**: Encoding magnitudes on hidden states from various types of token. **Middle**[5]: Encoding magnitudes with different k on the forerunner tokens. **Right**: Encoding magnitudes in layer 24 of Llama 3-70B against the causal language modeling loss of the input text with (upper) $k = 0$ and (lower) $k = 8$.

critical supplement to previous work which suggests the label tokens are pivots for collecting the information of demontrations (Wang et al., 2023).

Input Text Encoding is Enhanced by Demonstrations. We investigate the influence of contextual information on input text encoding by repeating the experiments with different k. As shown in Fig. 3.3 (Middle), when the demonstrations increase, kernel alignment is enhanced, which is counterintuitive since longer preceding texts are more likely to confuse encoding targets. Such findings indicate that LMs (1) utilize contextual information to enhance the input text encoding and (2) correctly segment different demonstrations (detailed operation discussed in Appendix A.3).

Perplexed Texts are Encoded Worse. We investigate the correlation between kernel alignment and the perplexity of encoding targets with various k. Fig. 3.3 (Right, upper) shows a negative correlation for $k = 0$, that is, LMs generate poorer encodings for more perplexed input text when no demonstrations are given, which can be identified as an **I**n-**w**eight **L**earning (IWL) property of the inner text encoding. While, when demonstrations are given in context (Fig. 3.3 (Right, lower)), the negative correlation disappears, which suggests that LMs effectively encode more complex samples with the help of demonstrations in context. More discussion about the correlation between classification performance and perplexity is in Appendix A.6.

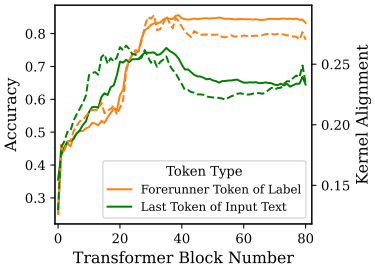


Figure 3.2: Test results of centroid classifier trained on ICL hidden states. **Solid**: Centroid classification accuracy, **Dotted**: Kernel alignment.

The above findings suggest that the inner text encoding is a hybrid process of ICL and IWL: Basic encoding capability presents from LMs weights, and is enhanced by demonstrations in context, which can be a clue to how demonstrations help ICL. Moreover, we are about to illustrate that these encodings are sufficiently informative for ICL tasks and linearly separable, which meets the presumption of simplified models on the linear and well-embedded input features.

3.3.2 Input Text Encoding is Linear and Task-Relevant but Position-biased

Input Text Encoding is Linear Separable and Task-relevant. We train a centroid classifier on hold-out 256 input samples (Cho et al., 2025c), using the hidden states of a specific token in $[x_t][s_t]$[6] from each layer and then predict the label y_t (see Appendix A.1.3 for details). The results are shown in Fig. 3.2, where considerably high classification accuracy of the forerunner token suggests the high linear separabilities of the hidden states in the task-semantic-relevant subspaces since the centroid classifier is linear. In addition, a similar emerging trend in accuracy and kernel alignment confirms the reliability of the kernel alignment measurement.

Input Text Encoding is Biased towards Position. Ideally, the inner representations of similar queries should

[5]Experiments of Fig. 3.3 (Middle) on Llama 3-70B do not involve results on AGNews.

[6]We skip the experiments on label tokens because of the leakage of ground-truth label information.

be highly similar regardless of their position in ICL inputs to support attention-based operations for classification. To verify this, for each encoding target, we extract the hidden states of forerunner tokens with various numbers of preceding demonstrations, and then calculate the cosine similarity between all possible pairs of the hidden states for the same target or different targets. As shown in Fig. 3.4, although the overall similarities on the same target are higher than on the different targets, they are both especially higher when their positions are close to each other.

As to be discussed in §3.5.3, such positional similarity bias may lead to one flaw: Demonstrations closer to the query have stronger impacts on ICL (Zhao et al., 2021; Lu et al., 2022; Chang & Jia, 2023; Guo et al., 2024). The principle of such bias is discussed in Appendix A.3.

3.4 Induction Circuits in Large Language Models

This section mainly shows how LMs utilize the encoded linear text representations in induction circuits with a typical 2-step form (Singh et al., 2024b): **Forerunner Token Heads** merge the demonstration text representations in the forerunner tokens into their corresponding label tokens with selectivity regarding the compatibility of demonstrations and label semantics. **Induction Heads** copy the information in the label representations similar to the query representation back to the query, which is conducted on task-specific subspaces, enabling LMs to solve multiple tasks by multiplexing hidden spaces.

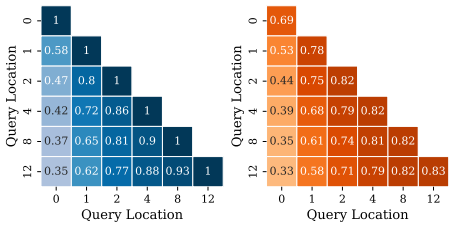


Figure 3.4: The similarities of ICL hidden states in different positions on layer 24 between **Left**: the same queries, **Right**: two different queries (on SST-2).

3.4.1 Step 2, Forerunner Token Head: Copy from Text Feature to Label Token

This subsection mainly examines and measures the forerunner token heads, which copy the information in the forerunner token into label tokens. We focus on how the representations in these tokens are merged, especially when the semantics of demonstrations and corresponding labels are disjoint, to explain the robustness of ICL against wrong labels present in the demonstrations.

Text Representations are Copied to Label Tokens. To confirm the existence of the representation copy process, we start by calculating the kernel alignment between the hidden state of forerunner token $[s_t]$ at layer l (the copy source) and that of label token $[y_t]$ at layer $(l+1)$ (the copy target). To suppress the high background values caused by the semantics of labels (refer to Appendix A.1.2.1), we use abstract label tokens $\{$"A", "B", "C", ...$\}$ instead of the original label tokens. The results are shown in Fig. 3.5 (Left), where the kernel alignment between the hidden states of the label token and the forerunner token gradually increases and then bumps up after the encoding magnitude (described in §3.3) in the forerunner token finished ascending. Such a phenomenon indicates that the hidden states of the input text representation encoded in the forerunner tokens are merged into their label tokens, suggesting the existence of copy processing from the forerunner token to the label token.

Text Representations are Copied without Selectivity. For each attention head, we extract the attention score[7] $\alpha_{s_t \to y_t}$ from the forerunner token $[s_t]$ (as attention key) to the corresponding label token $[y_t]$ (as attention query). We then mark the head with $\alpha_{s_t \to y_t} \geqslant 5/n_t$ (n_t: the length of tokens before $[y_t]$) as a Forerunner Token Head and count them in each layer. The results are shown in Fig. 3.5 (Middle, "Correct Label"), where the peak matches the copy period in Fig. 3.5 (Left). Moreover, to investigate the influence of the correctness of label tokens, we replace $[y_t]$ with a wrong label token[8], where the results in Fig. 3.5 (Middle, "Wrong Label") are almost identical to the correct-label setting, suggesting that the forerunner token heads don't show selectivity toward the semantic consistency between input text and labels, and simply merge the input text representations into the label tokens. Furthermore, we find (Appendix A.4) that the copy processing is *inherent*: During the copy processing, the preceding tokens are copied to the subsequent tokens, regardless of whether these copied tokens are forerunner tokens, which indicates that such copy processing is a universal inference behavior established during pre-training, rather than being evoked by the special tokens in the in-context learning (ICL) input, while still aiding ICL process (discussed in §3.5.1).

[7] This chapter use notation $\alpha_{K \to Q}$ to denote attention score with attention query Q and attention key K, which is same with the flow of information.

[8] For example, for a label space of "positive" and "negative", if $[y_t]$ is "positive", we replace it to "negative".

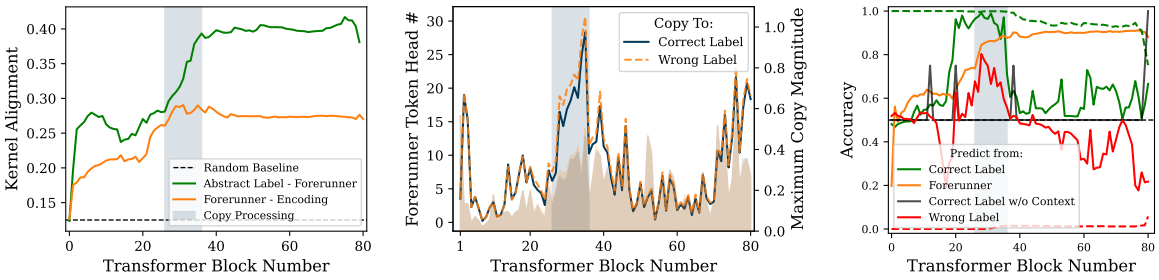


Figure 3.5: Hidden states copy magnitude from forerunner tokens to label tokens against layers. **Left**: Kernel alignment between the forerunner token (the copy source) and the abstract label token of the next layer (the copy target). **Middle**: **Curves**: The count of marked forerunner token heads with correct and wrong labels; **Colored Areas**: The maximum attention scores from forerunner token to query (*copy magnitude*) with correct and wrong labels (detailed attention head statistical data is in Appendix A.8.1). **Right**: Centroid classifier results predicted on the hidden states of correct and wrong *label tokens*, on SST-2 and MR. **Solid**: Predicted by classifiers $\mathcal{C}_s$ trained on hidden states of *forerunner tokens*. **Dotted**: Predicted by classifiers $\mathcal{C}_y$ trained on hidden states of *label tokens*.

Hidden States of Label Tokens are Joint Representations of Text Encodings and Label Semantics. Given the findings above, we probe the content of hidden states of label tokens $[y_t]$, i.e., how the copied text representation interacts with the original label semantics. We first train two centroid classifiers to predict the corresponding label y_t: (1) $\mathcal{C}_s$ trained on the hidden states of *forerunner tokens* $[s_t]$ and (2) $\mathcal{C}_y$ trained on the hidden states of *label tokens* $[y_t]$. To check whether the label tokens include the information of forerunner tokens, we use $\mathcal{C}_s$ to predict the label on the hidden state of label token $[y_t]$ in Fig. 3.5 (Right, solid). It shows that, during the copy processing, high classification accuracies can be achieved both on the correct label tokens and wrong label tokens, suggesting that the text features in the forerunner tokens can be partly and linearly detected in the label tokens. Moreover, results using $\mathcal{C}_y$ (dotted line) show extreme results, suggesting the label information remains in the label token. So, we can conclude that: Hidden states of label tokens are joint representations of label semantics and text representations. Moreover, interestingly, the accuracies from $\mathcal{C}_s$ shown in Fig. 3.5 (Right) decline after layer 35, suggesting that the information sharing between the forerunner tokens and the label tokens ends in later layers, which aligns with the results in Fig. 3.5 (Middle).

Label Denoising is Conducted on the Overlap of Label Semantics and Text Representations. Notice that in Fig. 3.5 (Right, solid), compared to the typical results predicted on the forerunner tokens, accuracies are improved on the correct label and suppressed on the wrong label, which suggests that information consistent with the label semantics is easier to be enhanced by the label tokens and vice versa, showing a feature selectivity on the consistency between the text representations and the label semantics. Given the observation that the information on label semantics and text features can be extracted separately and linearly, we can confirm that these two kinds of information are located in different sub-spaces of the hidden states, and *linearly* merged by the attention operation of forerunner token heads. Moreover, given the fact that there is no selectivity is observed in the copy behavior of the forerunner token heads (Fig. 3.5 (Middle)), it is intuitive that the feature selectivity shown in Fig. 3.5 (Right) comes from the arithmetical interaction of feature vectors on the *overlap* of sub-spaces between the label semantics and text features, making ICL stable against label noise (Min et al., 2022c). Moreover, as mentioned by Wei et al. (2023b), large models show poorer stability against label noise, and we infer that the larger hidden dimensions in larger models lower the overlap between the sub-spaces of label semantics and text representations to reduce the interaction.

3.4.2 Step 3, Induction Head: Feature Retrieval on Task Subspace

This subsection examines the existence of the induction heads, which retrieve label token features similar to the queries' forerunner token feature, and copy the retrieved features back to the query. We claim the necessity of multi-head attention in this process: Correct feature retrieval can only be conducted on the subspace of the hidden space, which is captured by some attention heads.

Induction is Correct in Minority Subspaces. Similar to Fig. 3.5 (Middle), we mark (1) the attention heads with the sum of attention scores from all the label tokens in the demonstrations $[y_1], \ldots, [y_k]$ (as attention keys) to the query's forerunner token $[s_q]$ (as attention query) of more than $5k/n_t$ as *Induction Heads*, and

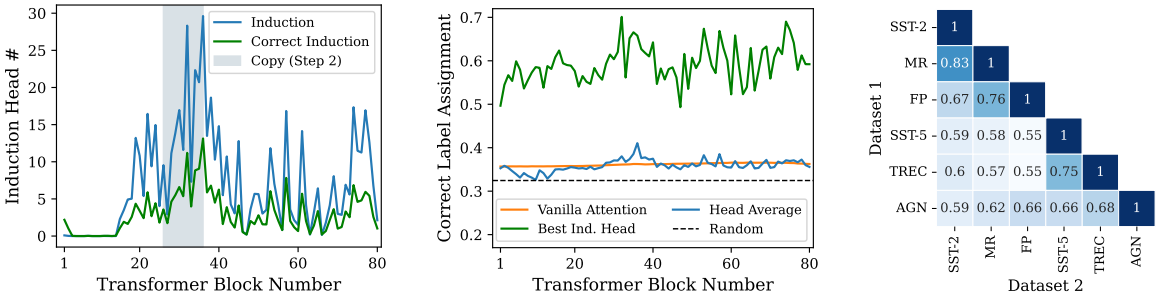


Figure 3.6: Measurements for induction heads. **Left**: The count of marked induction heads and correct induction heads against layers. **Middle**: The correctness of attention assignment (the sum of attention scores from query's forerunner token towards correct label tokens normalized by attention scores towards all the label tokens) extracted from the following attention calculations: **Vanilla Attention:** attention scores directly calculated on full dimensionality. **Head Average:** the averaged attention scores among all the heads. **Best Ind. Head:** the scores of attention head with the most correctness. **Right**: The correct induction heads overlap of all dataset pairs.

(2) attention heads with the sum of scores from all the correct label tokens more than $5k/|\mathbb{Y}|n_t$ as *Correct Induction Heads* ($\mathbb{Y}$ is the label space). We show the number of both kinds of induction heads in Fig. 3.6 (Left, detailed head statistics in Appendix A.8.1), where an unimodal pattern is observed later than the copy processing of Step 2. Moreover, more than half of the induction heads are not correct ones, suggesting that task-specific feature similarity can only be caught on some *induction subspaces* (defined by low-rank transition matrix $W_Q^{h\top}W_K^h$ of correct induction head h). We enhance this claim in Fig. 3.6 (Middle) (details in Appendix A.1.4), where both vanilla attention (without transformation and head split) and attention scores averaged among all heads show low assignment on correct label tokens, while some heads show considerable correctness. Considering the average value, the majority of attention heads almost randomly copy label token information to the query, causing the final prediction biased to the frequency of labels present in the demonstrations (Zhao et al., 2021). As the reason, we infer that the hidden states are sufficient (Fig. 3.2) but not minimum for ICL, where redundant information interferes with the operation of induction heads.

Some Induction Subspaces are Task-specific. We check if different tasks share the same induction subspaces based on the overlap of the correct induction heads across different datasets. Given $n_{\mathcal{D}}(h)$ as the number of times h is marked as correct induction head on dataset $\mathcal{D}$, the overlap rate S is defined as:

$$S(\mathcal{D}_1, \mathcal{D}_2) = \frac{2\sum_{\forall h} \min[n_{\mathcal{D}_1}(h), n_{\mathcal{D}_2}(h)]}{\sum_{\forall h} n_{\mathcal{D}_1}(h) + n_{\mathcal{D}_2}(h)}.$$

The results are shown in Fig. 3.6 (Right), where: (1) A significant overlap of induction heads indicates that a part of correct induction heads is *inherent* in the model, built by the pre-training process (Reddy, 2024; Singh et al., 2024b). (2) Such overlap is not fully observed, suggesting that some induction subspaces are *task-specific*: Input texts evoke task-specific induction heads, enabling the anisotropy multiplex of different subspaces in the hidden spaces to transmit relevant information for various tasks. Therefore, we can analogize ICL as an implicit end-to-end multi-task learning with hidden state multiplexing since multi-task learning also utilizes various task heads on common bottom network layers and informative hidden states (Zhang & Yang, 2021).

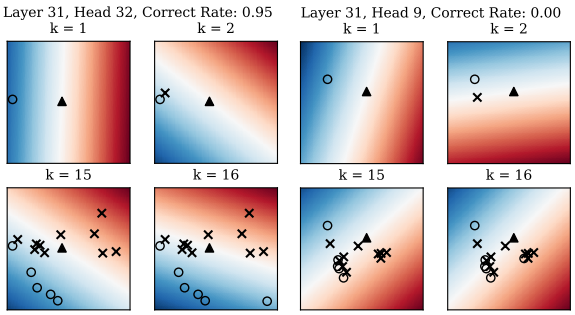


Figure 3.7: Label representations of k demonstrations visualized on **Left 4**: correct and **Right 4**: wrong induction head, on one sample of SST-2 (see Appendix A.8.3). ○: "positive" label, ×: "negative" label, ▲: zero vector. Color: Radial density of each categories, to estimate the attention assigned to query, negative to positive (cartography: Appendix A.1.5).

Demonstrations Saturate on Induction Subspace. We visualize the demonstrations' label token representations mapped on the induction subspaces by the transition matrix $W_Q^{h\top}W_K^h$ and principal component analysis in Fig. 3.7, indicating: (1) Compared to correct induction heads (Left 4), wrong induction heads (Right 4) are easier to map label representations linear-inseparably. (2) In the early stage of demonstration

feeding ($k = 1 \rightarrow 2$), when a new demonstration is given, the morphology of attention assignment towards query updates significantly (shown as background color in Fig. 3.7, estimated by the radial density of various labels; see §A.1.5 for details), while in the late stage ($k = 15 \rightarrow 16$), attention assignment morphology is stable. This can explain the demonstration saturation (Agarwal et al., 2024; Bertsch et al., 2024): ICL performance is submodular against the number of demonstrations. Intuitively, since demonstrations follow a prior distribution, representation of a new demonstration is likely to be located within the closure of existing demonstrations, making it less diverse to contribute to the attention assignment in the induction subspace.

3.5 Putting Things Together

So far, we have revealed the existence of the circuit with 3 steps, organized by the sequential inference process among Transformer layers. In this section, we find that the circuit is dominant in the ICL inference, while some bypass mechanisms activated by residual connection assist ICL inference. Moreover, a series of phenomena observed in ICL is successfully explained by the circuit.

3.5.1 Ablation Analysis

To (1) examine the causality between the attention connection specified by the 3 steps and ICL inference, and (2) demonstrate that our 3-phase circuit dominates or at least participates in ICL process, we disconnect the related attention connection of each step in the proposed circuit (see Appendix A.1.7 for details), and test the accuracies without such connections, as shown in Table 3.1. The results indicate that, compared to the **controlled results**, where trivial connections are removed, the accuracy of ICL significantly decreases when the non-trivial connections designated by the proposed circuit are ablated. This supports the existence of our circuit. However, the result doesn't fully match expectations, for example, the result without induction (line 5) should be consistent with the zero-shot inference result (line 6), since all the expected communication from demonstration to query is intercepted, but the real result is better; and the contribution of Step 1 **in later layers** are unexpectedly high, indicating the existence of some bypass mechanisms parallelly contributing to ICL accuracies.

Table 3.1: Accuracy variation (%) with each inference step ablated on Llama 3-8B. Small numbers are **controlled results** (mean ± std) of randomly ablating equivalent amounts of connections. Ablations are applied from the bottom to the top layers, and results with various ablated layers are reported (detailed settings: Appendix A.1.7).

#	**Attention Disconnected** Key → Query	**Affected Layers Ratio** (from layer 1) 25%	50%	75%	100%
1	**None** (4-shot baseline)	±0 (Acc. 68.55)			
	– Step1: Input Text Encode –				
2	**Demo. Texts** x_i → **Forerunner** s_i	−4.98 (−0.89 ± 0.00)	−15.82 (−1.19 ± 0.02)	−23.43 (−3.29 ± 1.87)	−30.60 (−1.61 ± 0.01)
3	**Query Texts** x_q → **Forerunner** s_q	−13.87 (−0.16 ± 0.00)	−21.10 (−0.08 ± 0.00)	−24.74 (−0.47 ± 0.04)	−28.38 (−0.55 ± 0.00)
	– Step2: Semantics Merge –				
4	**Demo. Forerunner** s_i → **Label** y_i	−2.24 (−0.00 ± 0.00)	−3.45 (−0.18 ± 0.00)	−3.39 (−0.10 ± 0.04)	−3.42 (−0.18 ± 0.01)
	– Step3: Feature Retrieval & Copy –				
5	**Label** y_i → **Query Forerunner** s_q	−5.14 (+0.03 ± 0.00)	−10.03 (−0.08 ± 0.00)	−11.36 (+0.00 ± 0.00)	−10.22 (−0.06 ± 0.00)
	Reference Value				
6	**Zero-shot**	−17.90 (Acc. 50.65)			
7	**Random Prediction**	−36.05 (Acc. 32.50)			

3.5.2 Bypass Mechanism

Motivated by the ablation results, we believe that several mechanisms including our circuit run parallelly for ICL, since the residual connection supports complex paths among layers and attention heads. We list some possible bypasses and plan a complete enumeration as future work.

Parallel Circuits. Multiple 3-step circuits can execute in parallel, and one layer can assign multiple inference functions to different heads, causing dispersion and deserialization as shown in Fig. 3.8, where a narrow (fewer heads per layer) and deep (more layers) model is more likely to generate localized inference, and vice versa.

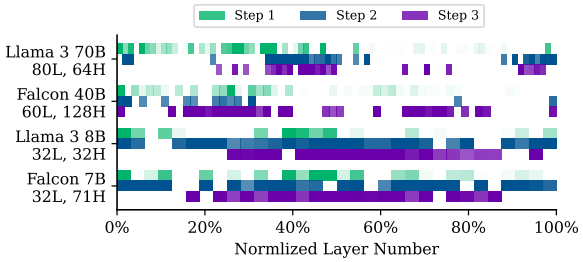


Figure 3.8: Dynamics and deserialization of magnitudes of proposed 3 inference steps (cartography details: Appendix A.1.6).

Direct Decoding. Residual connection to the LM heads allows intermediate hidden states to be decoded directly. Intuitively, a shortcut from Step 1 encodings to the LM head enables ICL with zero-shot capacities, since we have confirmed that encoded representations are informative for ICL tasks (§3.3.2), while the decoding methods should be selected carefully (Cho et al., 2025c) (e.g. with essential calibration). On the other hand, shortcuts from insufficiently encoded features may lead to meaningless information decoded by language model heads, causing prediction bias, i.e., even if no query is given, ICL still returns unbalanced results (Zhao et al., 2021) decoded from tokens of the prompt template (see Appendix A.5 for details).

Shortcut Induction. Note that a k-shot ICL input sequence always contains a $(k-1)$-shot sequence, where the k-th forerunner token (served as the query of $(k-1)$-shot sequence) is previously processed by the $(k-1)$-shot inference. So every forerunner can directly retrieve previously processed forerunner tokens in the demonstrations to copy their induction results directly. The non-trivial result (Table 3.2) with all forerunner tokens disconnected from each other confirms such inference.

Table 3.2: Accuracy drop with shortcut ablated.

Attention Disconnected ($\forall i$)	25%	50%	75%	100%
Forerunner $s_{1:i}$ → **Forerunner** $s_{i:q}$	$-1.30_{-0.00}$	$-0.75_{-0.16}$	$-0.78_{-0.16}$	$-1.56_{-0.71}$

3.5.3 Explanation towards Observed ICL Phenomena

Difficulty-based Demonstration Selection. In §3.3.1, we find that in the zero-shot scenario, perplexed texts are harder to encode, explaining the observation of PPL-ICL (Gonen et al., 2023): Selecting demonstrations with lower perplexity can improve ICL performance. Moreover, while the demonstrations increase, LMs can encode more complex inputs with diverse information to update the attention assignment shown in Fig. 3.7, making it beneficial to input harder demonstrations later, which explains the ICCL (Liu et al., 2024), which builds a demonstration sequence from easy to hard.

Prediction Bias. (1) **Contextual Bias**: As mentioned in §3.5.2 and shown in Appendix A.5, direct decoding insufficiently encoded information adds meaningless logits into LM's output, causing a background prediction value even if no queries are given (named *bias*). (2) **Positional Bias**: As shown in §3.3.2, closer demonstrations are encoded more similarly, so label tokens near the query have more similar information to the query, causing more attention assignment in the induction processing, so that more influences on the prediction. (3) **Frequency Bias**: As shown in §3.4.2, in the induction, some attention heads are without correct selectivity towards labels, causing an averaged copy processing from label tokens to the query, triggering a prediction bias towards the label frequency in the demonstration, even if their contribution (absolute value of attention score on label tokens) is small. All three biases are observed by Zhao et al. (2021), and can be removed by ICL calibration methods.

The Roles and Saturates of Demonstrations. It is well known that demonstrations improve the performance of ICL. We decompose such performance improvement into 2 parts: (1) demonstrations help early layers encode better (§3.3.1), and (2) more demonstrations provide larger label token closure, enabling more accurate attention assignment (§3.4.2), while the volume of such closure is submodular to demonstrations, causing the saturates of ICL performance towards demonstrations.

The Effect of Wrong Label. It is well-known that the label noise is less harmful in ICL (Min et al., 2022c) than in gradient-based learning (Zhang et al., 2021). We have explained in §3.4.1 that ICL implies labels denoise to stabilize ICL against label noise, while weakened by dimensionality.

3.6 Conclusion and Discussion

Conclusion. In summary, this chapter restores ICL inference into 3 basic operations and confirms their existence. Fine-grained measurements are conducted to capture and explain various phenomena successfully. Moreover, ablation studies show the proposed inference circuit dominates and reveals the existence of bypass mechanisms. We hope this chapter can bring new insight into ICL practice.

The Role of Early and Later Layers. Our framework shows: The encoding result of Step 1 can be directly used for classification with reliable decoding, and later transformer layers are not contributing to centroid classification accuracies (Fig. 3.2), leading to a taxonomy of *Input Encoding* for Step 1 and *Output Preparation* for Step 2 and 3: LMs complete multi-task classification implicitly in early layers, and verbalize it by merging task-specific label semantics in later layers. Therefore, we suggest an early-exiting inference: removing some top layers and using a centroid classifier (Cho et al., 2025c) to accelerate ICL as shown in Table 3.3 and Appendix A.5.

Table 3.3: Performance of full and layer-pruned ICL inference.

Inference	Acc.	# Param.	Speed
Full + LM Head	66.19%	70.6B	1×
Full + Centroid	83.24%	69.5B	1.00×
Layer34 + LM Head	49.29%	32.7B	2.16×
Layer34 + Centroid	**84.27%**	**31.2B**	**2.38**×

Pre-training Possibility from Natural Language Data. A large gap can be considered between such a delicate circuit and gradient descent pre-training on the wild data. However, we believe the wild training target contains the ICL circuit *functionally*: Based on the previous works finding trainability of ICL on simplified linear representation-label pairs (Chan et al., 2022; Reddy, 2024; Singh et al., 2024b), we speculate that in early training step, Transformers learn to extract linear representations shown in §3.3 from wild data (detailed in Appendix A.2), serving as the training input of later layers to evoke the emergence of induction heads with the same mechanism shown in aforementioned previous works. Moreover, our conclusion of Step 3 highlights the input data requirements for the later layers: These data should activate the multiplex of hidden space, i.e., it should implicate multi-task classification with a wide distribution, which is consistent with the aforementioned previous works.

Comparison with Previous Works. Several prior studies have sought to interpret ICL as known processes, including implicit gradient descent (Dai et al., 2023), kernel regression (Han et al., 2023a), and implicit Bayesian inference (Xie et al., 2022), etc. However, as stated in §3.2.1, these approaches fall short of fully explaining the phenomena observed during the ICL inference process. For instance, gradient descent is known to be fragile against label noise (Zhang et al., 2021), leading to a misalignment when analogies are drawn to ICL, which is robust against label noise (Min et al., 2022c; Liu et al., 2020). Similarly, attempts to explain ICL as kernel regression fail to account for positional bias, making them disconnected from empirical studies that have observed various inference phenomena in real-world LMs. Also, other works have employed induction circuits to explain ICL dynamics (Wang et al., 2023; Elhage et al., 2021; Olsson et al., 2022), yet significant gaps remain in aligning these explanations with empirical observations. Our work is the first to unify the fragmented conclusions from prior empirical studies (discussed in §3.5.3) through detailed experimental measurements. By demonstrating the alignment of these observations, we emphasize the novelty and primary contribution of our work.

Limitations & Open-questions. (1) These 3 basic operations are not functionally indivisible. Ideally, mechanistic interpretability aims to reduce every operation in ICL inference to the interconnection of special attention heads to ultimately examine how the operating subspaces interact between steps, and reconstruct ICL behavior from a minimal set of attention heads. Also, although we show the significance of the inference circuit in the ablation analysis (§3.5.1), measuring the connectivity of these attention heads can also be beneficial to get more insights into the circuit. (2) The conclusions may not align with scenarios where ground-truth labels are not provided in the context in some cases, which are often referred to as in-weight learning, where significant differences or even antagonism with standard ICL have been highlighted by previous works[9] (Chan et al., 2022; Reddy, 2024), reasonably and necessarily warranting separate discussion (discussed in Appendix A.7). This chapter explains the inference behavior of the model under ICL conditions, leaving the in-weight learning scenario for future work. (3) We only focus on classification tasks, while we believe that our findings can be applied to non-classification tasks, efforts are still needed to fill the gap.

After-review: What Has Our Work Inspired for Future Research? As published at the highly prestigious international top-tier conferences, this series of works has already provided significant inspiration for subsequent studies. For example, built on works in this chapter, (Bakalova et al., 2025) generalizes our framework and attempts to address the issue of unseen labels (i.e., scenarios where ground-truth labels are not provided in the context). Also, some of our works (Yang et al., 2025b;c) focus on this point, and find that: the induction heads are able to not only copy the label literal, but also help to recognize the task, and induce the task effect to the outputs. Other later related works but not directly inspired by this chapter have been reviewed faithfully in Chapter 2.2.

[9] In these works, toy models even struggle to work on both inputs simultaneously, so that it is natural to consider different inference mechanism for the so-called in-weight learning.

Chapter 4

Supplementary Mechanism Basis for ICL in Unseen Label Scenario

Mechanism of Task-oriented Information Removal in In-context Learning

This chapter mainly retells the paper "Mechanism of Task-oriented Information Removal in In-context Learning" by Yufeng Zhao (alias: Hakaze Cho) (refer to Publication List). Motivated by the defect of the aforementioned induction circuit-based ICL interpretation, this chapter proposes a novel ICL mechanism from the perspective of information removal, as a major *bypass* mentioned in Chapter 3.

Abstract

In-**c**ontext **L**earning (ICL) is an emerging few-shot learning paradigm based on modern **L**anguage **M**odels (LMs), yet its inner mechanism remains unclear. In this chapter, we investigate the mechanism through a novel perspective of information removal. Specifically, we demonstrate that in the zero-shot scenario, LMs encode queries into non-selective representations in hidden states containing information for all possible tasks, leading to arbitrary outputs without focusing on the intended task, resulting in near-zero accuracy. Meanwhile, we find that selectively removing specific information from hidden states by a low-rank filter effectively steers LMs toward the intended task. Building on these findings, by measuring the hidden states on carefully designed metrics, we observe that few-shot ICL effectively simulates such task-oriented information removal processes, selectively removing the redundant information from entangled non-selective representations, and improving the output based on the demonstrations, which constitutes a key mechanism underlying ICL. Moreover, we identify essential attention heads inducing the removal operation, termed Denoising Heads, which enables the ablation experiments blocking the information removal operation from the inference, where the ICL accuracy significantly degrades, especially when the correct label is absent from the few-shot demonstrations, confirming both the critical role of the information removal mechanism and denoising heads.

4.1 Introduction

In-**c**ontext **L**earning (ICL) is a promising application of modern **L**anguage **M**odels (LMs), in which a sequence of input-label pairs (demonstrations) concatenated with a query is fed to the LMs to predict the query label. However, the inner mechanism of ICL remains unclear, despite some progress (Zhou et al., 2024b), including: linking ICL to specific input components (Min et al., 2022c; Yoo et al., 2022; Pan et al., 2023; Kossen et al., 2024), relating ICL-style inputs to pre-training data (Li & Qiu, 2023; Gu et al., 2023b; Han et al., 2023b; Li et al., 2024; Cho et al., 2025b), or analogy to simpler algorithms (Zhang et al., 2023; Dai et al., 2023; Xie et al., 2022; Han et al., 2023a).

Among prior work, mechanistic interpretability studies (Elhage et al., 2021; Chan et al., 2022; Reddy, 2024; Wang et al., 2023; Singh et al., 2024b; Cho et al., 2025a; Yang et al., 2025c; Minegishi et al., 2025; Yin & Steinhardt, 2025; Bakalova et al., 2025) reduce ICL to functional LM components (e.g., attention heads), capturing many phenomena and establishing causal roots to ICL accuracy. Mainstream interpretations attribute ICL to Induction Heads (Elhage et al., 2021; Reddy, 2024; Cho et al., 2025a; Yang et al., 2025c) (introduced in §4.2), which copy the most relevant label token with the query's hidden states from demonstrations to the outputs. However, such explanations fail in *unseen label scenarios* as shown in Fig. B.1, i.e., the correct label is absent in the demonstrations (Minegishi et al., 2025; Cho et al., 2025a) where the ICL accuracy is damaged

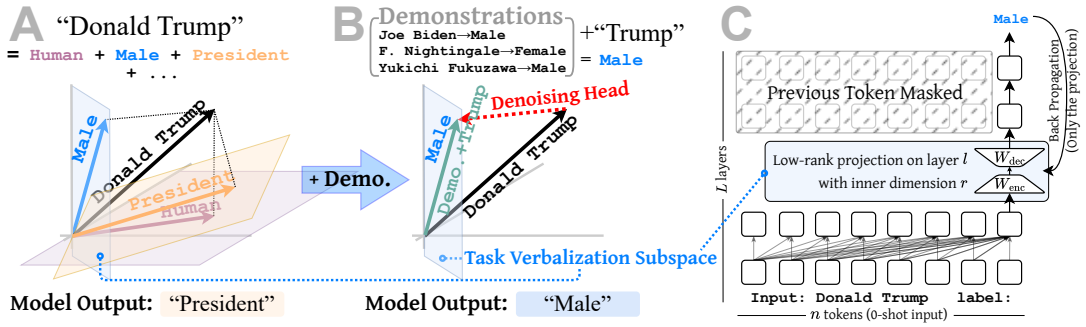


Figure 4.1: **(A)** A zero-shot query is encoded into a non-selective semantic representation containing all possible label information on various subspaces, making the output arbitrary among these labels. **(B)** Demonstrations help LM filter the label information, saving only the task-related one on the specific subspace (termed **T**ask-**V**erbalization **S**ubspace (TVS)), leading to the task-specific output. **(C)** We explicitly find the TVS by injecting a low-rank filter into the residual stream of zero-shot inputs, and train only the filter to drive the final outputs towards the ground-truth labels.

but well beyond the zero-shot level (refer to Table B.4), suggesting additional mechanisms supporting such accuracy.

Therefore, we propose a new perspective to interpret ICL, as shown in Fig. 4.1. Specifically, we argue that ICL should **not** be viewed as "copying new information to the output (Cho et al., 2025a) or learning new tasks (Li et al., 2024)", but as *removing task-irrelevant information from the query to highlight the specified task*. In detail, since the previous works demonstrate that various kinds of information are encoded in various subspaces in LM's hidden states (Saglam et al., 2025; Zhao et al., 2025), we hypothesize that: LMs encode zero-shot queries with superposed information from all possible subspaces that mix task-specific and task-irrelevant information, often causing zero-shot outputs to yield plausible but arbitrary and unexpected answers (e.g., "Donald Trump, label:" $\rightarrow$ "President" instead of "Male", illustrated in Fig. 4.1 and demonstrated in Fig. 4.2). Meanwhile, task-specific information is concentrated in one of these low-rank spaces (termed *Task-Verbalization Subspace* (TVS)) in the hidden states, which leads LM to output in the expected pattern but covered by the noisy information in the zero-shot scenario. To validate this, as shown in Fig. 4.1 (C), we inject a low-rank filter into the last token's residual stream of zero-shot inputs, and train the filter with the ground-truth output to recover the TVS. Surprisingly, filters preserving only 0.7% of dimensions boost accuracy from near-zero to a remarkable level, showing that removing the TVS-orthogonal irrelevant information suppresses arbitrary outputs and enforces task-specific predictions.

Based on such findings, we generalize the task-oriented information removal from the injected zero-shot settings to vanilla (i.e., un-injected) few-shot settings. As shown in Fig. 4.1 (B), we observe that LMs, guided by few-shot demonstrations, implicitly drive the hidden states towards TVS calculated independently in the previous low-rank filter injection experiments to produce task-specific outputs. Moreover, we also confirm a consistent behavior in the aforementioned unseen label scenario. Furthermore, we identify a set of attention heads (termed *Denoising Head* in this chapter), which are highly independent from the input labels and induction heads, conducting such task-oriented information removal by re-encoding the queries' information. Such head-localization enables ablation experiments by zeroing these heads' outputs, which results in substantial accuracy degradation, especially in unseen label scenarios, where the accuracy drops nearly to zero, strongly indicating the task-oriented information removal as an essential mechanism for ICL.

In summary, the main contributions of this chapter are: (1, §4.3) We propose a novel and systematic evaluation framework to measure the task-oriented information removal dynamics in hidden states, which is a versatile methodology for not only the ICL scenario. **(2, §4.4.2, 4.4.3)** Based on the proposed evaluation framework, we propose a novel ICL mechanism from the task-oriented information removal perspective, where demonstrations remove the task-irrelevant information from the query to drive LM's output on the specified task. **(3, §4.4.4)** We identify denoising heads responsible for task-oriented information removal, enabling subsequent ablation to verify their effectiveness. We further address induction heads' limitation on unseen labels by showing that ablating denoising heads (also the information removal) sharply reduces accuracy, and collapses to zero on unseen label scenarios.

4.2 Background

In-context Learning (ICL) (Radford et al., 2019). In an ICL scenario, given k input-label pairs $\{(x_i, y_i)\}_{i=1}^{k}$ as the *demonstration* and a x_q as the query, a concatenation $[x_1, y_1, \ldots, x_k, y_k, x_q]$ is fed into LMs for y_q corresponding to x_q. ICL enables LMs to perform diverse tasks without parameter updates, drawing significant research interest, particularly in mechanism analysis. However, current studies, especially those focusing on attention components such as Induction Heads (Elhage et al., 2021; Wang et al., 2023; Reddy, 2024; Singh et al., 2024b; Cho et al., 2025a), cannot account for all ICL phenomena, leaving them an incomplete explanation of ICL behaviors as described below.

Induction Heads-based Mechanistic Interpretation for ICL. Induction Head (Elhage et al., 2021) is a family of attention heads identified in LM, primarily responsible for prefix recognition and suffix replication. Specifically, for an input string shaped as $[\mathrm{A}, \mathrm{B}, \mathrm{A}]$, induction heads add the information of B, which is the suffix of the previous A, to the residual stream of the last A, so that drive the output towards B. Such behavior aligns with the form of ICL, where the x_is, especially the consistent structural final token (i.e., ":" in "label:") (Cho et al., 2025a), serve as the As, and the y_is serve as the Bs. The induction heads retrieve the most similar x_is with x_q, and copy the corresponding y_i to the output. However, such a framework has a serious issue: it determines the final answer only by copying the tokens that have appeared in the context; thus, if the correct label token does not appear in the demonstrations (termed Unseen Label Scenario), then the framework predicts ICL to fail. But this is not the case in reality, although the accuracy in the unseen label scenario is significantly harmed, but basically not zero (refer to Appendix B.1.3), which prompts for new investigations to supplement the induction head framework.

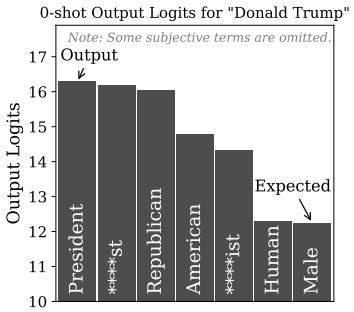


Figure 4.2: Logits distribution (selected) of zero-shot input "Donald Trump, label:".

4.3 Method: Tracing the Task-oriented Information Removal

In this section, we propose a systematic evaluation framework that traces the task-oriented information removal. Following this framework, in §4.4, we undertake a comprehensive investigation of task-oriented information removal in the ICL setting. While our discussion centers on the ICL scenario as the prototype, we believe the framework extends naturally to a wider range of scenarios.

Preliminary: Task-Verbalization Subspace (TVS). As shown in Fig. 4.1 (A), an LM encodes a zero-shot query x_q into a non-selective semantic representation $h_q^l \in \mathbb{R}^d$ after l transformer blocks, where the output derived from such h_q by the remaining layers is unlikely to be the desired answer to x_q, as demonstrated in Fig. 4.2. In such a scenario, we hypothesize the existence of a low-rank subspace in the representation space encoding the task-related information, equipped with a projection matrix $W \in \mathbb{R}^{d\times d}$ that projects h_q^l to remove task-irrelevant information, thereby redirecting the final output derived from $h_q^l W$ towards the task-specific answer. As shown in §4.4.2, such W defines the task and also the "verbalization mode" (i.e., how the output is expressed, such as "positive" vs. "+" in sentiment analysis), for which we term W as **T**ask-**V**erbalization **S**ubspace (TVS).

Step 1: Finding the Explicit TVS in Zero-shot Hidden States. As shown in Fig. 4.1 (C), we explicitly assume the existence of TVS parameterized as a low-rank filter $W_{\text{enc}}W_{\text{dec}} \in \mathbb{R}^{d\times r} \cdot \mathbb{R}^{r\times d}$, and verify this assumption by an injection method: we inject $W_{\text{enc}}W_{\text{dec}}$ into the residual stream of the inputs' last token (where the output is expected) in a specific layer l, and interrupt this last token's access to previous tokens in the subsequent layers to block the context-related additional information gain from the inputs afterward the filter to keep the attribution clear. Given some zero-shot prompts and ground-truth responses, with all other parts of the LM frozen, we train the $W_{\text{enc}}W_{\text{dec}}$ on zero-shot inputs towards corresponding ground-truth responses (refer to Appendix B.1.2 for details).

Step 2: Tracing the Implicit Task-oriented Information Removal towards TVS. We then assume that LMs intrinsically drive the hidden states towards TVS in the **uninjected** inference with the help of demonstrations. To validate this, we propose two carefully designed geometric metrics of the few-shot hidden states, utilizing the trained $W_{\text{enc}}W_{\text{dec}}$ in Step 1. In detail, given a N-amount set of hidden states $H^{l,k} = \{h_i^{l,k}\}_{i=1}^{N}$ from the last token of k-shot ICL inputs after l transformer blocks:

- **Eccentricity**: To measure the *magnitude* of information removal, we calculate the eccentricity as the

covariance loaded ratio on the first principal direction of $H^{l,k}$, indicating the enrichment, or anisotropy, of information on a single linear representation (Engels et al., 2025). A higher eccentricity indicates a purer representation with less extraneous information.

- **Covariance Flux on TVS**: To measure the *correctness* of information removal, we calculate the ratio of task-related information, we project all the $h_i^{l,k} \in H^{l,k}$ onto the $W_{\text{enc}}W_{\text{dec}}$ with $r = 8$ trained on layer l in Step 1, and calculate the covariance ratio of $H^{l,k}W_{\text{enc}}W_{\text{dec}}$ against $H^{l,k}$ (details in Appendix B.1.3). A higher covariance flux suggests that a larger proportion of the information in $H^{l,k}$ is task-related.

In summary, as intuitively shown in Fig. B.10, eccentricity quantifies the concentration of information into a single feature, thereby indicating the magnitude of information removal; and covariance flux evaluates whether the task-related information is preserved, i.e., the correctness of the information removal. Appendix B.7 relates these covariance metrics to entropy, a standard measure of information.

Step 3: Building the Causal Link between Information Removal and ICL Performance by Denoising Heads. To test whether the detected information removal process indeed has a causal significance, i.e., whether it actively influences accuracy, or merely constitutes a byproduct of accuracy improvement from other mechanisms, we conduct a causality evaluation which actively suppresses such mechanisms. Since the aforementioned two metrics are not directly controllable, we utilize the attention heads as the handle for such causal evaluation. In detail, we quantify each head's contribution to task-oriented information removal by ablating it and re-evaluating the metrics on the hidden states produced by the ablated layer. Then, we label the heads that contribute substantially to these metrics as "**D**enoising **H**eads (DH)", then ablate all the detected DHs and evaluate the ICL performance to verify the causal link. The design of the technical details relies in part on the conclusions from Step 2, as described in §4.4.4.

4.4 Task-oriented Information Removal in In-context Learning

In this section, we conduct comprehensive experiments to apply the aforementioned evaluation framework to trace the task-oriented information removal dynamics in the ICL scenario.

4.4.1 Experiment Settings

Models. We mainly conduct experiments on 6 modern LMs detailed in Appendix B.1: Llama (v3.2-1B, v3-8B, v3-13B Instruct) (Grattafiori et al., 2024); Qwen 2.5 (3B, 3B Instruct, 7B) (Team, 2024a; Yang et al., 2024). We default to reporting the results on Llama 3.2-1B (more results in Appendix B.9).

Datasets. We utilize **6 classification datasets**: SST-2, SST-5 (Socher et al., 2013), MR (Pang & Lee, 2005), FP (Malo et al., 2014), AGNews (Zhang et al., 2015), Subjective (Wang & Manning, 2012); and **3 non-classification datasets**: country-capitals (icyrockcom, 2011), people-profession (Laouenan et al., 2022), opus-100 (Zhang et al., 2020). Unless specified, we report the results on SST-2.

Others. To verify whether the LM spontaneously produces task-related outputs, we calculate the accuracy by greedy decoding, where hard matching of the model output among all the vocabularies is utilized for the accuracy, instead of the restricted decoding, where only the most likely output in the pruned output candidates is selected. Refer to Appendix B.1.1 for the prompt templates.

4.4.2 Step 1: Finding the Explicit TVS in Zero-shot Hidden States

In this section, following Step 1 in §4.3, we show that explicitly removing information from the orthogonal complement of the injected TVS in the zero-shot hidden states guides the LM to recognize the intended task rather than producing arbitrary outputs, suggesting zero-shot hidden states contain task-related information with redundant information interfering with the output, and filtering out such redundancy helps LMs focus on the target task.

Existence of Explicit TVS in Hidden States. We conduct the injection-and-train experiment mentioned in Step 1 of §4.3, with validation accuracy with various l and r on SST-2 and Llama 3.2-1B is shown in Fig. 4.3, where, globally, compared to zero-shot accuracy, filtering the hidden state while explicitly assuming the existence of a TVS of max-rank[10] 2 can significantly improve the output accuracy from the open-end

[10]Notice that even if the r is sufficient large, the effective rank of $W_{\text{enc}}W_{\text{dec}}$ can also be small, so the r is the maximum possibile rank. (Original embedding dimension is 2048.)

decoding (mentioned in §4.4.1). Such results confirm our hypothesis: selectively removing information from zero-shot hidden states steers outputs toward the task, inspiring a possibility that ICL implicitly applies this procedure, as verified in §4.4.3.

$W_{\text{enc}}W_{\text{dec}}$ is Information Removal. One suspicion is that $W_{\text{enc}}W_{\text{dec}}$ may not significantly remove information, such as when the principal component direction of h_q^ls is highly aligned with the eigenvector of $W_{\text{enc}}W_{\text{dec}}$. We rule out such doubt by: (1) Calculating the remaining covariance-load ratio out of the top-r principal components of h_q^ls, as a lower bound of information removal by $W_{\text{enc}}W_{\text{dec}}$ with maximum rank r (explained in Appendix B.2, and numerically visualized in Appendix B.9), as shown in the diamond size of Fig. 4.3, suggesting effective information removal by $W_{\text{enc}}W_{\text{dec}}$. (2) Calculating the cosine similarity of W_{enc}'s eigenvectors and the principal components of the hidden states, as an estimation of the positive gap between the actual information removal and the lower limit, shown in Appendix B.2. In summary, $W_{\text{enc}}W_{\text{dec}}$ is an effective information removal while enhancing the zero-shot accuracy.

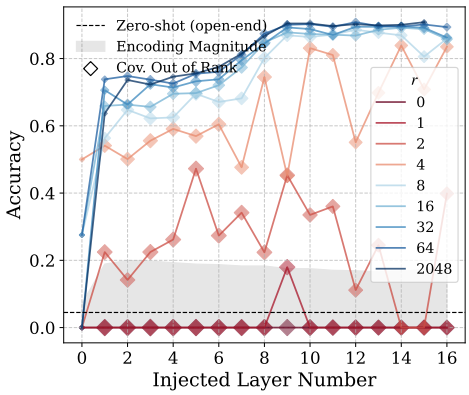


Figure 4.3: Filter-injection evaluations on various layers and ranks. **Marker size**: the remaining covariance out of the top-r principal components of the point cloud of h_q^l. Refer to Appendix B.9 for numerical results. **Encoding Magnitude**: the quality of the hidden states in the current layer as a sentence encoding of the query, emerging simultaneously with the test accuracy, calculated follows Cho et al. (2025b).

W_{dec} Controls the Verbalization Mode. Notice that, in the training of the $W_{\text{enc}}W_{\text{dec}}$, not only is the task-related information (as shown in Fig. 4.1 (B)), but also the verbalization mode, i.e., how the task-related information is transferred into output tokens (e.g., into "positive/negative" or "+/−") is defined in the trained parameters. This is also why we name $W_{\text{enc}}W_{\text{dec}}$ as "task-verbalization". Moreover, we find that the verbalization mode is mainly controlled by the W_{dec}: In detail, we first train the $W_{\text{enc}}W_{\text{dec}}$ filter on SST-2 using the standard verbalizations "positive" and "negative", then freeze either W_{enc} or W_{dec} and fine-tune the remaining one to transfer the outputs to symbolic verbalizations "A" and "B". The results, shown in Table 4.1, indicate that fine-tuning only W_{dec} successfully transfers the verbalization into symbolic form, whereas W_{enc} fails. This suggests that W_{enc} extracts task-specific but verbalization-irrelevant semantics from hidden states (otherwise, freezing it will not prevent successful verbalization transfer), while W_{dec} maps them into the target verbalization by aligning hidden states with the specified output unembedding vectors (Yang et al., 2025c), consistent with prior findings (Tao et al., 2024).

4.4.3 Step 2: Tracing the Implicit Task-oriented Information Removal

In §4.4.2, removing the TVS-orthogonal information produces task-specific output, which serves as efficient adaptors for LM to specific tasks, and motivates us to hypothesize the mechanism of ICL with few-shot demonstrations similar to such task-oriented information removal. In this section, following Step 2 in §4.3 to measure the two geometric metrics (i.e., Eccentricity and Covariance Flux) of the few-shot hidden states from uninjected inference processing, thereby verifying the hypothesis shown in Fig. 4.1 (B): LMs autonomously remove the redundant information out of the TVS with the help of few-shot demonstrations to produce task-specified ICL outputs.

Table 4.1: Fine-tuning results with parts of the filter frozen, on Llama 3.2-1B and SST-2.

Trained Part	Both	W_{enc}	W_{dec}
Accuracy	0.88	0.00	0.84

Language Models Intrinsically Compress Hidden States towards TVS. We test both the metrics on various k and l, with results as well as the output accuracy shown in Fig. 4.4 for Llama 3.2-1B and SST-2 (refer to Appendix B.9 for more cases). Globally, as the amount of demonstration (k) increases, the task-oriented information removal primarily occurs in the middle-to-late layers, causing an increasing eccentricity and covariance flux. In detail:

- **Eccentricity:** As shown in Fig. 4.4 (middle), with sufficient demonstration numbers (k), in the later layers, the eccentricity increases against k, reaching a maximum of around 60%, showing a highly anisotropic information compression with the help of demonstrations. Specifically, on certain datasets such as SST-2, the eccentricity exhibits a distinctive nonmonotonic *decrease-and-increase* pattern against k. This suggests that information removal initially compresses from the first principal component of zero-shot hidden states, which is understandable since the input distribution may be disaligned with the task,

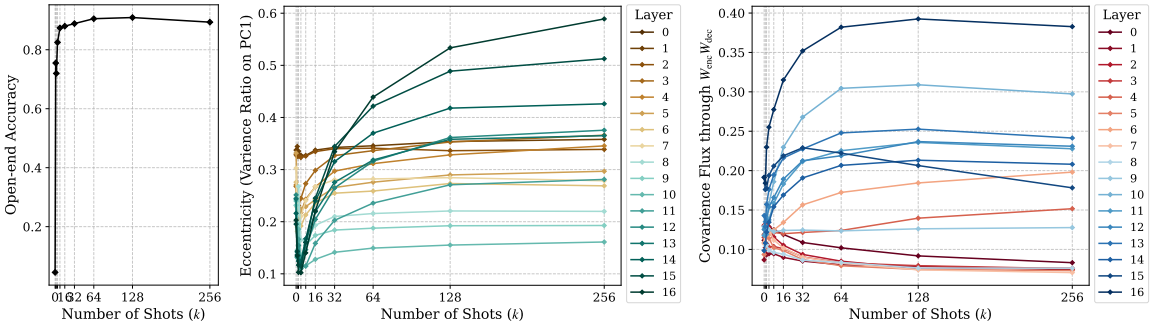


Figure 4.4: Test results of **(left)** open-end accuracy against demonstration numbers k, **(middle)** eccentricity of hidden states point cloud against demonstration numbers k on various layers, **(right)** covariance flux through the catched $W_{\text{enc}}W_{\text{dec}}$ against demonstration numbers k on various layers.

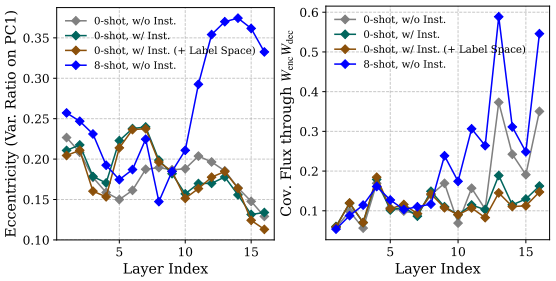


Figure 4.5: Eccentricity and covariance flux w.r.t. layer on MR, with instruction texts, demonstrations, or plain zero-shot inference.

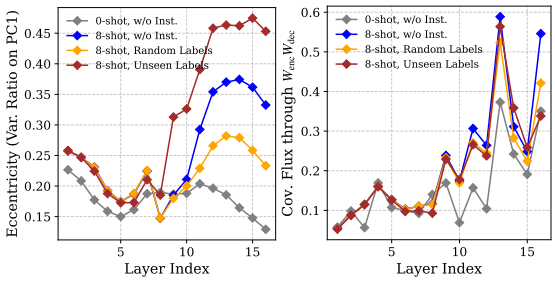


Figure 4.6: Eccentricity and covariance flux w.r.t. layer on MR, with random label, unseen label, and normal settings.

and confirms an important conclusion: task-oriented information removal primarily compresses from the task-irrelevant directions (Yang et al., 2025c; Kirsanov et al., 2025) of the hidden state point cloud, as discussed in Appendix B.3.

- **Covariance Flux:** As shown in Fig. 4.4 (right), also in the later layers, the covariance flux increases against k, suggesting that LMs correctly preserve the task-related information while removing the task-irrelevant one with the help of demonstrations. Moreover, the covariance flux of the later layers closely tracks the accuracy (Fig. 4.4 (left)), with both reaching their peak and then saturating at the same point, which indicates that the alignment of hidden states toward TVS constitutes a crucial mechanism shaping the inference output[11].

Additionally, such findings concur with previous works, where the early layers contribute to task-agnostic low-level semantic encodings (Jawahar et al., 2019; Chen et al., 2023; Wang et al., 2023; Cho et al., 2025a; Yang et al., 2025c), thus task-related behavior such as task-oriented information removal, can not be significantly observed in the early layers by our metrics. Conversely, in later layers, high-level features are being processed, causing the task-related operation to be obvious.

Effect of Instruction. In practice, instruction text describing the task (e.g., "Please predict the sentiment of this text:") can be utilized instead of the few-shot demonstrations in the inputs. Therefore, we examine whether such instructions produce the task-oriented information removal functionally similar to the few-shot demonstrations by measuring both metrics on the instructed inputs (configurations detailed in Appendix B.1.1). As shown in Fig. 4.5, hidden states with 8-shot demonstrations exhibit obvious morphological differences and stronger information removal over the 0-shot inference with or without instruction text. Also, hidden states with instruction, even if the label spaces are indicated (e.g., "Please predict in Positive and Negative"), show information removal dynamics almost consistent to the zero-shot inference (Kirsanov et al., 2025), despite the clearly higher accuracy (refer to Appendix B.1.3). Such observation suggests that the information removal is only evoked by few-shot demonstrations, with a different mechanism from instruction.

Effect of Random Labels and Unseen Labels. Previous works demonstrated that the correctness of labels presented in the demonstrations influences the inference dynamics and output accuracy insignificantly (Min

[11] The two curves differ in form, which is natural since accuracy is a nonlinear metric (Schaeffer et al., 2023).

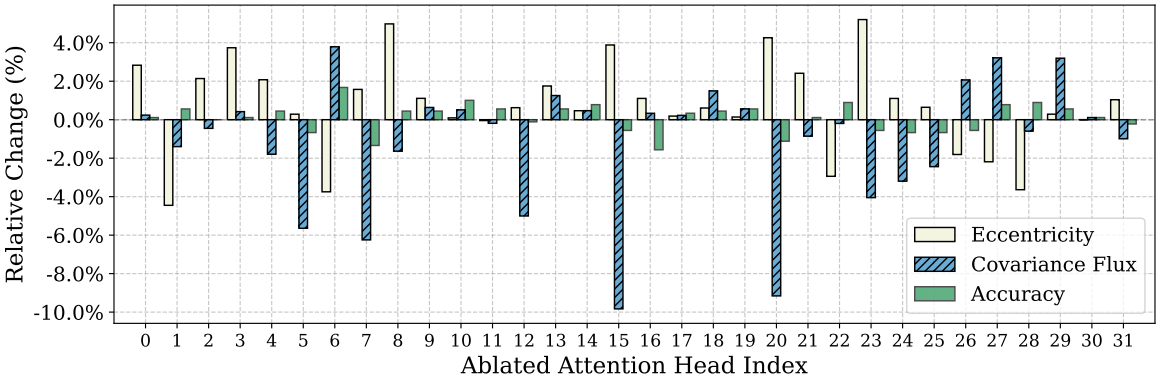


Figure 4.7: Ablation results for the two metrics and output accuracy of every head in Llama3.2-1B Layer 9 on 8-shot SST-2 (relative to the clean run). A smaller value indicates a more severe drop with ablation, i.e., this attention head contributes more significantly to the metric.

et al., 2022c; Yoo et al., 2022; Pan et al., 2023; Cho et al., 2025a). Therefore, we measure both the metrics with label tokens randomly sampled from all possible labels (detailed in Appendix B.1.1), as well as the unseen label settings, with results shown in Fig. 4.6. Focus on the eccentricity results, hidden states with randomized labels exhibit weaker information removal than the plain 8-shot inference, suggesting that mismatched labels weaken the impact of demonstration, leading to weaker task-oriented information removal, which echoes previous works (Cho et al., 2025a) showing that mismatched labels neutralize the functionality of demonstrations. In contrast, the demonstrations with unseen labels show a stricter removal, and we infer the reason as: demonstrations with a smaller label space actually specify a narrower task, causing more information to be identified as redundancy. More evidence is shown in Fig. B.40, where such an effect is weakened on multi-way tasks than on 2-way tasks, where the removal of each label has a large impact on the task scope. Overall, task-oriented information removal appears across all label settings, differing only in magnitude, regardless of the labels' completeness or alignment to the demonstration. The above degrading configurations merely shift the information removal away from the optimal range, resulting in reduced accuracy (we list the output accuracies in these configurations in Appendix B.1.3).

4.4.4 Step 3: Causality Test by Denoising Heads

We have shown that selectively removing information from zero-shot hidden states boosts accuracy in the injected settings, and few-shot learning under un-injected settings implicitly simulates a similar process. This section therefore investigates the causal role of such an operation. As outlined in Step 3 of §4.3, we first identify the model components responsible for this removal, thereby paving the way for subsequent ablation analysis to examine the significant accuracy influence, especially under the unseen label settings. The experiments in this section default to using 8-shot inputs.

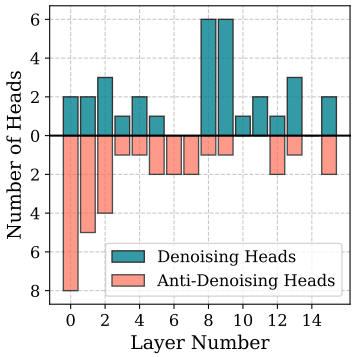


Figure 4.8: Counting of DHs and anti-DHs w.r.t. layers.

Identify Denoising Heads in LMs. Following the ablate-and-remeasure method mentioned in §4.3, a set of representative results on Layer 9 is shown in Fig. 4.7 (refer to Appendix B.9 for other settings). Because eccentricity trends have a phase-transition pattern[12], we use covariance flux to identify significantly contributing attention heads (i.e., ***Denoising Heads*** (DH)), with some (e.g., #5, 7, 12, 15, 20, . . .) showing clear contributions to both metrics and accuracy. Setting[13] $\pm 3.5\%$ as the thresholds, we observe the distribution of the DHs, and also the anti-DHs whose ablations increase the covariance flux remarkably, as shown in Fig. 4.8 (refer to Appendix B.9 for more

[12] As noted in §4.4.3, a contributive operation may reduce eccentricity, but in later layers (e.g., Layer 10), once redundancy is sufficiently removed and task-relevant information aligns with the first principal component, the contributive information removal begins to increase the eccentricity.

[13] Note that the goal of this experiment is to identify a subset of prominent denoising heads for visualization or causal analysis, rather than to exhaustively capture all potential candidates. Therefore, as discussed in Appendix B.5, the threshold should be a relatively selective but not extreme value. Based on the visualization in Fig. B.7, we consider 3.5% or 5% (mentioned below) to be an ideal choice.

settings). While the DHs can be observed in almost every layer, their numerical advantage against the anti-DHs emerges only in the middle-to-late layers, where task-oriented information removal becomes apparent as discussed in §4.4.3. This further suggests the transition from early-stage task-agnostic encoding (Cho et al., 2025a) to later-stage task-related filtering: in the former, most heads conduct non-selective query encoding, leading to anti-DHs; in the latter, more DHs play a selective and decisive role. Moreover, during this experiment, we find more interesting attention heads as shown in Appendix B.6.

Denoising Heads are Independent of Induction Heads. To demonstrate the independence of information removal behavior and also DHs against the induction heads, we observe the overlap of induction heads and the DHs. In detail, we calculate the magnitude of induction for each head as the sum of attention scores from all the label tokens to the last token, and visualize it together with the ablation effects on the covariance flux, as shown in Fig. 4.9 (refer to Appendix B.9 for other settings), where we find that: although induction heads and task-oriented information removal emerge in similar layers, these attention heads rarely overlap, which clearly suggests that: DHs, and their task-oriented information removal effect, is an independent and novel operation evoked by the ICL-prompted LMs.

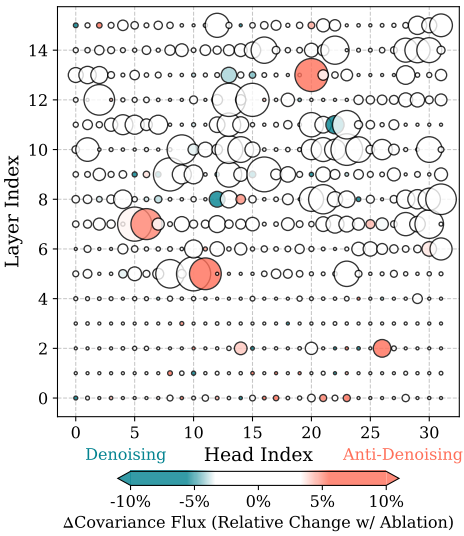


Figure 4.9: Head-wise (**scatter size**) induction magnitude and (**color**) covariance flux relative change after ablation.

Overlap of Denoising Heads among Tasks. To identify whether the evoking of the DHs is task-specific or task-irrelevant, which suggests **I**n-**w**eight **L**earning (IWL) (Chan et al., 2022; Reddy, 2024; Chan et al., 2025) or in-context learning characteristics, we calculate the overlap count of the DHs identified on all the task pairs as $|\mathcal{D}_A \cap \mathcal{D}_B|$, where the $\mathcal{D}_A$ or $\mathcal{D}_B$ is the DH set detected on the task A or B by bottom-K covariance flux relative change after ablation, as shown in Fig. 4.11 with $K = 1\%$. In the results, we observe several **common** DHs across different dataset pairs, indicating that they exhibit ICL properties by extracting and discriminating at least two task information from the context and leveraging it for operation. In contrast, most DHs are **dataset-specific**, reflecting the IWL property but still evoked by ICL demonstrations: on mismatched datasets, these heads are inactive, neither hindering nor facilitating information removal (e.g., Layer 9 Head 20 as a DH for sentiment-analysis in Appendix B.9), thus challenging current strict separation of ICL and IWL. These findings suggest that task-oriented information removal is a hybrid of ICL and IWL, supporting the previous works that ICL capability arises from latent tasks acquired during pretraining (Li et al., 2024) and is further strengthened by demonstrations (Yang et al., 2025c; Cho et al., 2025c).

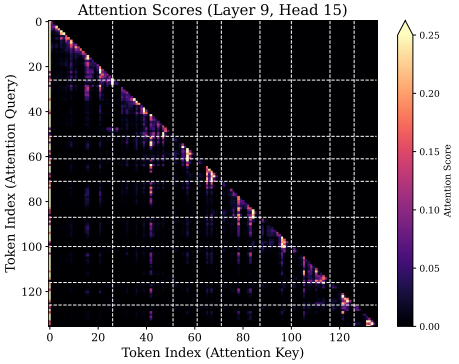


Figure 4.10: Attention map of a common DH (Layer 9, Head 15). White dotted lines are the position of label tokens.

Attention Pattern of Denoising Heads: Local Re-encoding. To find how the DHs conduct the task-oriented information removal operation, we visualize the attention maps of a common DH (Layer 9 Head 15) of Llama 3.2-1B, as shown in Fig. 4.10 (refer to Appendix B.1.4 for detail, Appendix B.9 for more cases). Surprisingly, although task-oriented information removal is a highly context-dependent operation in the ICL-based common DH, the DH exhibits a local attention pattern. Specifically, significant attention scores of the last token are almost confined to the query tokens (ignoring the attention sink (Gu et al., 2025)), thereby forming a local re-encoding pattern. In such processing, as demonstrated in Appendix B.4, DH appropriately identifies and enforces task-relevant information by selective attention calculation, where the vector extracted from the last-token hidden state by $W_Q^\top W_K$ of the DH may serve as an indicator for locating this information, since it serves an important part of the attention calculation, which is decisive for selecting the key tokens that contain the task-related information.

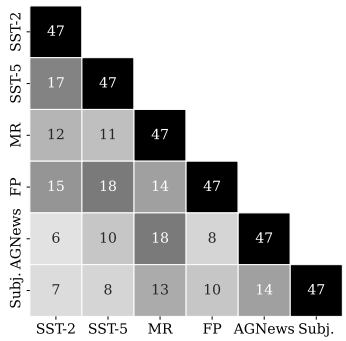


Figure 4.11: Overlap of DH over all dataset pairs.

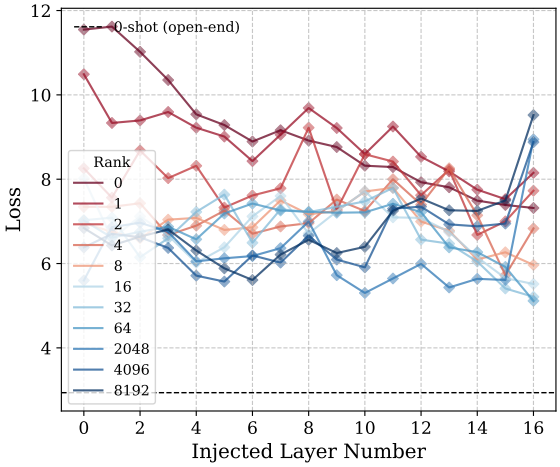


Figure 4.12: Filter-injection results on country-capitals task. In **cross-entropy loss**.

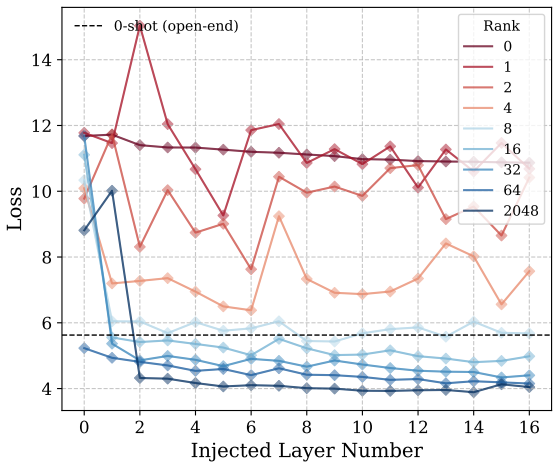


Figure 4.13: Filter-injection results on the people-profession task. In **cross-entropy loss**.

Task-oriented Information Removal as a Key Mechanism for ICL with Unseen Label. Identifying the DHs responsible for the task-oriented information removal enables ablation experiments to evaluate the significance of the operation to ICL. Therefore, we remove DHs from the inference, specifically, setting the outputs of all heads with a covariance flux relative change below -5% to zero vectors, and report the resulting accuracy in Table 4.2 (refer to Appendix B.9 for non-average results), where when demonstrations are randomly sampled, or at least one demonstration shares the same label as the query (Seen Label), zeroing out the DHs significantly but not critically reduces accuracy compared to randomly selected heads, since the induction heads provide a strong supportive mechanism in such cases. In the unseen label setting where induction heads fail, DHs become decisive: ablating them almost eliminates accuracy, even without exhaustively identifying all DHs[14]. This shows that DHs are the primary source of accuracy in unseen-label scenarios, effectively acting as the main complement to induction heads (Cho et al., 2025a).

Table 4.2: Ablated inference accuracy (%) of DHs, averaged on all 6 datasets. **Layer**: the ratio of the layer scanned for DH / total layers. **w/o DH**: accuracy with DHs ablated. **w/o RH**: accuracy with ablation of randomly sampled attention heads of equal amounts to DHs on each layer. **P-value**: the probability for "w/o RH" to sample a value less than "w/o DH" (i.e., the likelihood of DHs being randomly sampled). (Due to computational constraints, we only evaluate Llama 3-13B on SST-2.)

Model Layer	Demonstration Configuration	8-shot	8-shot w/o DH	8-shot w/o RH	P-value
Llama 3.2-1B 16 / 16	Random Sample	71.02	55.04	$67.29_{3.84}$	7×10^{-4}
	Seen Label	73.40	56.52	$69.02_{3.72}$	4×10^{-4}
	Unseen Label	14.54	1.06	$10.65_{4.65}$	2×10^{-2}
Llama 3-8B[14] 7 / 32	Random Sample	77.63	72.05	$76.38_{0.96}$	3×10^{-6}
	Seen Label	80.47	75.11	$79.70_{1.30}$	3×10^{-4}
	Unseen Label	21.66	7.52	$18.94_{3.98}$	2×10^{-3}
Qwen 2.5-3B[14] 18 / 36	Random Sample	73.97	68.41	$74.31_{1.29}$	2×10^{-6}
	Seen Label	76.41	70.31	$76.36_{1.57}$	1×10^{-4}
	Unseen Label	23.49	11.59	$26.82_{3.82}$	3×10^{-5}
Qwen2.5 3B-Ins[14] 18 / 36	Random Sample	77.24	75.57	$77.28_{1.90}$	0.18
	Seen Label	78.86	77.52	$78.05_{0.81}$	0.25
	Unseen Label	48.70	37.51	$46.32_{4.52}$	2×10^{-2}
Llama 3 13B Ins[14] 13 / 55	Random Sample	76.86	71.58	$77.83_{1.57}$	3×10^{-5}
	Seen Label	77.34	72.16	$81.20_{2.12}$	1×10^{-5}
	Unseen Label	29.59	10.35	$32.13_{4.89}$	4×10^{-6}

Amplifying the Outputs of DHs Boosts ICL Accuracy. As a prototypical empirical application utilizing the discovery of DH, we try to amplify the output of DH on 8-shot inputs, i.e., multiply a scalar $(1+\lambda)$ on the outputs of DHs detected similarly as the ablation in Table 4.2. The results with various λ are shown in Fig. 4.14, where a moderate amplification consistently increases ICL accuracy. Especially in the random-label scenario in §4.4.3, DH amplification yields a markedly stronger effect, consistent with Fig. 4.6, where information removal is weaker than in the true-label setting. This amplification provides a frugal, plug-and-play way to boost ICL accuracy and mitigate label noise without extra computational overhead, while reinforcing the causal link between information removal and ICL accuracy.

4.5 Conclusion and Discussion

Conclusions. This chapter proposes a novel perspective to interpret ICL inference as a task-oriented information removal from the query's hidden states. In detail, we first demonstrate that injecting explicit low-rank

[14]Due to computational constraints, we only identify the denoising heads on 50% of the layers of these models, as shown in Appendix B.9, so the ablation results here do not cover all denoising heads.

filters into the zero-shot hidden states to reduce their redundant information improves accuracy. We then show that few-shot ICL spontaneously performs this process, identify the DHs responsible for it, and confirm both the independence from the induction head, and the significant effectiveness of information removal for ICL tasks by ablation experiments.

Comparison and Connection to Previous Works. This chapter is related to the two categories of previous works: **(1) Task vector-based mechanism.** Our analysis of the attention patterns shows that denoising heads access subspaces of the last hidden states through $W_Q^\top W_K$ for the re-encoding operation. The $W_Q^\top W_K$ thus acts as an operator extracting task representations to trigger and guide the denoising operations. This view suggests a new direction for task-vector research: identifying the subspaces that purify the task representation from the coarse-grained hidden states, and finding the components that mediate their effects (Yang et al., 2025c; Yin & Steinhardt, 2025), rather than only relying on coarse steering. **(2) Induction heads-based mechanism.** As mentioned, current induction head-based works slough over the case of unseen labels and resort to vague explanations (Cho et al., 2025a), which our chapter addresses. From our ablation study (Table 4.2), ICL behavior can be attributed almost fully to induction and DHs, with no significant bypasses since ablating DHs in the unseen label scenario reaches near-0 accuracy. Although induction heads contribute more to accuracy, the DH, as the complementary mechanism identified when they fail, remains meaningful.

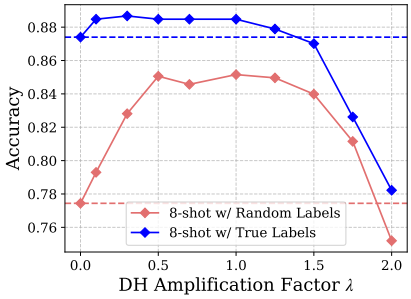


Figure 4.14: DH amplification results on Llama 3.2-1B and SST-2.

Failure Case: Clustering vs. Translation. Notice that the task-oriented information removal investigated in this chapter is essentially a dimensionality reduction operation, where it is conceivable that classification tasks, which cluster a vast input space (i.e., the whole set of the input sentences x) to a narrow label space, precisely align with such a dimensionality reduction. While, some works identified translation operation (Merullo et al., 2024; Jiang et al., 2024; Bu et al., 2025) in hidden states directing entities' semantics towards their attributes, especially unique ones (called "Fact Recall" (Zhang et al., 2025), e.g., "Japan"→"Tokyo"), where the clustering operation as presented in classification tasks can not be applyed since the mapping from the input space to the output space is a bijection. Therefore, the task-oriented information reduction process observed in this chapter is unlikely to apply to the bijection scenario, as information is not removed: input can be losslessly reconstructed from the output. We demonstrate this point by repeating the filter-injection experiment shown in §4.4.2 on the country-capitals task as shown in Fig. 4.12, where no loss lower than the 0-shot baseline can be obtained by any filter, even if high-dimensional (not necessarily high-rank) ones. As a comparison, we repeat the experiment on a similar fact recall task but with a clustering structure, which maps a person's name to their profession (e.g., "F. Nightingale"→"Nurse"). As shown in Fig. 4.13, evaluation results better than 0-shot can be obtained by a low-rank filter, which confirms our idea that the task-oriented information removal does not occur in bijective ICL tasks, while clearly highlighting the distinction between these two types of tasks. Some works on synthetic datasets also discuss the uniqueness of such a full-rank translation scenario (Dong et al., 2025) and obtain similar conclusions. We develop such a discussion on a generative scenario in Appendix B.8.

Limitations. (1) Long-term effect of head ablation. In Fig. 4.7, we identify DHs by ablated metrics only on the current layer's hidden states. However, since the effect of ablation may propagate to deeper layers, the DH sets could be underestimated. Exhaustively measuring all layers would be costly, yet our ablation already shows that these heads, though not the whole set of DHs, produce clear and significant effects on the outputs. **(2) Fine-grained mechanism.** We propose a prototype mechanism for DHs as the extraction of denoising signals from some subspaces of hidden states. A more detailed investigation is needed to validate such a mechanism prototype for DHs. **(3) Efficient localization of DHs.** In §4.4.4, we adopt a naive ablation-remeasurement method, which is costly for localization-controlling applications (Cho et al., 2025b). Future work may utilize gradient-based methods, or leverage DH characteristics, such as the attention pattern in Fig. 4.10, for more efficient localization. **(4) Complex information bottleneck.** In this chapter, we utilize a linear low-rank filter as the information bottleneck as shown in Fig. 4.1 (C). This essentially assumes that the features on these subspaces are linearly separable in the linear space, as illustrated in Fig. 4.1 (A). However, some works (Engels et al., 2025; Modell et al., 2025) show that features may also be embedded on manifolds and be nonlinearly separable. Therefore, extending the discussion to more complex tasks may require a more complex nonlinear information bottleneck.

Part III
Application

III Application **34**

5 Hidden Calibration for Improving the Decision Boundary of ICL **35**
5.1 Introduction . . . 35
5.2 Background . . . 37
5.2.1 In-context Learning . . . 37
5.2.2 Token-probability Calibration for ICL . . . 37
5.3 Methodology . . . 37
5.3.1 Token Probabilities Are Not Good Classification Criteria . . . 37
5.3.2 Hidden Calibration . . . 38
5.4 Experiments & Main Results . . . 39
5.4.1 Experimental Settings . . . 39
5.4.2 Main Results: Hidden Calibration is A New State-of-the-art of ICL . . . 40
5.4.3 Efficiency: Low Complexity towards Time, Space, Data, and Prompting . . . 40
5.5 Analysis . . . 41
5.5.1 Effectiveness: Hidden Calibration Finds Criteria with Lower Overlap . . . 42
5.5.2 Principle: The Inner Linear-separability . . . 42
5.6 Related Works . . . 43
5.7 Discussion . . . 44
5.8 Limitations . . . 44

6 Low-resource Attention Behavior Fine-tuning for ICL **46**
6.1 Introduction . . . 46
6.2 Background & Related Works . . . 47
6.3 Attention Behavior Fine-tuning . . . 48
6.4 Main Experiments . . . 49
6.4.1 Experiment Settings . . . 49
6.4.2 Main Results . . . 50
6.4.3 Comparison against End-to-end Fine-tuning . . . 51
6.5 Analysis . . . 51
6.5.1 Attention Visualization after ABFT . . . 51
6.5.2 Ablation Analysis . . . 52
6.5.3 Balance the Loss Factor A and B . . . 52
6.6 Consistency of Training Objective: ABFT and End-to-end Fine-tuning . . . 53
6.7 Discussion . . . 53
6.8 Limitations . . . 54

Chapter 5

Hidden Calibration for Improving the Decision Boundary of ICL

Token-based Decision Criteria Are Suboptimal in In-context Learning

This chapter mainly retells the paper "Token-based Decision Criteria Are Suboptimal in In-context Learning" in NAACL 2025 main conference paper by Yufeng Zhao (alias: Hakaze Cho) (refer to Publication List).

The connections from the previous chapters: In chapter 3, we use centroid classifiers on the hidden states of LMs and achieve considerable accuracies, which implies a better decoding strategy than the typical constrained decoding. This chapter mainly confirms such an idea and develops a new and robust state-of-the-art in ICL named Hidden Calibration.

Abstract

In-**C**ontext **L**earning (ICL) typically utilizes classification criteria from output probabilities of manually selected label tokens. However, we argue that such token-based classification criteria lead to suboptimal decision boundaries, despite delicate calibrations through translation and constrained rotation applied. To address this problem, we propose Hidden Calibration, which renounces token probabilities and uses the nearest centroid classifier on the LM's last hidden states. In detail, we assign the label of the nearest centroid previously estimated from a calibration set to the test sample as the predicted label. Our experiments on 6 models and 10 classification datasets indicate that Hidden Calibration consistently outperforms current token-based baselines by about 20%∼50%, achieving a strong state-of-the-art in ICL. Our further analysis demonstrates that Hidden Calibration finds better classification criteria with less inter-class overlap, and LMs provide linearly separable intra-class clusters with the help of demonstrations, which supports Hidden Calibration and gives new insights into the principle of ICL.

5.1 Introduction

In-**c**ontext **L**earning (ICL) (Dong et al., 2022) is a few-shot learning paradigm without model parameter updates on **L**anguage **M**odels (LMs). In detail, as shown in Fig. 5.1-(A, B), given a prompt consisting of demonstrations and a query, LMs conduct causal language modeling operation from the prompt to assign probabilities to the label token candidates designed by hand, and ICL chooses the one with the highest probability as the prediction.

One well-known issue of ICL is that the predicted probabilities are biased (*under-calibrated*), leading to a decrease in prediction performance (Fei et al., 2023; Han et al., 2023c; Zhao et al., 2021; Zhou et al., 2024a). To address this issue, previous work *calibrates* the predicted label token probabilities by performing affine transformations with estimated parameters to adjust these probabilities for more precise predictions.

These previous works and also the vanilla ICL are based on a potential assumption: the affine manifolds spanned by the decoding vectors in the LM head (*un-embedding vectors*) of *manually selected* label tokens are good subspaces of the hidden space to distinguish hidden vectors (i.e., the last hidden states to be fed to the LM head) onto various label appropriately, so that the label token probabilities decoded from these subspaces are accurate classification logits. However, although using the label un-embedding with task-related semantics (e.g. "positive" and "negative") seems intuitive, it should be noted that we have no reason to believe that these label un-embeddings have any explicit guarantee for decoding the last hidden states

into accurate classification logits (token-based decision criterion), even if various delicate calibrations are used to move these boundaries *inside the subspaces* (see §5.3.1). Also, some works have pointed out that randomly changing label spaces doesn't critically influence ICL performance (Min et al., 2022c; Wei et al., 2023a), which means the selected label subspaces are trivial and arbitrary, making a suspicion of: using manually selected label un-embeddings to decode the last hidden states, i.e., **utilizing manually selected label probabilities as classification criteria may not be good ICL practices**.

Previous work has shown that using the output probabilities of the *full vocabulary* increases ICL performance (Xu et al., 2022; Abbas et al., 2024). This is a good start to avoid the manually selected classification criteria, but there is still doubt that output probability distributions are not informative enough for classification (see §5.3.2). Therefore, we utilize the last hidden states instead, which are informative precursors of the token probabilities.

Concretely, we propose Hidden Calibration, training centroid classifiers on the last hidden states of ICL prompts. As shown in Fig. 5.3, **during the training**, we build standard ICL prompts similarly to Fig. 5.1-A from a supervised calibration set and input them into the LM to get the last hidden states of the last tokens of ICL prompts. Then, we calculate the centroids of the last hidden states w.r.t. the queries' label to get a centroid for each label, as an anchor for inference. **During the inference**, we input the test prompt, find the nearest centroid to the last hidden states of the test prompt, and assign the corresponding label of the centroid as the prediction.

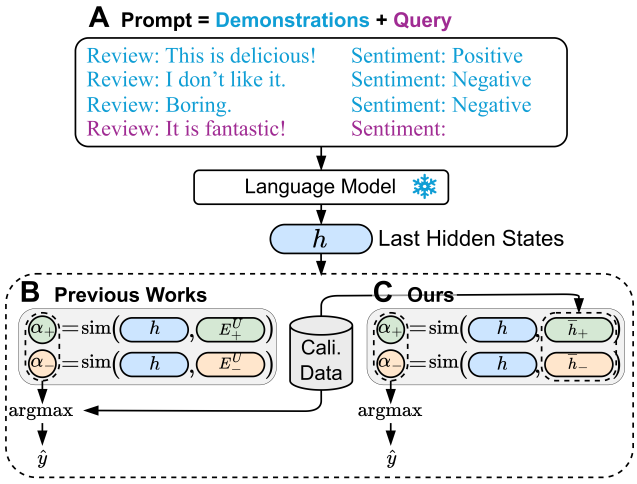


Figure 5.1: In an ICL diagram, **A.** The prompt of ICL consists of a concatenation of demonstrations and a query. LMs encode the prompt into the last hidden state h, then **B.** (previous works) use the un-embedding vectors of the label tokens (E^U_+, E^U_-) to decode the h to prediction $\hat{y}$, then calibrations are used to adjust the predicted logits. **C.** Our work uses the calibration dataset to calculate centroids ($\bar{h}_+$, $\bar{h}_-$) to decode the h.

Empirically, Hidden Calibration improves the ICL performance by approximately more than 20% on 10 text classification datasets and 6 modern LMs (§5.4.1), with an equal computational cost with previous calibration methods. To the best of the author's knowledge, Hidden Calibration consistently outperforms the calibration baselines, achieving a strong state-of-the-art in ICL. Additional experiments indicate that Hidden Calibration effectively alleviates the stress of prompt engineering, performing robust accuracy under different prompt designs.

Moreover, our subsequent analysis indicates that Hidden Calibration does find better logits-mapping subspaces that effectively separate data points. In detail, we find that the distribution of classification logits calculated from Hidden Calibration have less inter-class overlapping than from label probabilities, while such overlapping is proportional to the lower bound of the classification error. This suggests Hidden Calibration finds subspaces with essentially better classification performance.

Furthermore, we investigate the principle of Hidden Calibration, that is, the reason why a simple centroid-based linear decision boundary can divide the ICL hidden state properly. We find that LMs provide linearly separable clusters in the hidden states w.r.t. query labels, while more demonstrations can promote such a process.

Our contributions can be summarized as:

- We analyze the previous calibration practices on ICL, and find their consistent limitations: using predicted probabilities of manually selected label tokens for classification criteria, which is often under-guaranteed.
- We propose Hidden Calibration to address the problem before, eliminating the unreliable decoding on the hand-selected label, and using a centroid classifier on the LM's last hidden states. Our experiments suggest that Hidden Calibration reaches a strong state-of-the-art.
- Our further analysis indicates that Hidden Calibration does find better classification criteria with less inter-

class overlap, and LMs provide linearly separable intra-class clusters with the help of demonstrations, which supports Hidden Calibration to classify samples accurately.

5.2 Background

This section reviews previous work on ICL and denotes their mathematical descriptions as an introduction to the main motivation of this work.

5.2.1 In-context Learning

Prompting. Given a few-shot natural language classification dataset (*demonstration set*) $\mathcal{D} = \left\{\left(x^{(i)}, y^{(i)}\right) \in \mathcal{X} \times \mathcal{Y}\right\}_{i=1}^{n}$, where $x^{(i)}$ and $y^{(i)}$ are the input sequence and label token of i-th data point, and $\mathcal{X}, \mathcal{Y}$ is the input and label space, respectively, we sample a set of k samples $\mathcal{D}^{de} = \left\{\left(x^{(c_i)}, y^{(c_i)}\right)\right\}_{i=1}^{k}$ from $\mathcal{D}$ with an index set $\{c_i\}_{i=1}^{k}$ for a given query x^q as the demonstrations. Then, we use a template T to concatenate them in a natural language form into a prompt token sequence: $s = T\left(\mathcal{D}^{de}, x^q\right)$, as shown in Fig. 5.1-A.

Encoding. A decoder-structured LM receives the prompt token sequence s and encodes it into the *last* (from the last Transformer layer) hidden state matrix as $H \in \mathbb{R}^{|s| \times d}$ with a length of token $|s|$ and embedding dimension of d. We denote the hidden state of the last token as $h = H_{|s|} \in \mathbb{R}^d$.

Constrained Decoding. In a typical ICL setup, one chooses the un-embedding vectors of the label candidates in the output head[15] to decode h as the prediction. That is, for each label l, the similarity $\alpha_l = \mathrm{sim}(h, E_l^U)$ (usually the dot-product similarity) between h and the un-embedding vector E_l^U is calculated as the output logits α_l, as shown in Fig. 5.1-B for a binary classification example. Then, the label with the highest logits is chosen as the prediction $\hat{y}$, that is: $\hat{y} = \underset{l \in \mathcal{Y}}{\mathrm{argmax}}\, \mathrm{sim}(h, E_l^U)$.

5.2.2 Token-probability Calibration for ICL

However, Zhao et al. (2021) find that simply using the original logits for classification can not lead to a good ICL practice, since these logits have considerable prior bias and often tend towards specific labels even if the query is blank or meaningless (Zhao et al., 2021; Fei et al., 2023). Some calibrations have been proposed to mitigate such bias in a linear form: first, the logits are transformed into probabilities as $p = \mathrm{softmax}\left(\left[\alpha_1, \alpha_2, \ldots, \alpha_{|\mathcal{Y}|}\right]\right)$, then affine-transformed as calibrated classification criteria $p' = A \odot p + B$, where $A, B \in \mathbb{R}^{|\mathcal{Y}|}$ is the calibration terms estimated from m training examples from a calibration set, and $\odot$ is the Hadamard multiplication. Various estimations for A and B are used: some practices use examples with pseudo queries terms (Fei et al., 2023; Zhao et al., 2021), while other practices use Gaussian estimation on real prompts (Han et al., 2023c) or the mean value of p during the inference (Zhou et al., 2024a).

However, as to be discussed in §5.3.1 current calibrations are affine transformations on label token probability, without modifying the E_l^U, causing only limited improvement to ICL performance.

5.3 Methodology

Based on the above background, in this section, we demonstrate the limitations of the above calibrations, and then propose Hidden Calibration to address such limitations fundamentally.

5.3.1 Token Probabilities Are Not Good Classification Criteria

To better understand the limitations of the label token probability-based ICL, we show a prototypical visualization of the hidden states of ICL prompts (aforementioned h). Specifically, we input 2,048 ICL prompts (with $k = 8$) built from of `SemEval 2014-Task 4 Restaurant` (Pontiki et al., 2014) into OPT-2.7B (Zhang et al., 2022) and plot the h on a 2D-Principal Component plane in Fig. 5.2 (detailed in Appendix C.1.4).

[15] We omit the bias term in the output head (if any) for the sake of simplicity, which can be overridden by a fixed-to-one dimension, or covered by the calibration described below.

As a simple 2-way case, focusing on the data points labeled "positive" and "negative", we plot the difference direction $\left(E_+^U - E_-^U\right)$ between the un-embedding vectors of these two label tokens[16]. Then, the coordinates of the projected hidden states in this direction are the difference of predicted logits between these two labels, serving as a *token-based classification criteria*, i.e., when the coordinate is positive, a "positive" label will be assigned, and vise versa. Therefore, in this visualized scenario, the orthogonal line at the zero point is the original decision boundary, points below this boundary are classified as "positive", and vise versa. The batch calibrated boundary (Zhou et al., 2024a) is always parallel to the original one, and the other calibrations (Contextual Calibration (Zhao et al., 2021), Domain Calibration (Fei et al., 2023)) produce rotated mapping directions $\left(A_+E_+^U - A_-E_-^U\right)$, by positive-definite term A and thus rotated decision boundaries, with limited direction[17] between E_+^U and $-E_-^U$.

Intuitively, as shown in Fig. 5.2, the token-based decision boundaries cannot effectively classify these data points, which is due to the inherent direction of the token un-embedding vectors, regardless of a limited affine transformation by calibration. A straightforward better linear boundary is the equidistant points between both classes' centroids, we try to find it as follows.

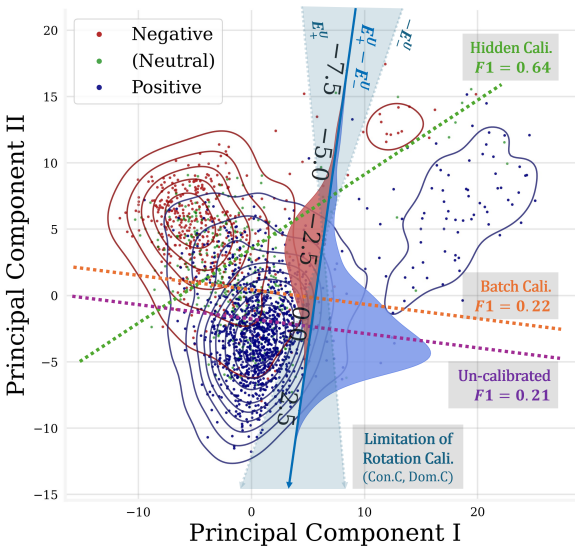


Figure 5.2: Token probability-based decision boundaries (original & batch calibrated) are suboptimal comparing to centroid-based boundary. Points and contour lines are ICL's last hidden states and kernel densities mapped by Principal Component Analysis. Oblique coordinate axis is the direction of the un-embedding difference of label tokens $\left(E_+^U - E_-^U\right)$, where the kernel densities of mapped data points are plotted. The rotating calibration by $A \neq \mathbf{1}$ (e.g. Contextual Calibration, Domain Calibration) has a limited feasible mapping direction[17].

5.3.2 Hidden Calibration

Motivated by the visualization, we propose Hidden Calibration, using the centroid similarity as the classification logits. In practice, we use a 2-step paradigm as shown in Fig. 5.3: first, as training, we calculate the centroid of the last hidden states of data points within each class on some (of amount m) prompt-label training examples. Then, in the inference, we select the closest centroid of the test prompt's hidden state as the prediction.

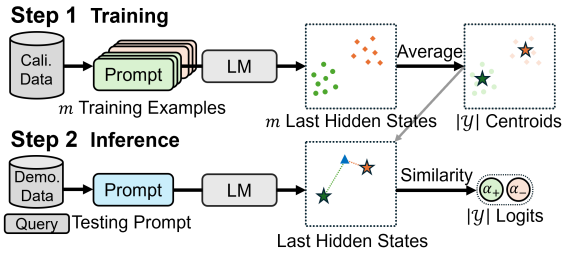


Figure 5.3: The diagram of Hidden Calibration. **Step 1:** Calculating the hidden state centroid of each label. **Step 2:** Find the label of the nearest centroid of the text sample to be the prediction.

In detail, **(1) Training:** Given a *calibration set* with m supervised prompt-label pair $\left\{\left(s^{(i)}, y^{(i)}\right)\right\}_{i=1}^m$, where the $s^{(i)}$s (*training examples*) are standard ICL prompts with k demonstrations, and $y^{(i)}$s are the ground-truth labels of corresponding $s^{(i)}$s' query, we input each training example $s^{(i)}$ to LM, and extract the last hidden state $h^{(i)}$. Repeating on the whole training example set, we can get a supervised hidden state set $\mathcal{H} = \left\{\left(h^{(i)}, y^{(i)}\right)\right\}_{i=1}^m$. Then, we calculate the centroids of label l as: $\bar{h}_l = \mathbb{E}_{\left(h^{(i)}, y^{(i)}\right) \in \mathcal{H}, y^{(i)}=l}\left[h^{(i)}\right]$.

Then, we utilize the calculated centroids in **(2) Inference:** Given a test ICL prompt, we input it into the LM

[16]Notice that Principal Component Analysis is an orthogonal transformation, keeping the dot-product and normal line fixed (In fact, beyond orthogonal transformations, they are also centralized. Therefore, the projection axis does not necessarily pass through the coordinate origin). See Appendix C.1.4.

[17]In current practices, the A are calculated from reciprocals of probabilities, which are positive-definite (Note that the calibration is trivial when A is not positively definite: the label with negative A components will never be assigned), and usually do not have significant relative values.

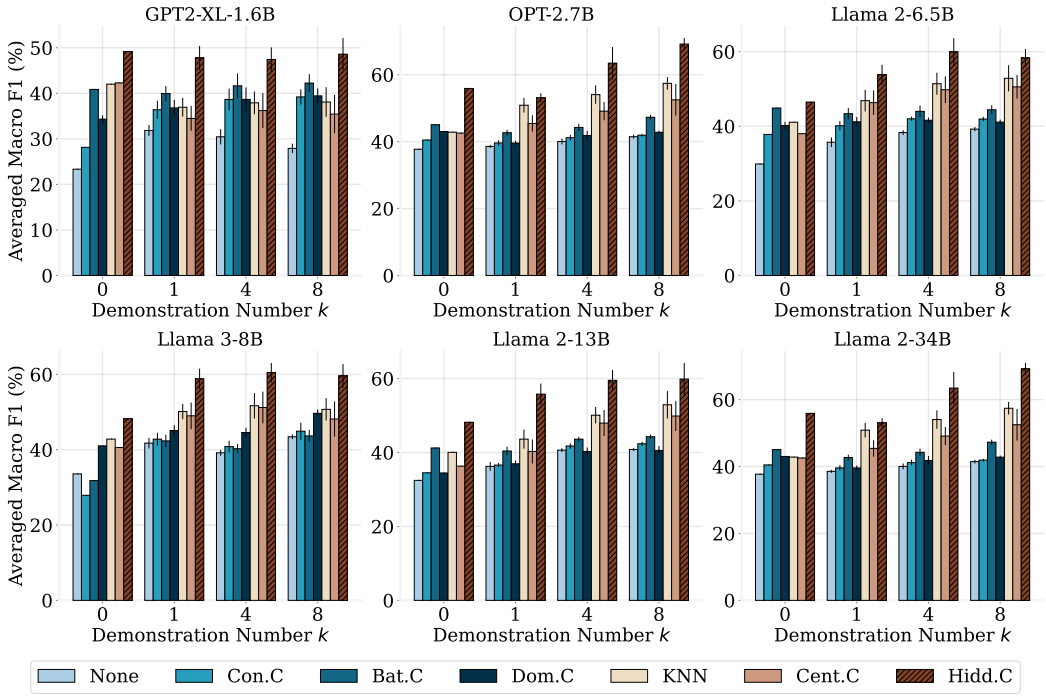


Figure 5.4: The classification performance (Macro F1(%)) of 6 models averaged on 10 datasets. Hidden Calibration (Hidd.C) is a new state-of-the-art of ICL, where demonstrations consistently improve the performance.

and get the last hidden state h, then calculate the similarity between h and every centroid $\bar{h}_l$ as the centroid-based logits α_l. In practice, the additive inverse of Euclidean distance is used as the similarity (that is, $\alpha_l = -\left\|h - \bar{h}_l\right\|_2^{\frac{1}{2}}$), while Appendix C.3.1 shows that Hidden Calibration acts equally on cosine similarity. We assign the label with the highest logits as the prediction.

"Why hidden states?" Notice that another intuitive solution to the problem in §5.3.1 is utilizing the logits or probabilities of the *whole vocabulary*, as shown in previous works (Xu et al., 2022; Abbas et al., 2024). However, since the input, hidden states, and logits form a Markov chain, no input-relevant information gain is propagated to the full-vocabulary logits. Moreover, the dimensionality of the full-vocabulary logits is typically significantly larger than the hidden states, therefore we choose the hidden states, a dense and informative precursor of token probabilities, as the classification feature.

"Why centroid classifier?" Moreover, more complex classifiers, such as a KNN classifier, or a multi-layer perceptron, can be used on the last hidden states instead of a centroid classifier. However, we choose the centroid classifier as the simplest implementation to avoid attribution confusion, that is, if even a rudimentary classifier on hidden states still outperforms, it is powerful enough to demonstrate that our hypothesis is robust. Also, a centroid classifier has a minimal computation cost to fit the scenario of low-resource (§5.4.3).

5.4 Experiments & Main Results

In this section, we empirically prove the effectiveness of Hidden Calibration by classification performance on 6 models and 10 datasets. Hidden Calibration outperforms all the baselines and reaches a strong state-of-the-art of ICL with high efficiency in calculation, data, and prompt engineering.

5.4.1 Experimental Settings

Models. We use 6 models: OPT-2.7B (Zhang et al., 2022), Llama 2 (Touvron et al., 2023) (6.9B, 13B, 34B), Llama 3 (AI@Meta, 2024) (8B) and GPT 2-XL (Radford et al., 2019) (1.6B). Models larger than 10B are quantized.

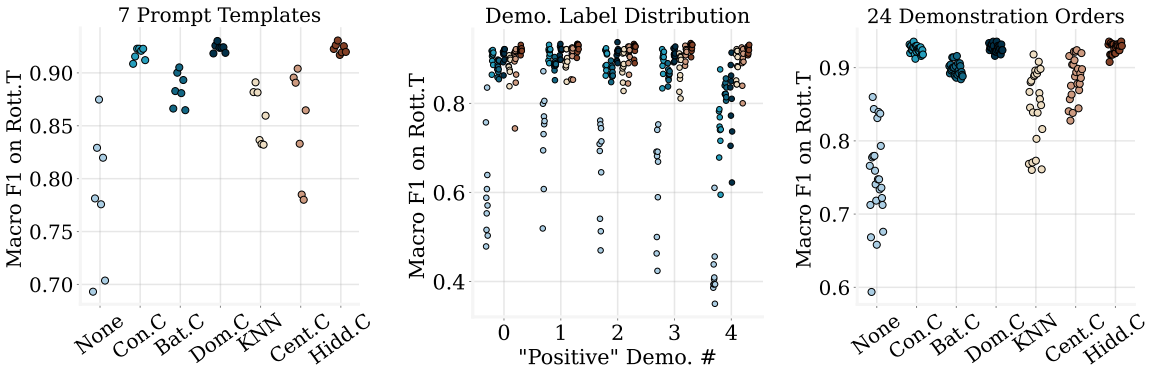


Figure 5.5: Sensitivities on (**left**) prompt template, (**middle**) demonstration label distribution, and (**right**) demonstration order on Llama 2-6.9B and `Rotten_Tomatoes`. Legend is consistent with Fig. 5.4, omitted.

Baselines. We use 6 baselines from the previous works, with 4 label token-based methods: **Vanilla ICL** (None) (Radford et al., 2019), **Contextual Calibration** (Con.C) (Zhao et al., 2021), **Batch Calibration** (Bat.C) (Zhou et al., 2024a), and **Domain Calibration** (Dom.C) (Fei et al., 2023); 2 whole vocabulary probabilities-based methods KNN (Xu et al., 2022) and **Centroid Calibration** (Cent.C), which we propose as a fair comparison with the same processing on the whole output vocabulary probability vectors instead of the hidden states. Details can be found in Appendix C.1.2.

Datasets. We use 10 commonly used classification datasets with some of the overlength data points excluded. See Appendix C.1.1 for details.

Other details. All the models and datasets are loaded from `HuggingFace`. Macro F1 is used as the classification metric. We use a simple template to generate the prompt, see Appendix C.1.3. We set $m = 16|\mathcal{Y}|$ training examples (16 examples per class), and for fairness, every baseline is given equal training examples for calibration. All the experiments are repeated 5 times.

5.4.2 Main Results: Hidden Calibration is A New State-of-the-art of ICL

The tested classification performance of Hidden Calibration and baselines is shown in Fig. 5.4, where Hidden Calibration (Hidd.C) consistently outperforms all the label token-based or vocabulary-based baselines. Comparing to the vanilla ICL (None), Hidden Calibration produces an improvement up to around 100%. In general, compared to the strongest baseline, Hidden Calibration improves the performance by approximately 20%. Detailed numeric and Accuracy results are in Appendix C.2.1.

Especially, compared to the Cent.C baseline proposed by us for a controlled trial, which conducts the same calculation but uses the whole output token probabilities instead of the hidden states, Hidden Calibration outperforms, which confirms our idea that token probability distribution is a less informative classification feature mentioned in §5.3.2.

5.4.3 Efficiency: Low Complexity towards Time, Space, Data, and Prompting

Time and Space Complexity. Intuitively, Hidden Calibration has little additional computational cost compared to the calibration baselines, since they require almost equivalent feedforward calculations, making it competitively efficient as listed in Table 5.1. Here, we are most concerned about the inference time cost, and Hidden Calibration is the fastest among all the non-label-based methods since the product $|\mathcal{Y}|d$ is usually not very large.

Training Data Complexity. Hidden Calibration requires additional annotated data compared to label token-based calibration methods of the same scale in an acceptable range. In detail, in Con.C and Dom.C of k demonstrations, k supervised data is needed with a synthetic query to build a training example, while Hidden Calibration needs a real query **for each label**, requiring one more supervised data. However, for classification tasks, preparing an example for each label can be easily done whether in an industry or laboratory scenario, furthermore, these data and trained centroids can be reused (Appendix C.3.2) to further reduce the requirement of annotated data.

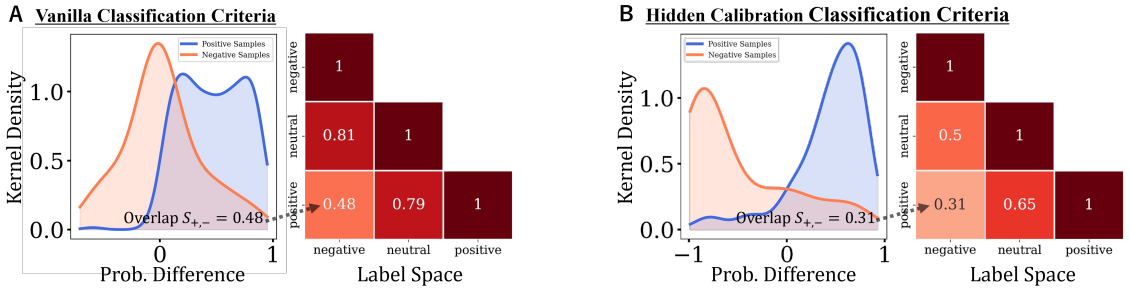


Figure 5.7: Diagrammatic sketch of the overlap calculation with GPT 2-XL on `SemEval 2014-Task 4 Restaurant`, $k = 4$. **Curves:** The kernel density of probability difference of l_1 ="positive" and l_2 ="negative". **Heatmaps:** The overlap of 2-combinations (we plot the combination with the same label with overlap 1, but omit them in averaging).

Training Sample Efficiency. Regarding the efficiency of training examples, we repeat the experiments with various m on OPT-2.7B (see Appendix C.1.7 for details), from 1 to 128 calibration examples **per class**. Also, for the vanilla ICL, we give equivalent demonstrations for a fair comparison. The results are shown in Fig. 5.6, which indicate that Hidden Calibration stably benefits from the size of the calibration set, while even one sample per class can still make it outperform. Meanwhile, vanilla ICL and label token-based methods can not benefit from more available data, making Hidden Calibration a better practice no matter how much supervised data can be accessed: data can be used to estimate the centroid to improve the classification in a linear cost, rather than increase the demonstrations in a quadratic cost with less benefit.

Table 5.1: The **additional** (compare to vanilla ICL) time and space on calibration and inference cost of various methods. Hidden Calibration has a similar cost upper bound to other calibrations. $|\mathbb{V}|$ is the vocabulary size.

Method	**Training Cost**		**Inference Cost**
	Add. Space	Add. Time	Add. Time
None	0	0	0
Con.C	$O(\lvert\mathcal{Y}\rvert)$	$O(m)$	$O(\lvert\mathcal{Y}\rvert)$
Bat.C	0	0	$O(m\lvert\mathcal{Y}\rvert)$
Dom.C	$O(\lvert\mathcal{Y}\rvert)$	$O(m)$	$O(\lvert\mathcal{Y}\rvert)$
KNN	$O(m\lvert\mathbb{V}\rvert)$	$O(m)$	$O(m\lvert\mathbb{V}\rvert)$
Cent.C	$O(\lvert\mathcal{Y}\rvert\lvert\mathbb{V}\rvert)$	$O(m)$	$O(\lvert\mathcal{Y}\rvert\lvert\mathbb{V}\rvert)$
Hidd.C	$O(\lvert\mathcal{Y}\rvert d)$	$O(m)$	$O(\lvert\mathcal{Y}\rvert d)$

Prompting Complexity. We find Hidden Calibration reduces the pressure on prompt engineering for Hidden Calibration and baselines on Llama 2-6.9B and `Rotten_Tomatoes` ($k = 4$) in three aspects: **(1) Prompt template** (Voronov et al., 2024). We select 7 different prompt templates (shown in Appendix C.1.3) and test ICL performance on them, shown in Fig. 5.5 (left). **(2) Label distribution in demonstrations.** We construct prompts with various numbers of "positive" demonstrations presented, and test ICL performance shown in Fig. 5.5 (middle), **(3) Demonstration order** (Lu et al., 2022). We enumerate the full arrangements from a fixed demonstration set of $k = 4$, and test the ICL performance using each demonstration arrangement, shown in Fig. 5.5 (right). All the results show that: compared to baselines, Hidden Calibration keeps narrow and high-performance distribution against all the three variables, i.e., Hidden Calibration stably works for various contexts, providing higher efficiency on prompt designing.

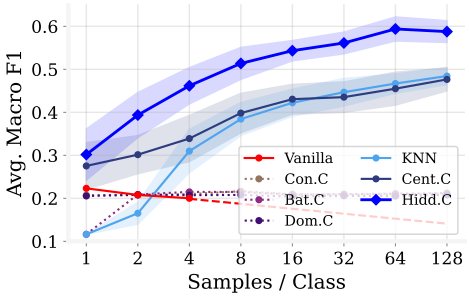


Figure 5.6: Classification performance against the number of training examples (m) of calibrations. As a comparison, we plotted the results of vanilla ICL using equivalent demonstrations, due to the quadratic overhead against the context length, we can test up to $k = 4|\mathcal{Y}|$.

5.5 Analysis

This section attempts to enhance our understanding of Hidden Calibration through comprehensive observations: **(1)** Similar to Fig. 5.2, we measure the inter-class overlapping area on data points projected into classification criteria, to find whether Hidden Calibration maps data into logits with lower inter-class overlap, i.e., better separability. **(2)** We further investigate why simple linear boundaries can effectively classify ICL hidden states, as observed in Hidden Calibration. We find that LMs provide a primary linear clustering in hidden states responding to query classes, and such clustering is enhanced by more demonstrations.

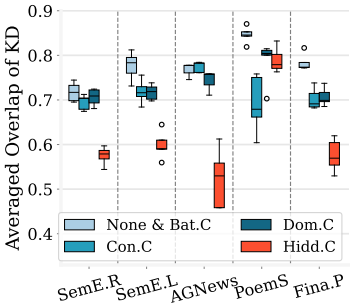


Figure 5.8: The Averaged Overlap of 4 inference methods on GPT 2-XL and 5 datasets.

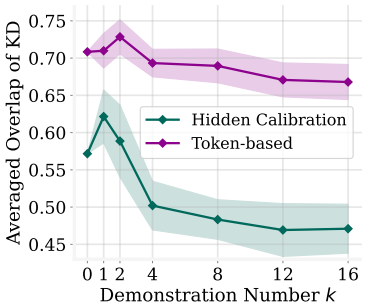


Figure 5.9: The Averaged Overlap on OPT-2.7B and 5 datasets against the demonstrations number.

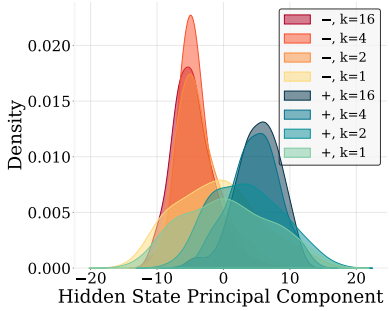


Figure 5.10: Hidden state clustering w.r.t. k of Fig. 5.12 visualized on the direction of principal component II.

5.5.1 Effectiveness: Hidden Calibration Finds Criteria with Lower Overlap

In Fig. 5.2, we projected the data points into the difference of the label logits (*vanilla classification criteria*) on the oblique coordinate axis, then a significant *overlap* between the projected data point cloud in two classes can be observed, making it difficult to find suitable classification boundaries vertical to the projection direction. Therefore, such overlap can be used to evaluate classification criteria, so, in this section, we quantify the intuitive observation as the area of overlap serving as a metric for classification criteria.

In detail, we first decompose the multi-way classification dataset into all possible binary classification combinations w.r.t. the ground-truth labels (for example, in a binary combination with labels "positive" and "negative", only "positive" and "negative"-labeled data is obtained). Then, for each combination, we build standard ICL prompts with queries labeled with a specific one of the selected binary combination. Input these prompts into the LM and map the last hidden state of the prompts onto the normal vector of the decision boundary formed by the calibration method, we get the mapped coordinate, as what is shown in the oblique coordinate axis of Fig. 5.2. To get a continuous distribution of the distance, we run kernel density estimations on the calculated coordinate, then repeat this processing on the other label in the binary combination, and get two density estimations for both labels in a binary combination, as shown in Fig. 5.7 (curves). Then, we calculate the overlap area of these two kernel density curves. We repeat such processing for each binary combination as shown in Fig. 5.7 (heat maps), and the final Averaged Overlap is the macro average of overlap area among all possible binary combinations (operation details are in Appendix C.1.5).

The overlap area of the two distribution curves is double to the *lower bound* of the classifier's error rate among these two labels (Appendix C.1.5.3), so Averaged Overlap is an intuitive metric of the classification criteria: the larger the overlap, the more difficult it is for the classifier, even (further) calibrated or ideal, to classify data points correctly, resulting in a potential decrease in accuracy.

We measure the Averaged Overlap of 4 un-quantized models on 5 datasets (see Appendix C.1.5.2 for experimental details). The result on GPT 2-XL is shown in Fig. 5.8 (see Appendix C.2.2 for other models), where the Averaged Overlaps from token-based methods are consistently high, causing that better classification performance cannot be achieved on such methods, which confirms our hypothesis in §5.3.1. Meanwhile, the overlaps from Hidden Calibration is much less than from token-based methods, meaning that Hidden Calibration produces better classification criteria with better *possible* classification performance than the token-based methods, even if delicate calibrations transfer or rotate these classification boundaries.

5.5.2 Principle: The Inner Linear-separability

In the practice of Hidden Calibration, simple linear boundaries are used to classify ICL examples, raising curious on the linear separability of hidden states. In this section, We find that LMs primarily produce linearly separable hidden state clusters corresponding to the ground-truth label, and the demonstrations facilitate this process.

As an intuitive visualization, we plot curves the same as the Fig. 5.7 but with various numbers of demonstrations k to visualize the *clustering dynamics* of hidden states in Fig. 5.12, where we find that: **(1)** the data points have a little linear separability when $k = 0$, and **(2)** such linear separability is being enhanced among

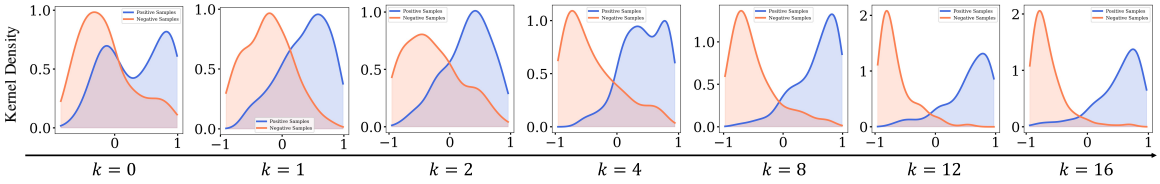


Figure 5.12: ICL hidden states clustering dynamics visualized on OPT-2.7B and `SemEval 2014-Task 4 Laptops` (Pontiki et al., 2014). The densities of data points appear in clusters responding to their query labels originally when no demonstrations are given, and gradually converge to the centroid w.r.t. the demonstrations number (k).

the increment of k, performing increasing intra-class converging tendency.

We further characterize this process. First, we calculate the Averaged Overlap similar to §5.5.1 against k in Fig. 5.9. We find that the token-based overlaps remain high and stable w.r.t. k, which indicates that the token-based methods can not benefit much from the demonstrations. However, the overlaps from Hidden Calibration significantly decrease with the increase of k, indicating that Hidden Calibration benefits from the demonstrations as expected, aligning with our observations in §5.4.2.

More generally, we visualize the distribution of the last hidden states from similar inputs of Fig. 5.12 on the **second** principal components of hidden states to get an essential observation in Fig. 5.10, where as k increases, the hidden state shows more clear intra-class clustering, enabling separability through a linear boundary.

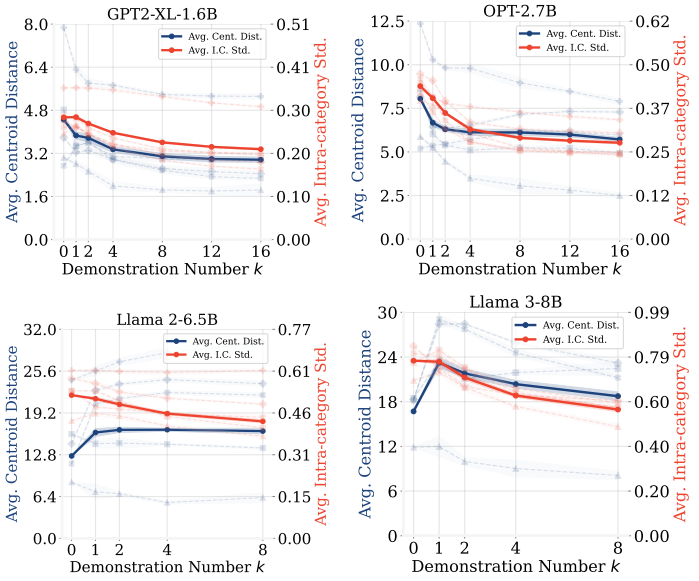


Figure 5.11: The averaged intra-class standard error of data points and the inter-class averaged centroid distance against k. **Solid curves**: means on 5 datasets; **Dashed and pale curves**: Individual results for each dataset.

More directly, on the last hidden states, we measure the intra-class standard error and the inter-class averaged centroid distance against k (see Appendix C.1.6 for details), both are a first-order moment for a joint measurement of intra-class clustering and inter-class clustering. The results are shown in Fig. 5.11, where the two curves are both diminishing, showing an obvious intra- and inter-class clustering trend w.r.t. k. However, the inter-class clustering has weaker and less persistent decreasing trends, presenting only in early demonstrations, or even ascending, which indicates that demonstration enhances intra-class clustering stronger than the inter-class clustering, which is beneficial to linear classification. Moreover, a model with more parameters shows a stronger difference between these clustering.

5.6 Related Works

Given the topic of enhancing in-context learning, we classify the literature into 3 categories.

Model parameter update-based method: Although it is pointed out that the ICL objective is implicitly included in pre-training data (Han et al., 2023b), explicitly fixing the gap between the ICL objective and causal language modeling objective can still be beneficial. Such methods are usually based on supervised fine-tuning (Min et al., 2022b; Gu et al., 2023b; Wei et al., 2021; 2023a; Iyer et al., 2022; Wang et al., 2022), and also self-supervised training (Chen et al., 2022) and non-gradient method (Zhao et al., 2024). Such methods usually require huge amounts of computation and data overhead to update billions of LM parameters.

In contrast, lightweight solutions focus on **classification criteria-based method (calibration)**. Such methods focus on re-calculating output label probabilities, keeping the main feed-forward calculation processes and model parameters un-modified. The original motivation for these works is to eliminate prior bias and unfaithful confidence in ICL, by calibrating the output label probabilities (Holtzman et al., 2021; Shi et al., 2022; Fei et al., 2023; Zhao et al., 2021; Han et al., 2023c; Zhou et al., 2024a; Jiang et al., 2023). While, as described in the main text, some practices without the usage of label-specific probabilities have also been proposed (Xu et al., 2022; Abbas et al., 2024; Min et al., 2022a).

Also, a careful **design of input prompts** can help improve the ICL performance. **(1) Demonstration selection.** Gonen et al. (2023) finds that selecting the demonstrations with lower perplexity improves the ICL performance, similarly, Kim et al. (2022) generate the demonstrations from pre-trained LMs, etc. **(2) Demonstration ordering.** It is found that the ordering of demonstrations can significantly influence the performance (Lu et al., 2022; Liu et al., 2024; Xu et al., 2024b), as also shown in our experiments in Fig. 5.5. Specifically, Lu et al. (2022) detect the optimal demonstration ordering by some synthetic detecting sequences, while Liu et al. (2024) orders the demonstrations from easy to hard, following a curriculum learning form.

5.7 Discussion

Conclusion. In this chapter, we analyze the current token-based ICL decision boundaries and point out a limitation of using token probability for ICL prediction. To address such a drawback, We propose Hidden Calibration by decoding the classification logits from centroid classifiers on LM's last hidden states. Our experiments show that Hidden Calibration is a new state-of-the-art of ICL, with high efficiency on time & space, data, and prompt engineering. Then, we confirm that Hidden Calibration indeed creates better classification logits by reducing the inter-class overlap. Moreover, we discover the hidden state convergence promoted by demonstrations, as an explanation of the principle of the performance improvement by a single linear classification boundary in Hidden Calibration. We hope this work can inspire exploration of the ICL by investigating the hidden state instead of token probabilities, and update the community's understanding of ICL calibration.

Comparison to Previous Works. (1) Comparison to Probe Methods. One concern is that our work can be regarded as a degraded linear probe (Abbas et al., 2024) of the hidden states. However, we believe our work has more advantages: In terms of application, we use fewer samples and require no gradient-based training, which makes our method more user-friendly, efficient, elegant, and interpretable. Moreover, compared to fitting a universal approximation (Hornik et al., 1989), our method and settings fully utilize the hidden state convergence on decoder LMs (described in §5.5.2), making it a true ICL practice. **(2) Comparison to Supervised Fine-tuning.** Some practices (Gu et al., 2023b; Min et al., 2022b) build training objectives to fine-tune models for better ICL performance. These efforts are efficient but costly, while our work avoids such an enormous overhead, making it more usable and elegant. **(3) Comparison to Other Calibrations.** Our method can be seen as a disruptive innovation for methods based on token probability (even the ones based on the whole vocabulary). Experimental comparisons of these methods have been given throughout this chapter.

5.8 Limitations

Due to computability limitations, we are not able to compare the performance of Hidden Calibration with the baseline based on supervised fine-tuning. However, we believe that Hidden Calibration is not within the same methodology as the fine-tuning method, due to the significant difference in computational cost. So such a lack of comparison will not seriously hurt the soundness of this work.

We argue that human intuition in the label token choice is not reliable. However, we have not eliminated such human intuition completely from the ICL loop: when we build prompts, we still choose the label token. How to automatically select the optimal label token in the prompt will be an important issue, leaving it as a future research direction for improving the performance of ICL further.

Other label probability calibrations (e.g. Batch Calibration) can be combined with Hidden Calibration for further performance improvements, since the 0-point is not necessarily an exact classification boundary, as shown in Fig. 5.12. Also, more complex prompts can be used. However, due to space constraints, we have

not attempted this incremental approach, remaining it for an empirically possible practice.

Observation in §5.5.2 needs more theoretical and experimental analysis. As we can see, some models (GPT 2-XL) do not benefit from demonstrations even through the lens of hidden state clustering or Hidden Calibration, which needs to be explained. An explanation of "why such clustering occurs or not", and "how to enhance the intra-class clustering by fine-tuning or prompt engineering" will be considerably beneficial for understanding ICL.

Chapter 6

Low-resource Attention Behavior Fine-tuning for ICL

Mechanistic Fine-tuning for In-context Learning

This chapter mainly retells the paper "Mechanistic Fine-tuning for In-context Learning" in the EMNLP 2025 BlackBox Workshop paper by Yufeng Zhao (alias: Hakaze Cho) (refer to Publication List).

The connections from the previous chapters: In Chapter 3, we raised that induction heads attending to wrong label tokens could reduce ICL performance, therefore, in this chapter, we try to suppress such wrong induction, and promote the correct induction.

Abstract

In-**c**ontext **L**earning (ICL) utilizes structured demonstration-query inputs to induce few-shot learning on **L**anguage **M**odels (LMs), which are not originally pre-trained on ICL-style data. To bridge the gap between ICL and pre-training, some approaches fine-tune LMs on large ICL-style datasets by an end-to-end paradigm with massive computational costs. To reduce such costs, in this chapter, we propose **A**ttention **B**ehavior **F**ine-**T**uning (ABFT), utilizing the previous findings on the inner mechanism of ICL, building training objectives on the attention scores instead of the final outputs, to force the attention scores to focus on the correct label tokens presented in the context and mitigate attention scores from the wrong label tokens. Our experiments on 9 modern LMs and 8 datasets empirically find that ABFT outperforms in performance, robustness, unbiasedness, and efficiency, with only around 0.01% data cost compared to the previous methods. Moreover, our subsequent analysis finds that the end-to-end training objective contains the ABFT objective, suggesting the implicit bias of ICL-style data to the emergence of induction heads. Our work demonstrates the possibility of controlling specific module sequences within LMs to improve their behavior, opening up the future application of mechanistic interpretability.

6.1 Introduction

In-**C**ontext **L**earning (ICL) (Radford et al., 2019; Dong et al., 2022) is an emerging few-shot learning paradigm where only a concatenation of few-shot *demonstrations* and a *query* is needed to conduct the specified task on the query, requiring only feed-forward calculation on the pre-trained **L**anguage **M**odels (LMs), as shown in Fig. 6.1 (A, B). However, trained on natural language data, LMs may face a distribution gap with ICL-style inputs, potentially hindering ICL performance. Therefore, some prior studies (see §6.2) try to bridge such a gap by fine-tuning LMs on the ICL-style data on end-to-end paradigms, with enormous datasets and calculation cost, preventing practical application, especially on the scaling **L**arge **LM**s (LLMs).

Therefore, in this chapter, we try to propose an efficient fine-tuning approach towards better ICL performance, utilizing some previous observations on the inner mechanisms of ICL. In detail, we focus on the *Induction Heads* in Transformer-based LMs, which are a set of critical attention heads towards ICL, where the attention scores of the last token in the ICL input (where the predictions are generated) are dominant on the label tokens in the demonstrations, as shown in Fig. 6.1 (C), for the clue that the tendency of attention scores from induction heads influences the tendency of prediction synchronously (Reddy, 2024; Cho et al., 2025a) (e.g., if the attention scores of the induction heads focus on the label token "negative" in the context, then the prediction is biased towards "negative").

Consequently, we can directly control the attention scores to make the induction heads focus on the correct label tokens for correct predictions. Given such an objective, as shown in Fig. 6.1 (C, D), we propose

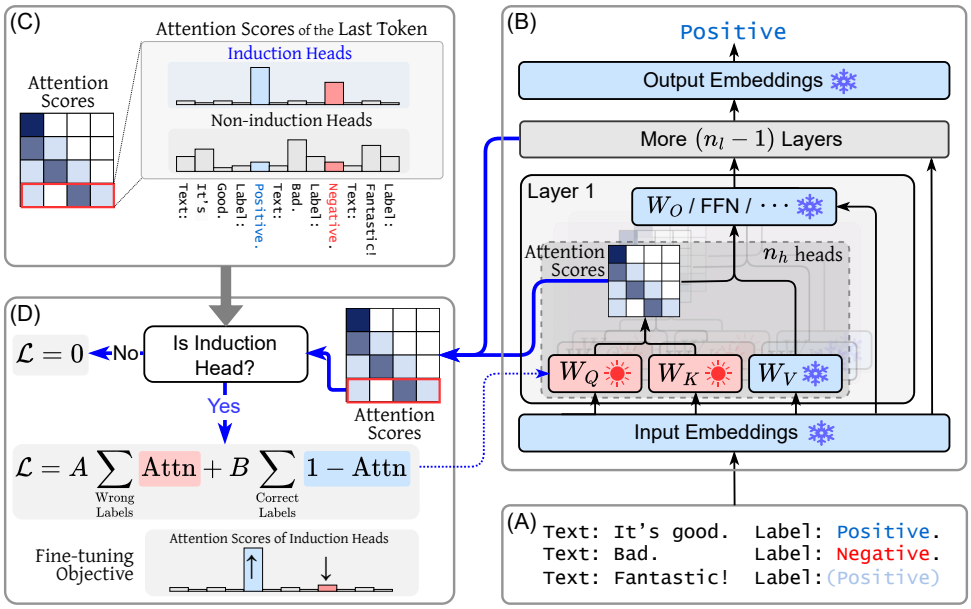


Figure 6.1: Diagram of ABFT framework. (A) **An example of ICL-style inputs.** We build datasets from such examples to fine-tune models. (B) **Feed-forward inference of ICL.** We collect the attention scores of every attention head in every layer to calculate the training objective. and we only enable the gradient of the W_Q and W_K matrices. (C) **The criterion for induction head.** Only attention heads producing attention scores with a significant focus on the label tokens can be identified as induction heads. (D) **Loss calculation of ABFT.** Only induction heads return a non-zero loss, and such loss contains a punishment on "wrong" attention scores to wrong label tokens, and a reward on "correct" attention scores to correct label tokens.

Attention **B**ehavior **F**ine-**T**uning (ABFT), **calculating fine-tuning objective (loss function) only on the attention scores of induction heads**, to mitigate "wrong" attention score focusing on the wrong label tokens, and promote "correct" attention score focusing on the correct label tokens. On such an objective, we fine-tune only the W_K and W_Q projection matrices of every attention head, with an ICL-style training set of only a few hundred samples, and only a few million of the parameters with gradient activated, which is highly efficient compared to previous works.

Our experiments on 9 modern (L)LMs and 8 downstream datasets demonstrate that ABFT significantly improves ICL performance with satisfactory efficiency, robustness, unbiasedness, and harmlessness, even outperforming previous works of end-to-end fine-tuning the whole model on massive datasets that are approximately $7,000\times$ larger than ours. Moreover, our analysis finds that the ABFT objective is an implicit bias of direct end-to-end training objective on ICL-style data, suggesting that the causal language modeling on the ICL-style data may naturally evoke the emergence of induction heads.

Our contribution can be summarized as:

- We propose **A**ttention **B**ehavior **F**ine-**T**uning (ABFT), which efficiently fine-tunes LMs on ICL inputs using attention-based objectives without supervision on the final output.
- Subsequent analysis indicates that the training objective of ABFT is implicitly encompassed by the end-to-end training objective on ICL-style data, suggesting that these data may naturally evoke the induction heads, which enhances the previous works on the inner mechanism of ICL.
- Also, we prototypically confirm the possibility of optimizing model performance directly by controlling the intermediate behavior, without any error propagation from the output. This is a hint toward *Mechanistic Controllability*, a valuable future of mechanistic interpretability.

6.2 Background & Related Works

In-context Learning. Given a few-shot *demonstration* set $\{(x_i, y_i)\}_{i=1}^k$ and a query x_q, typical ICL creates a concatenation formed like $[x_1, y_1, x_2, y_2, \ldots, x_k, y_k, x_q]$, and feeds it into the forward calculation of a pre-

trained LM (Radford et al., 2019; Dong et al., 2022) for the next token as the prediction to x_q, as shown in Fig. 6.1 (A).

LM Warm-up for ICL. Since LMs are typically pre-trained on plain natural language data instead of ICL-style data, it can be expected that a distribution gap between the pre-training and ICL testing occurs to prevent optimal performance. Therefore, some works focus on tuning LMs on the ICL data (Chen et al., 2022; Min et al., 2022b; Mishra et al., 2022; Wang et al., 2022; Wei et al., 2023a). Even effective, these works need gradient-based whole-model and full-precision training on large datasets, making it hard to adapt to real-world applications due to the calculation overhead, and misaligning with the low-resource purpose of ICL.

Induction Heads in ICL Inference. As shown in Fig. 6.1 (C), it has been found that some attention heads (called *induction heads*) in LMs have a nontrivial influence on ICL inference (Elhage et al., 2021; Olsson et al., 2022; Singh et al., 2024b; Reddy, 2024; Cho et al., 2025a), where the attention scores from the last token of the query (the location for prediction, e.g., the last ": " in Fig. 6.1 (A), as the attention query) concentrate on the label tokens presented in the demonstrations (e.g., "positive"s and "negative"s in Fig. 6.1 (A), as the attention key). Attention connections from induction heads transfer label information from the demonstration to the output, biasing predictions toward labels with higher attention scores. Consequently, the accuracy of ICL prediction critically depends on whether these attentions are on the correct labels.

6.3 Attention Behavior Fine-tuning

Given the inspiration from the previous works, where the ICL predictions are biased towards the more attention-score-concentrated labels in the induction heads, in this chapter, as shown in Fig. 6.1, we propose **A**ttention **B**ehavior **F**ine-**T**uning (ABFT), a novel low-resource fine-tuning method to induce attention scores to focus on the correct labels.

Method Pipeline. Globally, ABFT utilizes such a pipeline: **(1) Dataset Building**: from a selected downstream dataset, we build a training set composed of ICL-style sequences as shown in Fig. 6.1 (A). **(2) Feed-forward Calculation**: For each training sample, as shown in Fig. 6.1 (B), we conduct a standard feed-forward calculation on the pre-trained LM, and collect the attention matrices of all the attention heads in all the layers. **(3) Loss Calculation**: For each attention matrix, we only focus on the attention scores of the last token (i.e., the last row of the attention matrix), where the predictions of queries are made. As shown in Fig. 6.1 (D), we first filter (detailed below) the non-induction head out, and return a loss of 0 for these heads. Then, for the remaining induction heads, we calculate a loss function composed of a punishment of attention scores on wrong labels and a reward of attention scores on correct labels (detailed below). **(4) Back Propagation**: We back-propagate the calculated loss only to the W_Q and W_K matrices of every attention head, and update the model parameters.

Induction Head Filter. As shown in Fig. 6.1 (C), we skip the attention matrices where the attention scores of the last token do not dominate on the label tokens. To identify the attention matrices to skip, in detail, given an attention matrix $\mathcal{A} \in \mathbb{R}^{n_t \times n_t}$, where the n_t is the input token sequence length, as mentioned before, we focus on the last row $\alpha = \mathcal{A}_{n_t}$. Given the position index of label tokens as $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^{k}$, we calculate the attention score sum on these label tokens as $S = \sum_{j \in \mathcal{I}} \alpha_j$.[18] Then, we set a threshold $T = \frac{k}{k+\log(n_t)}$, if $S > T$, we assert the attention head of score $\mathcal{A}$ is an induction head, and vice versa. We will verify the necessity and benefits of this induction head filter in §6.5.2.

Loss Function. As shown in Fig. 6.1 (D), given an attention matrix $\mathcal{A}$, if judged as a non-induction head by the aforementioned head filter, the loss $\mathcal{L}$ for $\mathcal{A}$ is assigned to 0. Else, we conduct the following calculation: given the position index of label tokens consistent with the ground-truth label of the query as $\mathcal{I}^+$, and the others $\mathcal{I}^- = \mathcal{I}n\mathcal{I}^+$,[19] we calculate the loss from the last row (α) of $\mathcal{A}$ as:

$$\mathcal{L}(\mathcal{A}) = A \sum_{i \in \mathcal{I}^-} \alpha_i + B \sum_{i \in \mathcal{I}^+} 1 - \alpha_i \,. \tag{6.1}$$

That is, as shown in Fig. 6.1 (D), we punish the "wrong" attention scores towards the label tokens different

[18] For the case shown in Fig. 6.1 (C), $\mathcal{I} = \{4, 8\}$ (0-started), and S is the sum of the values of the red-highlighted bar and the blue-highlighted bar.

[19] For the case shown in Fig. 6.1 (C), $\mathcal{I}^+ = \{4\}$ (the position of "positive"), $\mathcal{I}^- = \{8\}$ (the position of "negative").

Table 6.1: Accuracies (%) of ABFT and baselines on 9 LMs and 8 datasets. The best results are in **bold**.

Model	Param.	Method	SST2	MR	FP	SST5	TREC	SUBJ	TEE	TEH	Average
GPT 2-L	812M	Vanilla	56.35	61.13	42.09	29.69	35.45	49.12	39.06	54.00	45.86
		MetaICL	85.94	80.96	37.30	**42.09**	33.98	50.49	45.41	54.20	53.80
		PICL	74.70	73.34	54.49	33.79	32.91	51.37	**47.46**	53.42	52.68
		Calibrate	56.35	60.94	36.91	25.10	34.08	49.32	36.72	53.52	44.12
		ABFT	**88.18**$_{1.54}$	**85.40**$_{0.95}$	**81.30**$_{1.69}$	36.84$_{3.61}$	**50.24**$_{2.49}$	**61.99**$_{4.39}$	46.51$_{3.11}$	**55.20**$_{0.20}$	**63.21**
GPT 2-XL	1.61B	Vanilla	67.87	69.53	51.07	30.66	35.25	50.98	42.68	53.03	50.13
		PICL	74.80	74.32	51.17	32.71	33.20	51.46	47.95	53.42	52.38
		Calibrate	68.16	75.00	36.43	28.52	35.55	50.10	39.26	51.56	48.07
		ABFT	**87.92**$_{1.47}$	**86.52**$_{1.50}$	**87.67**$_{0.45}$	**37.55**$_{2.67}$	**51.83**$_{2.73}$	**75.07**$_{2.96}$	**60.01**$_{5.38}$	**55.35**$_{0.04}$	**67.74**
Falcon 3	7.46B	Vanilla	91.11	92.77	85.35	46.00	50.00	62.60	60.55	52.05	67.55
		Calibrate	90.53	**93.07**	82.71	44.04	54.30	62.40	54.79	51.76	66.70
		ABFT	**92.14**$_{0.21}$	92.17$_{0.04}$	**96.14**$_{0.36}$	**47.32**$_{0.16}$	**75.81**$_{0.19}$	**94.87**$_{0.82}$	**67.97**$_{0.25}$	**70.34**$_{0.22}$	**79.59**
Llama 3	8.03B	Vanilla	89.35	92.87	75.78	44.24	55.76	62.30	57.91	54.59	66.60
		Calibrate	90.04	**93.36**	43.95	41.60	54.39	65.23	54.79	52.83	62.02
		ABFT	**93.14**$_{1.39}$	92.50$_{0.84}$	**94.02**$_{1.72}$	**52.10**$_{2.31}$	**73.09**$_{1.11}$	**92.70**$_{1.44}$	**72.02**$_{1.81}$	**72.02**$_{5.87}$	**80.20**
DeepSeek-R1 Dist. Qwen `4-bit`, LoRA	14.8B	Vanilla	90.92	91.21	92.18	**46.97**	62.50	66.60	66.60	63.87	72.61
		Calibrate	90.04	91.41	**92.68**	46.09	61.62	65.43	65.33	62.30	71.86
		ABFT	**93.51**$_{1.22}$	**91.85**$_{0.34}$	91.17$_{4.44}$	46.09$_{1.27}$	**69.14**$_{2.54}$	**92.92**$_{3.27}$	**69.82**$_{0.00}$	**71.19**$_{3.42}$	**78.21**
Qwen 2.5 `4-bit`, LoRA	32.8B	Vanilla	93.85	94.43	86.23	47.17	58.40	87.50	65.14	69.63	75.29
		Calibrate	93.75	94.82	74.22	44.82	58.79	84.08	63.96	63.96	72.30
		ABFT	**94.92**$_{0.00}$	**94.83**$_{0.10}$	**94.04**$_{0.00}$	**48.49**$_{0.05}$	**69.24**$_{0.10}$	**96.00**$_{0.00}$	**69.29**$_{0.24}$	**70.31**$_{0.00}$	**79.64**
SimpleScaling s1.1 `4-bit`, LoRA	32.8B	Vanilla	94.82	94.24	91.11	**50.20**	69.63	89.65	68.36	72.17	78.77
		Calibrate	94.43	93.85	88.96	48.63	68.07	89.26	68.55	72.36	78.02
		ABFT	**94.92**$_{0.10}$	**94.29**$_{0.05}$	**96.00**$_{0.00}$	49.95$_{0.05}$	**72.46**$_{0.39}$	**95.71**$_{0.10}$	**71.73**$_{0.05}$	**73.10**$_{0.04}$	**81.02**
Llama 3 `4-bit`, LoRA	43.2B	Vanilla	93.26	94.04	73.92	**49.41**	58.98	71.58	62.60	66.70	71.31
		Calibrate	**95.02**	93.07	54.20	44.53	59.08	72.56	61.03	65.82	68.16
		ABFT	**95.02**$_{0.10}$	**93.85**$_{0.10}$	**94.87**$_{0.05}$	48.10$_{0.14}$	**64.70**$_{0.14}$	**90.09**$_{0.05}$	**69.24**$_{0.29}$	**70.85**$_{0.15}$	**78.34**
Llama 3 `4-bit`, LoRA	55.6B	Vanilla	93.94	92.19	78.81	**51.37**	67.19	66.70	56.44	60.94	70.95
		Calibrate	92.29	92.77	69.92	50.49	68.75	65.92	58.11	62.70	70.12
		ABFT	**94.53**$_{0.10}$	**93.41**$_{0.14}$	**93.21**$_{0.44}$	49.02$_{0.78}$	**71.78**$_{0.58}$	**92.68**$_{0.48}$	**70.76**$_{0.64}$	**70.75**$_{0.34}$	**79.52**

from the query's ground-truth with magnitude $A \geqslant 0$, and reward the "correct" attention scores with magnitude $B \geqslant 0$. These two terms in the loss function may seem redundant, but we will demonstrate in §6.5.3 that they actually contain antagonistic implicit biases, therefore, the factors A and B should be balanced well.

Why not End-to-end LoRA? Intuitively, directly adding LoRA bypasses (Hu et al., 2022) to the trained projection matrices and fine-tuning them on an end-to-end training objective is also a possible approach. However, in end-to-end LoRA, gradients are propagated from the output logits of LMs, which requires that the final layer of the LMs (i.e., output embeddings, or LM Head) must be in full precision and with gradients activated, to get stable gradients into the residual stream. This introduces a non-negligible overhead, an issue avoided by ABFT as it does not supervise the final output. Moreover, fine-tuning attention projections without selectivity may cause harmful side effects toward the ICL out of the fine-tuned domain. We will compare the performance and efficiency of ABFT against end-to-end LoRA in the following experiments (Table 6.2) to highlight the efficiency and harmlessness of the ABFT method.

6.4 Main Experiments

In this section, we mainly confirm the effectiveness of the proposed ABFT, and find that: ABFT effectively improves the ICL performance to about 10%∼20% relatively, which requires the minimum parameters less than 0.05% to be full precision and gradient, with other parameters free to be quantized and gradient-free, and utilize 0.01% data cost compared to the previous works.

6.4.1 Experiment Settings

Models and Datasets. We conduct our experiment on **9 modern LLMs**: GPT 2 (Large, XL) (Radford et al., 2019), Falcon 3-7B (Team, 2024b), Llama 3 (8B, 43B, 56B) (AI@Meta, 2024), DeepSeek-R1 Distill Qwen-14B (DeepSeek-AI, 2025), Qwen 2.5-32B (Team, 2024a; Yang et al., 2024), and SimpleScaling s1.1-32B (Muennighoff et al., 2025); and **8 datasets**: SST-2, SST-5 (Socher et al., 2013), MR (Pang & Lee, 2005), **F**inancial **P**hrasebank (Malo et al., 2014), TREC (Li & Roth, 2002; Hovy et al., 2001), **Subj**ective (Wang

& Manning, 2012), **T**weet **E**val **E**motion (Mohammad et al., 2018), **T**weet **E**val **H**ate (Basile et al., 2019) (Refer Appendix D.1.1 for details).

Hyperparameters. We set: training samples $n_d = 512$, the number of demonstrations per ICL sample $k = 4$. A standard Adam optimizer (Kingma & Ba, 2014) is used with learning rate $\mathrm{lr} = 2 \times 10^{-5}$ and pseudo-batch-size $n_b = 32$ (i.e., we average gradients per $n_b = 32$ samples before performing a single gradient step). We set the initial values $A_0 = 0.5$, $B_0 = 1.0$, and dynamically balance them with the PID algorithm (refer to Appendix D.1.3), stabilizing the number of attention heads identified as induction heads (see §6.5.3). The models are trained for $n_{\text{step}} = 32$ steps.

Quantization Settings. Models over 10B are quantized to `4-bit`, with full-precision LoRA (Hu et al., 2022) (inner dimension $r = 16$) trained on W_Q and W_K with learning rate 10^{-4}.

Baselines. We compare with: Contextual Calibration with 512 training samples (Zhao et al., 2021), MetaICL end-to-end fine-tuning with 3.55M samples (Min et al., 2022b), and PICL re-pre-training with 80M samples (Gu et al., 2023a).

Others. We conduct all the experiments on a single NVIDIA A40 with 48GB VRAM. We repeat each experiment 4 times ($\leqslant$10B) or 2 times ($>$10B), and report the averaged results on 1024 fixed test inputs for each dataset. ICL-style inputs are built with library STAICC (Cho & Inoue, 2025).

6.4.2 Main Results

The test results are shown in Table 6.1, where ABFT consistently outperforms all the baselines, even with enormous training sets (to MetaICL, $3.55\text{M}/512 \approx 7000\times$) and full-model fine-tuning (remind that ABFT only focuses on the W_Q and W_K matrices), suggesting that ABFT is satisfyingly efficient in both time and data cost. Such results also provide strong empirical evidence for the effectiveness of induction heads in LLMs.

Towards Mechanistic Controllability. To the best of our knowledge, ABFT is the first approach to train models without accessing final outputs, enabling model controlling via intermediate features or activations. Through this practice, we prototypically implement one of the visions of mechanistic interpretability (Rai et al., 2024): by attributing the model's inference to specific modules (*circuits*), we enable their local optimization, thereby improving overall performance effectively and efficiently.

Table 6.2: A comparison between ABFT and End-to-end Fine-tuning (E2E FT). Param.*: Parameters which are required in `FP16/32` and with gradient on.

Model	Method	Param.*	Time	Acc_{ID}	Acc_{OD}
Llama 3 8.03B `4-bit`, LoRA	Vanilla	-	-	66.02	
	E2E FT	0.5B	2.2×	78.33	61.74
	ABFT	6.8M	1×	72.54	64.34
DeepSeek-R1 14.8B `4-bit`, LoRA	Vanilla	-	-	72.61	
	E2E FT	0.8B	2.2×	78.26	63.62
	ABFT	12M	1×	78.21	67.21
Qwen 2.5 32.8B `4-bit`, LoRA	Vanilla	-	-	75.29	
	E2E FT	0.8B	2.6×	82.09	62.24
	ABFT	17M	1×	79.64	64.96
Llama 3 55.6B `4-bit`, LoRA	Vanilla	-	-	70.95	
	E2E FT	1.1B	2.7×	82.80	64.86
	ABFT	33M	1×	79.52	67.32

Table 6.3: Prediction consistency metrics (%) on each models averaged among 8 datasets.

Model	Template Consist.		Demonstration Consist.	
	w/o ABFT	w/ ABFT	w/o ABFT	w/ ABFT
GPT 2-XL	81.28	**91.74**	68.38	**82.75**
Llama 3-8B	86.93	**90.32**	76.99	**92.00**
DeepSeek-R1-14B	89.64	**92.79**	81.30	**85.97**
Qwen 2.5-32B	88.97	**92.78**	84.52	**87.94**
Llama 3-56B	92.78	**93.90**	82.10	**87.49**

Prediction Consistency. We evaluate the prediction consistency against variations in (1) prompt templates and (2) demonstration sampling on STAICC-DIAG (Cho & Inoue, 2025). For each query, we repeat predictions across different prompt templates and sampling strategies, and measure consistency as the ratio of the maximum consistent predictions (e.g., 6 positive vs. 3 negative yields $6/9 = 2/3$), averaged over the dataset; see Cho & Inoue (2025) and Appendix D.1.5 for implementation details. As shown in Table 6.3, ABFT significantly improves consistency across all 8 datasets, stabilizing ICL under diverse contexts and enhancing prompt design efficiency.

Prediction Bias against Wrong Labels. Moreover, a known concern in ICL is the bias toward seen labels when ground-truth labels are absent in demonstrations (i.e., $\mathcal{I}^+ = \emptyset$) (Zhao et al., 2021; Cho et al., 2025a), which can lead to incorrect predictions. Our testing on such scenario with and without ABFT in Fig. 6.3 shows that (see Appendix D.1.4 for experiment details): ABFT mitigates this issue via the punish term A, which penalizes incorrect labels during training and reduces the bias effects of induction heads. Notably, ABFT outperforms the 0-shot setting under unseen-label, suggesting the existence or emergence of unknown

mechanisms that enable demonstrations in other categories to enhance ICL[20].

6.4.3 Comparison against End-to-end Fine-tuning

As mentioned before, end-to-end (E2E) fine-tuning on the in-domain dataset (not a wide dataset like MetaICL) with LoRA is also an alternative solution. However, in this section, we will show that compared with E2E fine-tuning, ABFT is more efficient, and more harmless on tasks out of the fine-tuning domain.

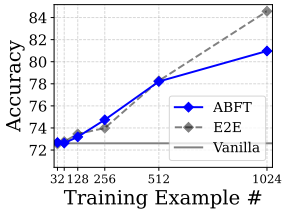


Figure 6.2: Data efficiency: accuracy against data size on DeepSeek-R1 Distill Qwen-14B / 8 datasets.

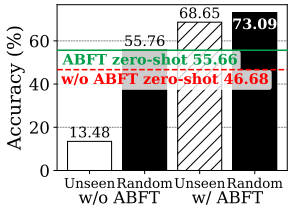


Figure 6.3: Accuracy on unseen label inputs and randomly sampled inputs, w/ and w/o ABFT.

Time and Memory Cost. Notice that in the E2E scenario, since the gradient is propagated from the final output of the model, the LM head, which is the top of the model, should be in full precision, to ensure a sufficient numerical resolution to utilize the mini-batch for mitigating the gradient noise in the stochastic gradient descent (Hubara et al., 2016). This introduces a non-negligible overhead, as measured in Table 6.2. E2E fine-tuning slightly outperforms ABFT in in-domain accuracy (ACC_{ID}), but incurs substantial training time and memory costs.

Harmlessness. We evaluate **o**ut-of-**d**omain (OD) performance on datasets different from the fine-tuned one (Table 6.2, ACC_{OD}). Both ABFT and E2E fine-tuning degrade OD performance, but ABFT causes less harm. This supports a conclusion from Cho et al. (2025a): some induction heads are intrinsic and task-independent, while others are task-induced. ABFT on intrinsic heads harms OD performance, whereas ABFT on task-induced heads does not. In contrast, E2E fine-tuning on all heads broadly degrades OD performance.

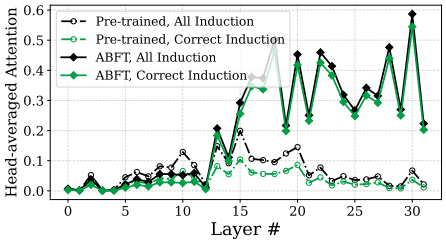


Figure 6.4: Induction attention scores and correct (on the correct labels) induction attention score averaged among heads on each layer. **ABFT enhances induction attention, and improves the correctness**.

Data Efficiency. We test the accuracy against the training set size as a metric of data efficiency, for both ABFT and E2E fine-tuning, as shown in Fig. 6.2 (refer to Appendix D.3.1 for results on other models). In the results, ABFT and E2E fine-tuning consistently benefit from more data samples, and in few-shot scenarios ($\leqslant$ 512), ABFT and E2E fine-tuning act equally, while E2E fine-tuning acts better when more training data is given. However, given the low-resource objective of ICL, and also the far more expensive time and memory cost of E2E fine-tuning, we can claim that ABFT has an advantage in the few-shot and low-resource scenario.

6.5 Analysis

6.5.1 Attention Visualization after ABFT

As shown in Fig. 6.4, we average the global (to $\mathcal{I}$) and correct (to $\mathcal{I}^+$) induction attention scores on the last token among attention heads and input samples on each transformer layer, on the validation set. Also, we provide a direct visualization of attention scores in Appendix D.3.3. Compared to the pre-trained model, the ABFT model tends to eliminate attention scores towards incorrect label tokens ($\mathcal{I}^-$), and shift the attention scores from the attention sinks (the first token) (Xiao et al., 2024) and plain tokens to the correct label token, causing an enhancement to induction attention scores, continuously on the

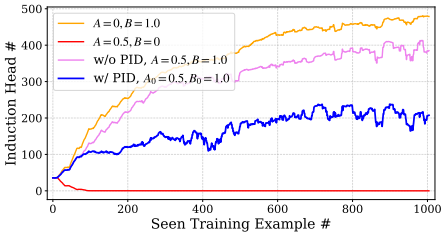


Figure 6.5: Number of attention heads that are judged as induction heads on 4 settings, against the training processing (the number of seen data samples). **PID effectively stabilizes the induction head numbers**.

[20]Since that: such a phenomenon contrasts with existing views (Cho et al., 2025a), where the explicit copying by the induction head is the only channel through which information is transferred from the demonstrations to the query (in Fig. D.1, we show that the induction heads in ABFT model are almost fully suppressed in unseen-label scenario), where unseen-label demonstrations are harmful to ICL.

middle-to-last layers. Such visualization indicates that ABFT successfully generalizes to correct the behavior of attention heads.

6.5.2 Ablation Analysis

In this section, we disable some components utilized in the ABFT training protocol to suggest their necessity. The main results of such ablation experiments are shown in Table 6.4, where:

ABFT should be Localized. In Table 6.4, disabling the head filter ((D) in Fig. 6.1) harms the accuracy. Knowing that all attention heads are trained to be induction heads under unfiltered ABFT loss, we can infer that: in LLMs, some attention heads with functions other than the induction head are still necessary for ICL, aligning with and enhancing the previous work (Cho et al., 2025a). However, considering that some implicit antagonistic effects induced by unfiltered ABFT loss still promote the formation of other essential heads (i.e., when the ABFT loss from deeper heads propagates to shallower heads, its function becomes antagonistic with the ABFT loss directly connected to those shallow heads), the accuracy degradation with no head filters is not so significant.

Loss Factor (A and B) should be Balanced. As mentioned in §6.4.1, we use the PID algorithm to adaptively balance the value of A and B in the loss calculation. In Table 6.4, we disable such adjustment and observe an accuracy drop. Especially, when we set the A or B to 0, the accuracy significantly degrades. This eliminates the doubt of "whether the loss factors are redundant" raised in §6.3, and we will discuss this in-depth in the next section (§6.5.3).

Table 6.4: Ablation analysis of removing some components from ABFT. Notice that the PID algorithm is to stabilize the induction head number by adjusting the factor A, so when we disable the head filter or fix the A or B, the PID algorithm naturally loses its function.

Model	Method	Time	Acc.
Falcon 3 7.46B	Vanilla	-	67.55
	ABFT	1×	**79.59**
	w/o PID, $A = 0.5, B = 1.0$	1.0×	76.86
	w/o PID, w/o Head Filter	1.3×	75.57
	w/o PID, $A = 0, B = 1.0$	1.0×	72.13
	w/o PID, $A = 0.5, B = 0$	0.6×	56.47
Llama 3 8.03B	Vanilla	-	66.60
	ABFT	1×	**80.20**
	w/o PID, $A = 0.5, B = 1.0$	1.1×	80.07
	w/o PID, w/o Head Filter	1.2×	70.39
	w/o PID, $A = 0, B = 1.0$	1.2×	63.54
	w/o PID, $A = 0.5, B = 0$	0.6×	58.79
DeepSeek-R1 14.8B Dist. Qwen `4-bit`, LoRA	Vanilla	-	72.61
	ABFT	1×	**78.21**
	w/o PID, $A = 0.5, B = 1.0$	1.0×	73.35
	w/o PID, w/o Head Filter	2.1×	73.51
	w/o PID, $A = 0, B = 1.0$	0.9×	72.71
	w/o PID, $A = 0.5, B = 0$	0.9×	73.36

6.5.3 Balance the Loss Factor A and B

Punish and Reward Influence Induction Heads Antagonisticly. As shown in Table 6.4, removing either loss term in Eq. 6.1 degrades performance, indicating that both are essential. Interestingly, the two terms introduce antagonistic implicit biases: the punish term A disperses attention across labels, reducing induction heads, while the reward term B concentrates attention on specific labels, increasing induction heads through a stepwise positive feedback loop. We track the number of induction heads during training on Llama 3-8B and SST-2 (see Appendix D.3.2 for more cases), as shown in Fig. 6.5, to support this observation.

Automatically Stabilizing Induction Head Number. Ablation studies reveal that too many induction heads hinder fine-tuning and overall performance, as other functional heads are also needed for ICL; whereas an insufficient number prevents the model from handling ICL tasks. To maintain a stable number, we automatically adjust the antagonistic factor A in Eq. 6.1 using a classical PID controller (Appendix D.1.3)[21]. As shown in Fig. 6.5, PID stabilizes induction head count, improves ABFT performance (Table 6.4), and reduces sensitivity to hyperparameters A_0 and B_0, with accuracy remaining stable across settings (Fig. 6.6).

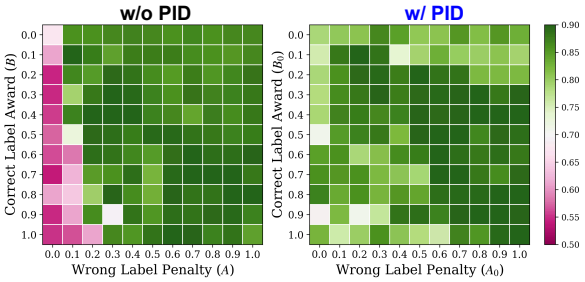


Figure 6.6: Accuracies with various settings on hyperparameter A and B, with and without PID algorithm. **PID weakens the sensitivity to initial parameters**.

[21] Since A and B are antagonistic, controlling A alone suffices to stabilize induction head numbers.

6.6 Consistency of Training Objective: ABFT and End-to-end Fine-tuning

To explore whether the emergence of induction head is from ICL-style data—a key question in interpretability (Chan et al., 2022; Reddy, 2024; Singh et al., 2024b)—we examine the consistency between ABFT and E2E objectives.

Principle. Due to the lack of qualitative thresholds, it is hard to utilize statistic-based similarity measures to determine whether two models exhibit comparable similarity sufficient to indicate consistent training objectives. Therefore, our experiment is based on such a principle: if the fine-tuning terminations on both training objectives fall into the same *basin* of the loss function, then both fine-tuning trajectories are similar (Neyshabur et al., 2020), so that the two training objectives are consistent. For such an end, we investigate the *linear-connectivity* (Neyshabur et al., 2020; Ilharco et al., 2023) among the pre-trained parameters θ_0 (as the start point of the fine-tuning) and fine-tuned parameters θ_{E} for E2E fine-tuning, and θ_{A} for ABFT. In detail, we mix these three parameters into a new model parameter set θ in the following form:

$$\theta = \theta_0 + \alpha_{\mathrm{E}}(\theta_{\mathrm{E}} - \theta_0) + \alpha_{\mathrm{A}}(\theta_{\mathrm{A}} - \theta_0), \tag{6.2}$$

and then test the accuracy of θ as an anti-metric of model loss. As shown in Fig. 6.7 (upper), if the accuracy of mixed θ is better (or at least, not significantly worse) than the accuracy of θ_{E} and θ_{A}, we can infer that the θ_{E} and θ_{A} are in the same loss basin, with linear low-loss path observed.

Experiment and Result. We conduct the aforementioned experiment protocol on SST-2 and GPT 2-XL. The results are shown in Fig. 6.7 (lower), showing no high-loss paths between the E2E fine-tuned model and the ABFT model. It suggests that they are in the same basin of the loss landscape, indicating the high similarity between these two training objectives. Moreover, we visualize the attention scores on ABFT and E2E fine-tuned models as shown in Fig. 6.8: compared with the pre-trained model, attention scores of the fine-tuned models on both objectives consistently focus on the correct label tokens, suggesting that E2E objectives imply a promotion to correct induction head. Moreover, as shown in Fig. 6.9, repeating the experiment on MetaICL shows MetaICL model lies in the same basin as ABFT, suggesting that full-model tuning on large datasets is essentially equivalent to ABFT, which can be seen as *essential ICL fine-tuning*.

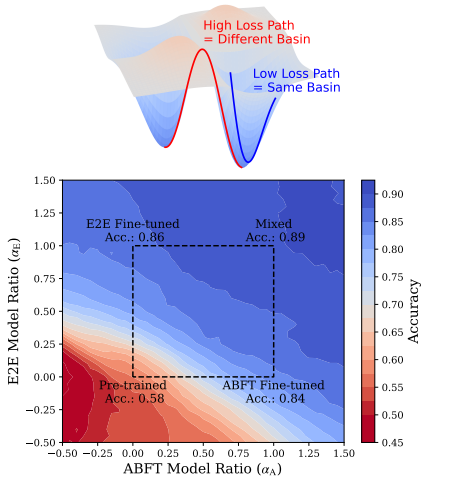


Figure 6.7: **Upper**: A better accuracy (lower loss) in the interpolation path suggests the same basin, and vice versa. **Lower**: Contour map of accuracies against the coefficient α_{E} and α_{A} in Eq. 6.2. **E2E fine-tuned and ABFT models are located in the same low-loss area.**

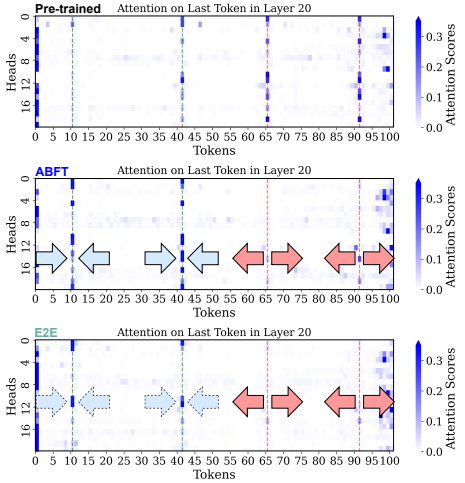


Figure 6.8: Attention score visualization of the pre-trained model, ABFT model, and E2E fine-tuned model, on the same input as Fig. D.7, and the same models as Fig. 6.7 (Refer to Appendix D.3.3 for details and more cases).

6.7 Discussion

Conclusion. In this chapter, we propose a fine-tuning objective that strengthens the correctness of the induction head by accessing only the attention matrix, and demonstrate that it significantly improves the performance of ICL. Our results reinforce the induction head hypothesis for ICL interpretability and represent a first step toward controlling model behavior through mechanistic interpretability.

Towards Mechanistic Controllability. In this chapter, we raise the possibility of controlling the model's behavior by some specific modules (often called *circuit* in the context of mechanistic interpretability), which opens up a new neural network model behavior-controlling paradigm: controlling only the modules that make significant contributions to the output, thereby substantially reducing the number of parameters that need to be adjusted and achieving excellent efficiency.

6.8 Limitations

Towards Open-end Tasks. The first limitation lies in the fact that the induction head-based explanation of ICL (Elhage et al., 2021; Singh et al., 2024b; Cho et al., 2025a), so that our proposed ABFT approach, applies only to classification tasks with a finite label set. As mentioned in §6.4.2 and Fig. 6.3, since our training objective consists of two factors, our approach is not limited to the simple retrieval setting where the ground-truth label appears in the demonstrations, while extending these methods to open-ended tasks remains an open challenge that requires further investigation on the basic mechanism. Nevertheless, given the current state of research on ICL interpretability, we have made full use of these findings and provided a valuable foundation for advancing model control through the scope of interpretability, i.e., Mechanistic Controllability.

Towards Better Mechanistic Controllability. For our vision of Mechanistic Controllability, even though this chapter successfully identifies a small set of modules (i.e., circuits) that require controlling towards better ICL performance, the control methods based on gradients and moderate amounts of data remain coarse. Therefore, future work could focus on gradient-free and data-free model editing, which directly edits some parameters utilizing a deeper understanding of the functional roles of model parameters.

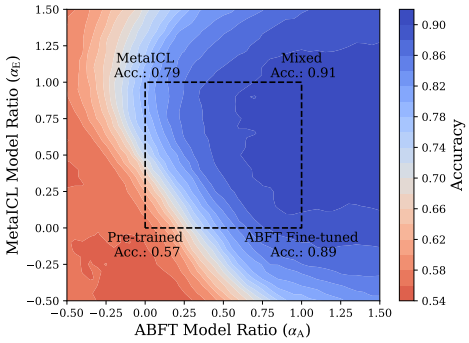


Figure 6.9: Contour map of accuracies against the α_E and α_A on GPT 2-L, and the θ_E is set as MetaICL model.

Towards Better Performance. It can be considered that some hyperparameters (see §6.4.1 and Appendix D.1.3), and the induction head filter (see §6.3) may be not optimal, restricting the performance. Discussing them in detail, and automatically optimizing them can be helpful for better performance of ABFT. Also, in Fig. 6.7, we observe that the extended line from the pre-trained model towards the ABFT model leads to better accuracy, suggesting a possibility of utilize model parameter $\theta = \theta_0 + \alpha_A(\theta_A - \theta_0), \alpha_A > 1$ to further improve accuracy without any gradient-based cost.

Towards Further Efficiency. As shown in Appendix D.2, the W_Q and W_K projections are significantly modified after ABFT only in some layers, that is, it is possible to further restrict the gradient-on parameters to some Transformer layers for better efficiency (notice that currently we activate the gradients of the attention mappings of all layers).

Part IV
Discussion

IV Discussion **55**

7 Discussion **56**
7.1 Summarization 56
7.2 Still Open Questions and Future Directions 56

Chapter 7

Discussion

7.1 Summarization

In this dissertation, we systematically investigate the inner mechanisms of ICL in LMs, and utilize these insights to improve the ICL performance. In detail, motivated by the gap between the current interpretation of ICL and the observed ICL inference phenomena, first, we develop the typical 2-step induction circuit into a 3-step framework to capture almost all the observations into one model. Then, since the proposed framework can not adapt to the unseen label scenario, we propose a new mechanism as the task-oriented information removal, serving as a supplement mechanism of induction circuits, cooperating in the typical scenario, and dominating in the unseen label scenario. Additionally, utilizing such a mechanistic understanding, we develop applicable performance improvement methods: Hidden Calibration and Attention Behavior Fine-tuning.

The contribution of this dissertation can be summarized as:

- **Mechanistic Framework of In-context Learning.** We propose a comprehensive inference circuit of ICL based on induction circuits, decomposing the process into three fundamental operations to capture most of the inference behaviors observed in ICL.
- **Mechanism of Task-oriented Information Removal.** We identify a novel mechanism, task-oriented information removal, explaining ICL performance in unseen-label scenarios. We demonstrate that LMs selectively remove redundant information from hidden states to align with the intended task, establishing a key principle, especially for unseen label generalization in ICL.
- **Hidden Calibration for Better Logits Calculation.** Utilizing the insight that hidden states contain linearly separable task-related representations, we propose Hidden Calibration, using nearest-centroid classification on the last hidden states. Experiments across multiple models and datasets show consistent and substantial performance improvements over calibration baselines.
- **Attention Behavior Fine-Tuning for Mechanistic Control.** Based on the discovered induction-circuit-based mechanism, we develop **A**ttention **B**ehavior **F**ine-**T**uning (ABFT), which optimizes attention scores instead of token probability outputs to fine-tune the model to focus on correct labels presented in the demonstrations, demonstrating that targeted control of attention behavior can directly improve ICL performance.

7.2 Still Open Questions and Future Directions

Open-end Generative Tasks. Nearly all of the aforementioned analyses have been conducted on classification tasks. While we are confident that the classification setting has been thoroughly investigated, further work is needed for open-ended tasks such as sentence translation and question answering. A preliminary hypothesis is that the LMs construct a "pivot" similar to the label tokens in classification tasks, which functions as a summary representation for induction-like processing and subsequent outputs. Confirming this hypothesis would further complete and generalize our theoretical framework.

Reverse-order In-context Learning. As described in §2.2.1.2, Min et al. (2022a) adopted the noise channel method with a reversed ICL prompt, i.e., placing the label before the sentence in the demonstration. And for a given query, the noise channel method iterates over all possible labels and uses the generation probability of the query sentence as the logits for each label. Although this method can achieve high-performance ICL inference, its internal mechanism remains unclear. Moreover, it represents a simplified case of generative tasks, so that investigating the mechanism behind the noise channel method may produce insight into the mechanism of the generative scenario.

Towards Mechanistic Controllability. In the §6, we propose a method to optimize the correctness of induction circuits based on the mechanism understanding, thereby improving ICL performance. This method

serves as a prototype that establishes a new area: localizing submodules that exhibit strong effectiveness for a certain inference behavior through mechanistic interpretability methods, and refining these localized submodules to replicate the full-model fine-tuning effects. Such methods benefit from improved efficiency while preserving out-of-domain generalization ability since only the minimal modules are updated. We refer to this approach as Mechanistic Controllability. Future research can extend it into a generalizable methodological framework.

Feature Dynamics. In current interpretation works based on attention heads, most methodologies focus on observing the attention scores and their variants (Kobayashi et al., 2020), identifying the dominant attention links, and thereby concluding that "this attention head *mixes* information from certain hidden states". However, the term *mixing* is ambiguous: depending on the nature of features in the hidden states, this mixing process may lead to semantic enhancement or cancellation, as shown in Fig. 3.5 (Right), or, when the feature directions are near-orthogonal, it may simply result in a straightforward appending. And these different behaviors lead to entirely different interpretations of the attention operations. Therefore, designing a method framework to examine the concrete dynamics of features conducted by the addition calculation in attention heads is crucial. Although this dissertation avoids the issue by using specialized techniques such as feature alignment computation or semantic probes, resolving it would still provide a powerful toolkit for the mechanistic interpretability area.

The Function of Instruction. In this dissertation, we do not discuss the instruction much, which is the task description located before the demonstrations, and also the instruction does not seem to participate in the mechanism before (neither the induction mechanism, nor the information removal mechanism), yet it indeed improves accuracy, especially in fact recall tasks. This phenomenon remains unexplained. A plausible hypothesis is that instructions actually induce a translational task vector, shifting the output toward the desired output space through a consistent bias. This hypothesis requires further validation.

Module Connectivity in Circuits. In this dissertation, particularly in §3, we proposed a circuit composed of several attention heads. However, the degree of connectivity within this circuit, i.e., how the output of one step is utilized by the next, has not been evaluated, which limits the reliability of our conclusions. At present, we lack a reliable framework to evaluate such connectivity. Therefore, future work could focus on developing a method to verify the tightness of circuit connections, which would serve as an essential toolkit for mechanistic interpretability.

References

Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen. Enhancing in-context learning via linear probe calibration. *arXiv preprint arXiv:2401.12406*, 2024. URL http://arxiv.org/abs/2401.12406v1.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie CY Chan, Biao Zhang, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://arxiv.org/abs/2404.11018.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023. URL https://arxiv.org/abs/2311.16867.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL http://arxiv.org/abs/1607.06450v1.

Aleksandra Bakalova, Yana Veitsman, Xinting Huang, and Michael Hahn. Contextualize-then-aggregate: Circuits for in-context learning in gemma-2 2b. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=rGNAyHReSg.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki, and Saif M. Mohammad (eds.), *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2007. URL https://aclanthology.org/S19-2007/.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. URL http://arxiv.org/abs/2303.08112v5.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024. URL https://arxiv.org/abs/2405.00200.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023. URL https://arxiv.org/abs/2304.01373.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Marc-Etienne Brunet, Ashton Anderson, and Richard Zemel. Icl-markup: Structuring in-context learning using soft-token tags. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. URL https://arxiv.org/pdf/2312.07405.

Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Qingfu Zhang, Hau-San Wong, and Taiji Suzuki. Provable in-context vector arithmetic via retrieving task concepts. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DbUmeNnNpt.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=d4dpOCqybL.

Wang Cai, Hsiu-Yuan Huang, Zhixiang Wang, and Yunfang Wu. Beyond demonstrations: Dynamic vector construction from latent representations. *arXiv preprint arXiv:2505.20318*, 2025. URL http://arxiv.org/abs/2505.20318v1.

Bryan Chan, Xinyi Chen, András György, and Dale Schuurmans. Toward understanding in-context vs. in-weight learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aKJr5NnN8U.

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*, 35:18878–18891, 2022. URL https://arxiv.org/abs/2205.05055.

Ting-Yun Chang and Robin Jia. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8123–8144, 2023. URL https://arxiv.org/abs/2212.10378.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. When parts are greater than sums: Individual LLM components can outperform full models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10280–10299, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.574. URL https://aclanthology.org/2024.emnlp-main.574/.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024. URL https://arxiv.org/abs/2402.03216.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3558–3573, 2022. URL https://arxiv.org/pdf/2205.01703.

Yixiong Chen, Alan Yuille, and Zongwei Zhou. Which layer is learning faster? a systematic exploration of layer-wise convergence rate for deep neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=wlMDF1jQF86.

Hakaze Cho and Naoya Inoue. Staicc: Standardized evaluation for classification task in in-context learning. *arXiv preprint arXiv:2501.15708*, 2025. URL https://arxiv.org/abs/2501.15708.

Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=xizpnYNvQq.

Hakaze Cho, Peng Luo, Mariko Kato, Rin Kaenbyou, and Naoya Inoue. Mechanistic fine-tuning for in-context learning. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 2025b. URL https://arxiv.org/abs/2505.14233.

Hakaze Cho, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, and Naoya Inoue. Token-based decision criteria are suboptimal in in-context learning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5378–5401, Albuquerque, New Mexico, April 2025c. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.278. URL https://aclanthology.org/2025.naacl-long.278/.

Hakaze Cho, Yoshihiro Sakai, Kenshiro Tanaka, Mariko Kato, and Naoya Inoue. Understanding token probability encoding in output embeddings. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10618–10633, Abu Dhabi, UAE, January 2025d. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.708/.

Hakaze Cho, Haolin Yang, Brian M. Kurkoski, and Naoya Inoue. Binary autoencoder for mechanistic interpretability of large language models. *arXiv preprint arXiv:2509.20997*, 2025e. URL http://arxiv.org/abs/2509.20997v1.

Hakaze Cho, Haolin Yang, Gouki Minegishi, and Naoya Inoue. Mechanism of task-oriented information removal in in-context learning. *arXiv preprint arXiv:2509.21012*, 2025f. URL http://arxiv.org/abs/2509.21012v1.

Liam Collins, Advait Parulekar, Aryan Mokhtari, Sujay Sanghavi, and Sanjay Shakkottai. In-context learning with transformers: Softmax attention adapts to function lipschitzness. *arXiv preprint arXiv:2402.11639*, 2024. URL https://arxiv.org/abs/2402.11639.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=89ia77nZ8u.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://arxiv.org/abs/2212.10559.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pp. 11–20, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5102. URL https://www.aclweb.org/anthology/W18-5102.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

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. URL https://arxiv.org/abs/2301.00234.

Yuxin Dong, Jiachen Jiang, Zhihui Zhu, and Xia Ning. Understanding task vectors in in-context learning: Emergence, functionality, and limitations. *arXiv preprint arXiv:2506.09048*, 2025. URL http://arxiv.org/abs/2506.09048v1.

Carsten Eickhoff. Impact factors for computer science conferences. *arXiv preprint arXiv:2310.08037*, 2023. URL http://arxiv.org/abs/2310.08037v1.

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*, 1(1):12, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022. URL http://arxiv.org/abs/2209.10652v1.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, pp. 14014–14031, 2023. URL http://arxiv.org/abs/2305.19148v3.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=5uwBzcn885.

Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in neural information processing systems*, 33:5922–5932, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/41c542dfe6e4fc3deb251d64cf6ed2e4-Paper.pdf.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10136–10148, 2023. URL http://arxiv.org/abs/2212.04037v2.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024. URL http://arxiv.org/abs/2407.21783v3.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=78Nn4QJTEN.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-training to learn in context. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4849–4870, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.267. URL https://aclanthology.org/2023.acl-long.267/.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-training to learn in context. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, pp. 4849–4870, 2023b. URL http://arxiv.org/abs/2305.09137v1.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017. URL http://arxiv.org/abs/1706.04599v2.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 14892–14904, 2024. URL https://aclanthology.org/2024.findings-acl.884/.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023a. URL https://arxiv.org/abs/2305.12766.

Seungwook Han, Jinyeop Song, Jeff Gore, and Pulkit Agrawal. Emergence and effectiveness of task vectors in in-context learning: An encoder decoder perspective. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0ysC6VS0y3.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12660–12673, 2023b. URL http://arxiv.org/abs/2306.15091v1.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*, 2023c. URL https://arxiv.org/abs/2205.10183.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2022. URL http://arxiv.org/abs/2212.06713v1.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, 2021. URL https://aclanthology.org/2021.emnlp-main.564.pdf.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. URL https://www.semanticscholar.org/paper/f22f6972e66bdd2e769fa64b0df0a13063c0c101.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001. URL https://www.aclweb.org/anthology/H01-1069.

Aliyah R. Hsu, Georgia Zhou, Yeshwanth Cherapanamjeri, Yaxuan Huang, Anobel Odisho, Peter R. Carroll, and Bin Yu. Efficient automated circuit discovery in transformers using contextual decomposition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=41HlN8XYM5.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023. URL https://arxiv.org/abs/2310.05249.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024. URL https://arxiv.org/abs/2405.07987.

icyrockcom. country-capitals, 2011. URL https://github.com/icyrockcom/country-capitals.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL http://arxiv.org/pdf/2212.04089.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022. URL https://arxiv.org/abs/2212.12017.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356/.

Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis of in-context learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2401.15530.

Gangwei Jiang, Caigao JIANG, Zhaoyi Li, Siqiao Xue, JUN ZHOU, Linqi Song, Defu Lian, and Ying Wei. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gc8QAQfXv6.

Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=otuTw4Mghk.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2312–2333, 2023. URL http://arxiv.org/abs/2310.10266v1.

Mariko Kato, Hakaze Cho, Yoshihiro Sakai, and Naoya Inoue. Affinity and diversity: A unified metric for demonstration selection via internal representations. *arXiv preprint arXiv:2502.14380*, 2025. URL http://arxiv.org/abs/2502.14380v1.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*, 2022. URL https://arxiv.org/abs/2206.08082.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

Artem Kirsanov, Chi-Ning Chou, Kyunghyun Cho, and SueYeon Chung. The geometry of prompting: Unveiling distinct mechanisms of task adaptation in language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1855–1888, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.100. URL https://aclanthology.org/2025.findings-naacl.100/.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *arXiv preprint arXiv:2305.06329*, 2023. URL https://arxiv.org/abs/2305.06329.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574/.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017. URL http://arxiv.org/abs/1703.04730v3.

Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.12375.

Morgane Laouenan, Palaash Bhargava, Jean-Benoît Eyméoud, Olivier Gergaud, Guillaume Plique, and Etienne Wasmer. A cross-verified database of notable people, 3500bc-2018ad. *Scientific Data*, 9(1):290, Jun 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01369-4. URL https://doi.org/10.1038/s41597-022-01369-4.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

Jiaoda Li, Yifan Hou, Mrinmaya Sachan, and Ryan Cotterell. What do language models learn in context? the structured task hypothesis. *arXiv preprint arXiv:2406.04216*, pp. 12365–12379, 2024. URL http://arxiv.org/abs/2406.04216v3.

Jiaqian Li, Yanshu Li, Ligong Han, Ruixiang Tang, and Wenya Wang. Towards generalizable implicit in-context learning with attention routing. *arXiv preprint arXiv:2509.22854*, 2025a. URL https://arxiv.org/abs/2509.22854.

Maximilian Li and Lucas Janson. Optimal ablation for interpretability. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=opt72TYzwZ.

Xiaonan Li and Xipeng Qiu. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6219–6235, 2023. URL https://arxiv.org/abs/2302.13539.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, GUOTONG XIE, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023. URL http://arxiv.org/abs/2305.04320v2.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://www.aclweb.org/anthology/C02-1150.

Zhuowei Li, Zihao Xu, Ligong Han, Yunhe Gao, Song Wen, Di Liu, Hao Wang, and Dimitris N. Metaxas. Implicit in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=G7u4ue6ncT.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, 2022. URL https://aclanthology.org/2022.deelio-1.10.pdf.

Jinlong Liu, Guoqing Jiang, Yunzhi Bai, Ting Chen, and Huayan Wang. Understanding why neural networks generalize well through gsnr of parameters. *arXiv preprint arXiv:2001.07384*, 2020. URL https://arxiv.org/abs/2001.07384.

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024. URL http://arxiv.org/abs/2402.10738v2.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022. URL http://arxiv.org/abs/2104.08786v2.

Gaoxiang Luo and Aryan Deshwal. Com-bom: Bayesian exemplar search for efficiently exploring the accuracy-calibration pareto frontier. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://arxiv.org/abs/2510.01178.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014. URL https://arxiv.org/abs/1307.5336.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*, 2023. URL http://arxiv.org/abs/2310.20046v1.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-h6WAS6eE4.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5030–5047, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.281. URL https://aclanthology.org/2024.naacl-long.281/.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5316–5330, 2022a. URL http://arxiv.org/abs/2108.04106v3.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, 2022b. URL https://aclanthology.org/2022.naacl-main.201.pdf.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022c. URL https://aclanthology.org/2022.emnlp-main.759.pdf.

Gouki Minegishi, Hiroki Furuta, Shohei Taniguchi, Yusuke Iwasawa, and Yutaka Matsuo. Beyond induction heads: In-context meta learning induces multi-phase circuit emergence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Xw01vF13aV.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pp. 3470–3487. Association for Computational Linguistics (ACL), 2022. URL https://aclanthology.org/2022.acl-long.244.pdf.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025. URL https://arxiv.org/abs/2505.18235.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 task 1: Affect in tweets. In Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat (eds.), *Proceedings of the 12th International Workshop on Semantic Evaluation*, pp. 1–17, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1001. URL https://aclanthology.org/S18-1001/.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. URL https://proceedings.neurips.cc/paper/2020/file/0607f4c705595b911a4f3e7a127b44e0-Paper.pdf.

nostalgebraist. interpreting GPT: the logit lens — AI Alignment Forum — alignmentforum.org, 2020. URL https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. URL https://arxiv.org/abs/2209.11895.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.527. URL https://aclanthology.org/2023.findings-acl.527/.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 115–124, 2005. URL https://aclanthology.org/P05-1015/.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In Preslav Nakov and Torsten Zesch (eds.), *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 27–35, Dublin, Ireland, 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2004. URL https://aclanthology.org/S14-2004.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*, 2023. URL http://arxiv.org/abs/2310.09881v4.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://www.semanticscholar.org/paper/9405cc0d6169988371b2755e573cc28650d14dfe.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024. URL https://arxiv.org/abs/2407.02646.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024a. URL http://arxiv.org/abs/2404.16014v2.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024b. URL http://arxiv.org/abs/2407.14435v3.

Allan Raventos, Mansheej Paul, Feng Chen, and Surya Ganguli. 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.

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. URL https://arxiv.org/abs/2312.03002.

Sara Rosenthal, Noura Farra, and Preslav Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 502–518, 2017. URL http://arxiv.org/abs/1912.00741v1.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191/.

Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. Large language models encode semantics in low-dimensional linear subspaces. *arXiv preprint arXiv:2507.09709*, 2025. URL http://arxiv.org/abs/2507.09709v2.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ITw9edRDlD.

Emily Sheng and David Uthus. Investigating societal biases in a poetry composition system, 2020. URL http://arxiv.org/abs/2011.02686v1.

Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models. *arXiv preprint arXiv:2503.08200*, 2025. URL http://arxiv.org/abs/2503.08200v3.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. Nearest neighbor zero-shot inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3254–3265, 2022. URL https://arxiv.org/pdf/2205.13792.

Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? *arXiv preprint arXiv:2405.19592*, 2024. URL http://arxiv.org/abs/2405.19592v1.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. On the effect of pretraining corpora on in-context learning by a large-scale language model. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5168–5186, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.380. URL https://aclanthology.org/2022.naacl-main.380/.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *arXiv preprint arXiv:2503.05613*, 2025. URL https://arxiv.org/abs/2503.05613.

Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 36, 2024a. URL https://arxiv.org/abs/2311.08360.

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. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://arxiv.org/abs/2404.07129.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=qY1hlv7gwg.

Xiaofei Sun, Diyi Yang, Xiaoya Li, Tianwei Zhang, Yuxian Meng, Han Qiu, Guoyin Wang, Eduard Hovy, and Jiwei Li. Interpreting deep learning models in natural language processing: A review. *arXiv preprint arXiv:2110.10470*, 2021. URL http://arxiv.org/abs/2110.10470v2.

Junyi Tao, Xiaoyin Chen, and Nelson F. Liu. Inference and verbalization functions during in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16394–16421, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.957. URL https://aclanthology.org/2024.findings-emnlp.957/.

Qwen Team. Qwen2.5: A party of foundation models, September 2024a. URL https://qwenlm.github.io/blog/qwen2.5/.

TII Team. Falcon 3 family of open foundation models, December 2024b. URL https://huggingface.co/tiiuae/Falcon3-7B-Base.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2012. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL http://arxiv.org/abs/2307.09288v2.

Minh-Hao Van, Xintao Wu, et al. In-context learning demonstration selection via influence analysis. *arXiv preprint arXiv:2402.11750*, 2024. URL http://arxiv.org/abs/2402.11750v2.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. URL http://arxiv.org/abs/1706.03762v7.

Anton Voronov, Lena Wolf, and Max Ryabinin. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*, 2024. URL http://arxiv.org/abs/2401.06766v3.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. URL https://www.aclweb.org/anthology/W18-5446.pdf. In the Proceedings of ICLR.

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. URL https://arxiv.org/abs/2305.14160.

Sida Wang and Christopher D. Manning. Baselines and bigrams: simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pp. 90–94, USA, 2012. Association for Computational Linguistics. URL https://www.semanticscholar.org/paper/5e9fa46f231c59e6573f9a116f77f53703347659.

Xiaolei Wang, Xinyu Tang, Junyi Li, Xin Zhao, and Ji-Rong Wen. Investigating the pre-training dynamics of in-context learning: Task recognition vs. task learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=htDczodFN5.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021. URL http://arxiv.org/abs/2010.13166v2.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, 2022. URL https://aclanthology.org/2022.emnlp-main.340.pdf.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021. URL http://arxiv.org/abs/2109.01652v5.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 968–979, 2023a. URL http://arxiv.org/abs/2305.08298v2.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023b. URL https://arxiv.org/abs/2303.03846.

Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024. URL http://arxiv.org/abs/2303.07895v1.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, 2023. URL http://arxiv.org/abs/2212.10375v2.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. k nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. In *The Eleventh International Conference on Learning Representations*, 2022. URL http://arxiv.org/pdf/2303.13824.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-Guang Lou, and Shuai Ma. Re-reading improves reasoning in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15549–15575, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.871. URL https://aclanthology.org/2024.emnlp-main.871/.

Zhichao Xu, Daniel Cohen, Bei Wang, and Vivek Srikumar. In-context example ordering guided by label distributions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2623–2640, 2024b. URL http://arxiv.org/abs/2402.11447v1.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL http://arxiv.org/abs/2407.10671v4.

Haolin Yang, Hakaze Cho, Kaize Ding, and Naoya Inoue. Task vectors, learned not extracted: Performance gains and mechanistic insight. *arXiv preprint arXiv:2509.24169*, 2025a. URL http://arxiv.org/abs/2509.24169v1.

Haolin Yang, Hakaze Cho, and Naoya Inoue. Localizing task recognition and task learning in in-context learning via attention head analysis. *arXiv preprint arXiv:2509.24164*, 2025b. URL http://arxiv.org/abs/2509.24164v1.

Haolin Yang, Hakaze Cho, Yiqiao Zhong, and Naoya Inoue. Unifying attention heads and task vectors via hidden state geometry in in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c. URL https://openreview.net/forum?id=FIfjDqjV0B.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=C7XmEByCFv.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. Ground-truth labels matter: A deeper look into input-label demonstrations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2422–2437, 2022. URL https://arxiv.org/abs/2205.12685.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.acl-main.148.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. URL https://arxiv.org/abs/1611.03530.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023. URL https://arxiv.org/abs/2306.09927.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://www.semanticscholar.org/paper/13a0d8bb38f739990c8cd65a44061c6534f17221.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015. URL https://arxiv.org/abs/1509.01626.

Ying Zhang, Benjamin Heinzerling, Dongyuan Li, Ryoma Ishigaki, Yuta Hitomi, and Kentaro Inui. Understanding fact recall in language models: Why two-stage training encourages memorization but mixed training teaches knowledge. *CoRR*, abs/2505.16178, May 2025. URL https://doi.org/10.48550/arXiv.2505.16178.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021. URL https://arxiv.org/abs/1707.08114.

Haiyan Zhao, Heng Zhao, Bo Shen, Ali Payani, Fan Yang, and Mengnan Du. Beyond single concept vector: Modeling concept subspace in LLMs with gaussian distribution. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CvttyK4XzV.

Yufeng Zhao, Yoshihiro Sakai, and Naoya Inoue. Noisyicl: A little noise in model parameters calibrates in-context learning. *arXiv preprint arXiv:2402.05515*, 2024. URL https://arxiv.org/abs/2402.05515.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021. URL http://arxiv.org/abs/2102.09690v2.

Bowen Zheng, Ming Ma, Zhongqiao Lin, and Tianming Yang. Distributed rule vectors is a key mechanism in large language models' in-context learning. *arXiv preprint arXiv:2406.16007*, 2024. URL https://arxiv.org/abs/2406.16007.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=L3FHMoKZcS.

Yuxiang Zhou, Jiazheng Li, Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14365–14378, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.795. URL https://aclanthology.org/2024.emnlp-main.795/.

Appendix A

Supplementary Materials for Chapter 3

A.1 Experiment Details and Settings

A.1.1 Detailed Overall Experimental Settings

Table A.1: Prompt templates used in this work.

Dataset	Prompt Template (Unit)	Label Tokens
SST-2	`sentence: [input sentence] sentiment: [label token] \n`	negative, positive
MR	`review: [input sentence] sentiment: [label token] \n`	negative, positive
FP	`sentence: [input sentence] sentiment: [label token] \n`	negative, neutral, positive
SST-5	`sentence: [input sentence] sentiment: [label token] \n`	poor, bad, neutral, good, great
TREC	`question: [input sentence] target: [label token] \n`	short, entity, description, person, location, number
AGNews	`news: [input sentence] topic: [label token] \n`	world, sports, business, science

Prompt Template. We conduct experiments on a specific prompt template for each dataset as shown in Table A.1. Moreover, similar to typical ICL practices, we reduce the label into one token to simplify the prediction decoding. The reduced label tokens are also shown in Table A.1.

Quantization. In our experiments, we use `BitsAndBytes`[22] to quantize Llama 3-70B and Falcon-40B to `INT4`. For the other models, full-precision inference is conducted.

Other. All the experiment materials (models and datasets) are downloaded from `huggingface`. For the BGE M3, we use its `pooler_output` as the output feature.

A.1.2 Calculation of Mutual Nearest-neighbor Kernel Alignment

In this chapter, we need to measure the similarity between features from two different models or model layers. There are many approaches (Klabunde et al., 2023), and we use mutual nearest-neighbor kernel alignment (Huh et al., 2024), which is relatively concise and accurate, calculated as follows to measure the similarity of representation from the same object set $\mathcal{X} = \{x_i\}_{i=1}^n$ in different feature spaces.

Calculation of Mutual Nearest-neighbor Kernel Alignment. Given a representation mapping $\delta : \mathcal{X} \to \mathbb{H}^d$ from the objects to a space where similarity measurement $\langle \cdot, \cdot \rangle : \mathbb{H}^d \times \mathbb{H}^d \to \mathbb{R}$ is defined, we can calculate the similarity map from dataset $\mathcal{X}$ as $\mathcal{S}_\delta \in \mathbb{R}^{n \times n}$, where the elements are $\mathcal{S}_{\delta|i,j} = \langle \delta(x_i), \delta(x_j) \rangle$, especially, we axiomatic define $\langle x, x \rangle = 1$, so we set the diagonal element $\mathcal{S}_{\delta|i,i} \coloneqq 0$ since they are trivial values that disturb the following calculation with values 1.

Given two representation functions δ_1 and δ_2, two similarity map can be calculated as $\mathcal{S}_{\delta_1}$ and $\mathcal{S}_{\delta_2}$ on the same object set $\mathcal{X}$. For each line vector index $i = 1, 2, \ldots, n$ in $\mathcal{S}_{\delta_1}$, we select **the index of** top-K elements from greater to lower as $\text{top}_k\left(\mathcal{S}_{\delta_1|i}\right)$. Similarly, we get $\text{top}_K\left(\mathcal{S}_{\delta_2|i}\right)$ from $\mathcal{S}_{\delta_2}$.

Then, we calculate the kernel alignment for sample i as:

$$\text{KA}_{\mathcal{X}}(\delta_1, \delta_2)_i = \frac{\left|\text{top}_K\left(\mathcal{S}_{\delta_1|i}\right) \cap \text{top}_K\left(\mathcal{S}_{\delta_2|i}\right)\right|}{k}. \tag{A.1}$$

The kernel alignment for dataset $\mathcal{X}$ is the average on each $\text{KA}_{\mathcal{X}}(\delta_1, \delta_2)_i$.

Implementation. In our experiments, we choose cosine similarity as the $\langle \cdot, \cdot \rangle$, and $K \coloneqq 64$. According to experiment settings in §3.2.2, $n \coloneqq 512$ is defined, and a randomlized matrix $\mathcal{S}$ have $\text{KA} = 64/512 = 0.125$ as the random baseline.

[22] https://huggingface.co/docs/bitsandbytes/main/en/index

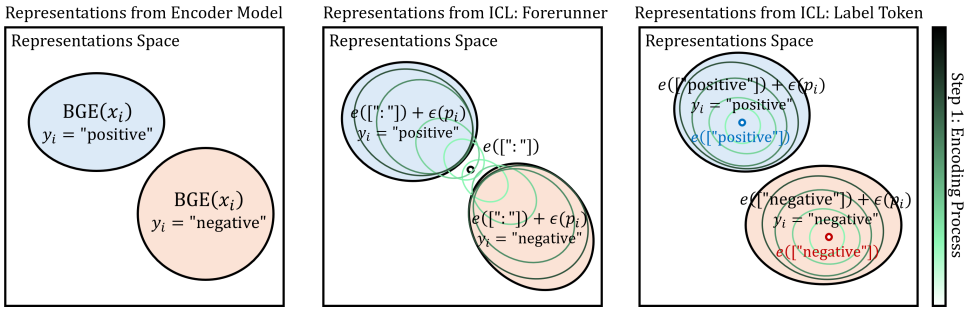


Figure A.1: Distributions (*clusters*) of representations generated by: **Left**: encoder model (BGE), clustering w.r.t. the label. **Middle**: ICL on the forerunner token, where representations gather into one point ($e([": "])$) when no Transformer operation is conducted. **Right**: ICL on the label token, where representations gather into points w.r.t. label (using a 2-way example) when no Transformer operation is conducted, causing a high background value.

A.1.2.1 Background Values of Kernel Alignment: Label Token - Text Encoding.

In a specific layer of a decoder Transformer, the representations of a token x can be written as $\delta(x) = e(x) + \epsilon(p)$, where $e(x)$ is the embedding vector of the token x, and $\epsilon(p)$ are the residual side-flow w.r.t. the context p. Given two specific tokens x_i and x_j where kernel alignment is calculated from different ICL-styled input sequences p_i and p_j, we discuss the background value of kernel alignment as below.

Intuition. As shown in Fig. A.1, the hidden states on the label token have a prior clustering, making them naturally similar to the representation generated by the encoder model, even if the ICL process does not encode it sufficiently. So, at layer 0, since the model is not able to perform any contextual encoding in this layer, the kernel alignment on the forerunner is on a random baseline value of 0.125, but the value on the label token will be greater (estimated below). In the case where (1) BGE generates fully linearly separable clusters against x in each class, with sufficient inter-cluster distance, and (2) the number of samples for each label is not less than 64, such upper bias can be expected to be $0.125(|\mathbb{Y}| - 1)$ (proof omitted). Intuitively, such bias can propagate along the residual network from the embedding of $[y]$ to every layer, making the results of every layer upper-biased.

Similarity on Forerunner Tokens. The cosine similarity between $\delta(x_i)$ and $\delta(x_j)$, where the x_i and x_j are forerunner tokens of the input sequence can be written as:

$$\langle \delta(x_i), \delta(x_j) \rangle = \langle e(x_i) + \epsilon(p_i), e(x_j) + \epsilon(p_j) \rangle \tag{A.2}$$

$$= \frac{\langle e(x_i), e(x_j) \rangle + \langle e(x_i), \epsilon(p_j) \rangle + \langle e(x_j), \epsilon(p_i) \rangle + \langle \epsilon(p_i), \epsilon(p_j) \rangle}{\|e(x_i) + \epsilon(p_i)\|_2 \, \|e(x_j) + \epsilon(p_j)\|_2}. \tag{A.3}$$

Denote $B_{i,j} = \langle e(x_i), \epsilon(p_j) \rangle + \langle e(x_j), \epsilon(p_i) \rangle$, $C_{i,j} = \|e(x_i) + \epsilon(p_i)\|_2 \, \|e(x_j) + \epsilon(p_j)\|_2$, and notice that $x_i = x_j$ since they are forerunner tokens which is kept consistent in experiments, we have:

$$\langle \delta(x_i), \delta(x_j) \rangle = \frac{1 + B_{i,j} + \langle \epsilon(p_i), \epsilon(p_j) \rangle}{C_{i,j}}. \tag{A.4}$$

Similarity on Label Tokens. Similarly, the cosine similarity between $\delta(y_i)$ and $\delta(y_j)$ on label tokens can be written as:

$$\langle \delta(y_i), \delta(y_j) \rangle = \frac{\langle e(y_i), e(y_j) \rangle + B_{i,j} + \langle \epsilon(p_i), \epsilon(p_j) \rangle}{C_{i,j}}. \tag{A.5}$$

Encoding on Label Tokens Enhances the Similarity with Same Labels. Given $K = 1$ for simplicity, the probability $\mathrm{top}_1\left(\mathcal{S}_{\delta|i}\right)$ selects a sample with the similar label with i-th sample **on the forerunner token** can be written as:

$$\mathrm{P}_F\left[y_i = y_{\mathrm{top}_1(\mathcal{S}_{\delta|i})}\right] = r_{y_i} \frac{\mathbb{E}_{j|y_i=y_j}\left[\langle \delta(x_i), \delta(x_j) \rangle\right]}{\mathbb{E}_j\left[\langle \delta(x_i), \delta(x_j) \rangle\right]} \tag{A.6}$$

$$\approx r_{y_i} \frac{1 + \mathbb{E}_{j|y_i=y_j}\left[B_{i,j}\right] + \mathbb{E}_{j|y_i=y_j}\left[\langle \epsilon(p_i), \epsilon(p_j) \rangle\right]}{1 + \mathbb{E}_j\left[B_{i,j}\right] + \mathbb{E}_j\left[\langle \epsilon(p_i), \epsilon(p_j) \rangle\right]}, \tag{A.7}$$

where the r_{y_i} is the label ratio of y_i. Similarly, the probability on the label token can be written as:

$$\begin{aligned}
&\mathrm{P}_L\left[y_i = y_{\mathrm{top}_1(\mathcal{S}_{\delta|i})}\right] \\
&\approx \frac{r_{y_i}(1 + \mathbb{E}_{j|y_i=y_j}\left[B_{i,j}\right] + \mathbb{E}_{j|y_i=y_j}\left[\langle\epsilon(p_i), \epsilon(p_j)\rangle\right])}{\begin{array}{l} r_{y_i}\left(1 + \mathbb{E}_{j|y_i=y_j}\left[B_{i,j}\right] + \mathbb{E}_{j|y_i=y_j}\left[\langle\epsilon(p_i), \epsilon(p_j)\rangle\right]\right) \\ \quad + (1 - r_{y_i})\left(\langle e(y_i), e(y_j)\rangle + \mathbb{E}_j\left[B_{i,j}\right] + \mathbb{E}_j\left[\langle\epsilon(p_i), \epsilon(p_j)\rangle\right]\right) \end{array}} \\
&\geqslant \mathrm{P}_F\left[y_i = y_{\mathrm{top}_1(\mathcal{S}_{\delta|i})}\right]. \qquad \text{(A.8)}
\end{aligned}$$

That is, the inputs with the same labels with x_i are easier to be selected into the $\mathrm{top}_1\left(\mathcal{S}_{\delta|i}\right)$. Notice that we make approximations here: (1) we consider the $\mathbb{E}_{j|y_i=y_j}\left[C_{i,j}\right] \approx \mathbb{E}_j\left[C_{i,j}\right]$, i.e., the 2-norm of two encoding vectors are considered equal granted by normalization used in Transformer. (2) We consider the context term c in the label token scenario the same as the forerunner scenario since the difference is only a label token, which usually occupies quite a small part of the input sequence.

Background Values of Kernel Alignment. According to the explanation above, as shown in Fig. A.1, it is intuitive to conclude that $\mathrm{top}_K\left(\mathcal{S}_{\delta|i}\right)$ from label tokens is easier to cluster samples with the same label as y_i. Moreover, a well-pre-trained encoder can catch the prior distribution of the input texts determined by their labels, and also cluster samples with the same label, causing a high similarity of similarity map, so that a high but unfaithful kernel alignment as the background value from the similarity on $e(y)$ but not $\epsilon(p)$. An intuitive verification of such background value is shown in the Fig. 3.3 (Left), where the "Label Token" curve has a high value in layer 0 with $\epsilon(p) = 0$. However, the background value also indicates that the representation generated by BGE correctly clusters the samples, confirming its reliability.

A.1.3 Training and Inference of Centroid Classifier

In this chapter, we follow Cho et al. (2025c) to train centroid classifiers as a probe toward hidden states of LMs. In detail, given the LM's hidden states set $\left\{h_i^l\right\}_{i=1}^m$ of the selected tokens (according to the experimental setting, the last label token or forerunner token) in layer l from an [ICL input]-[query label] set $\mathcal{Z} = \{(p_i, y_i)\}_{i=1}^m$, where the labels are limited in label space $\mathbb{Y}$, in the **training phase**, we calculate the centroid of the hidden state $\bar{h}_y^l$ for each label respectively:

$$\bar{h}_y^l = \mathbb{E}_{i|y_i=y}\left[h_i^l\right]. \qquad \text{(A.9)}$$

In the **inference phase**, we extract the equitant[23] hidden state h_t^l as the training phase from the test input, and calculate the similarity between h_t^l and the centroids calculated above. Then, we choose the label of the most similar centroids as the prediction:

$$\mathcal{C}(h_t^l) = \underset{y}{\mathrm{argmax}}\left\langle h_t^l, \bar{h}_y^l\right\rangle. \qquad \text{(A.10)}$$

Implementation. In our experiments, we set training sample number $m \coloneqq 256$, similarity function $\langle a, b\rangle \coloneqq -\|a - b\|_2$.

A.1.4 Cartography Details of Fig. 3.6 (Middle)

In Fig. 3.6 (Middle), we define a Correct Label Assignment, here we introduce how this measurement is calculated. Suppose we have an attention score $\mathcal{A}_{W,f}(K, Q)$ calculated as (also written as $\alpha_{K\to Q}$ in brief):

$$\mathcal{A}_{W,f}\left(K, Q\right) = f\left(Q^\top W K\right), \qquad \text{(A.11)}$$

with hidden dimensionality of d, give a certain layer, $K \in \mathbb{R}^{d\times n_t}$ is the hidden state matrix of full context, $Q \in \mathbb{R}^{d\times 1}$ is the hidden state of query's forerunner token ($Q^\top = K_{n_t}^\top$), $f : \mathbb{R}^{n_t} \to \Omega^{n_t}$ is a normalization mapping from n_t-dimensional real vector to n_t-dimensional probability vector (usually $\mathrm{softmax}$ function), the W is a linear kernel, usually $W_Q^{h\top} W_K^h$ for multi-head attention or $\mathrm{I} = \mathrm{diag}\left(\mathbb{1}^{\mathrm{n_t}}\right)$ for vanilla attention.

For one input sample, given the token-index set of label tokens as $\mathcal{L}$, the token-index set of label tokens which is the same as the query's ground truth label as $\mathcal{L}^+$, we define the **C**orrect **L**abel **A**ssignment (CLA) of one

[23] Equitant refers to hidden states from the same layer and token type. While, in experiments shown in Fig. 3.5 (Right), we don't keep the token type consistent.

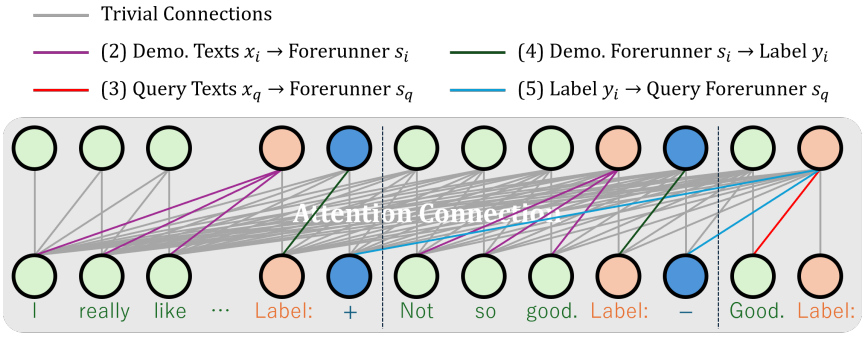


Figure A.2: Visualization of non-trivial attention connection defined and disconnected in ablation analysis in §3.5.1 and Table 3.1. Notations and line numbers are same as Table 3.1.

sample as:

$$\mathrm{CLA}_{W,f}(K,Q,\mathcal{L},\mathcal{L}^+) = \frac{\sum_{i\in\mathcal{L}^+}\mathcal{A}_{W,f}\left(K,Q\right)_i}{\sum_{i\in\mathcal{L}}\mathcal{A}_{W,f}\left(K,Q\right)_i}. \tag{A.12}$$

Intuitively, CLA reflects the accuracy of attention computation $\mathcal{A}_{W,f}$ towards label tokens on one input. For an input set built from a dataset, we calculate the averaged CLA on these inputs, and repeat in every layer to plot a curve of Averaged CLA against layer numbers. Specifically:

(1) Vanilla attention. We assign $W \coloneqq \mathrm{I}$, f to linear normlization.

(2) Best Induction Head. For each attention head h, we assign $W^h \coloneqq W_Q^{h\top}W_K^h$, $f \coloneqq \mathrm{softmax}$. For each input, we calculate $\max_h \mathrm{CLA}_{W^h,f}\left(K,Q,\mathcal{L},\mathcal{L}^+\right)$ as the result for single input.

(3) Head Average. For each attention head h, we assign $W^h \coloneqq W_Q^{h\top}W_K^h$, $f \coloneqq \mathrm{softmax}$. For each input, we calculate $\sum_h \mathrm{CLA}_{W^h,f}\left(K,Q,\mathcal{L},\mathcal{L}^+\right)/|\mathcal{H}|$, where the $|\mathcal{H}|$ is the amount of heads in current layer, as the result for single input.

Note that we do not consider the absolute value of attention assignment on label tokens in this experiment, and most of the heads have little scores assigned to the label (Fig. 3.6 (Left)), therefore, although the average assignments tend to be average, this result shown in Fig. 3.6 (Middle) does not contradict the phenomenon that ICL can achieve high accuracy.

A.1.5 Cartography Details of Fig. 3.7

For Fig. 3.7, we input one sample from SST-2 into Llama 3-70B, take the output of layer 30 on the label tokens to span a matrix $K_{\mathcal{L}}$, and map them by $W_Q^{h\top}W_K^h$ of head 32 (the best induction heads in this layer) and 9 (the worst induction heads) of layer 31 (the layer with the most correct induction heads), respectively. We visualize the distribution of these mapped $W_Q^{h\top}W_K^hK_{\mathcal{L}}$, we conduct principal component analysis on them, and plot them on the plane of the first two components.

For each point $q \in \mathbb{R}^2$ on the principal component plane, we calculate the attention assignment as follows. Give the index set of "positive" label token in $K_{\mathcal{L}}$ as $\mathcal{L}^+$, the index set of "negative" label token as $\mathcal{L}^-$, we calculate the attention assignment, which can be an estimate of ICL prediction (Wang et al., 2023), as:

$$\mathrm{AttAssign}(q) = \sum_{i\in\mathcal{L}^+} q^\top W_Q^{h\top}W_K^hK_{\mathcal{L}|i} - \sum_{i\in\mathcal{L}^-} q^\top W_Q^{h\top}W_K^hK_{\mathcal{L}|i}. \tag{A.13}$$

We map this value to the degree of blue color of each pixel. The larger the positive value, the bluer it is, and the smaller the negative value, the redder it is.

A.1.6 Cartography Details of Fig. 3.8

For Fig. 3.8, we calculate the magnitude of Step 1 as the finite differences of kernel alignment in Fig. 3.3 (Left, "Forerunner Token of Label"). We directly use the head counting of Fig. 3.5 (Middle) and Fig. 3.6 (Left) as the magnitude of Steps 2 and 3. These data are regularized and converted into transparencies.

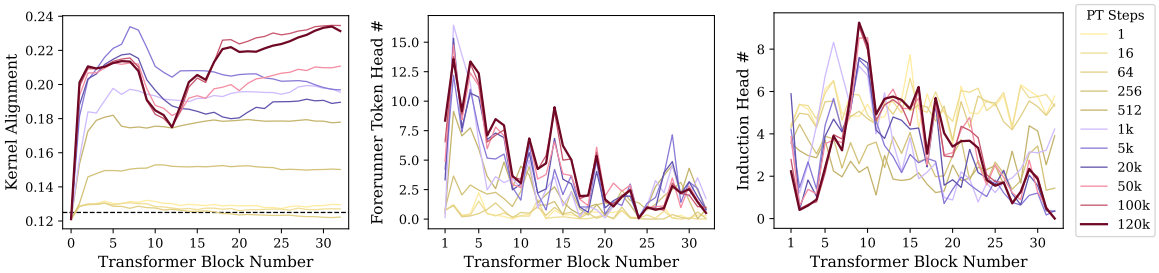


Figure A.4: The 3 operating magnitudes on Pythia-6.9B with various pre-training steps. **Left**: Step 1, Input Text Encode. **Middle**: Step 2, Semantics Merge. **Right**: Step 3, Feature Retrieval and Copy, measured by the correct induction head numbers.

A.1.7 Experiment Setting of Ablation Experiments in Table 3.1

In Table 3.1, we attribute each step of the inference process to specific attention connections, also shown in Fig. A.2. When we aim to remove this step from the inference, we eliminate (i.e. zeroing) all corresponding attention connections from layer 0 to layer $\{25\%, 50\%, 75\%, 100\%\} \times \text{TotalLayers}$.

For reference, for each experiment, we also conduct controlled experiments where the same amount of randomly selected attention connections are removed from the same layers as the controlled values. The controlled results are shown as smaller numbers under the experimental results.

A.2 LM Pre-training Dynamics Measured by ICL Circuit

We extend the discussion of pre-training dynamics in §3.6 here.

One can divide a self-regression model into an early part and a later part in depth of layers, where the early part encodes the input into a hidden representation, and the later part decodes the hidden representation back to the input. So, the training object can also be divided into an encoding loss and a decoding loss. According to the discussion in §3.6, the operation of Step 1 can be classified as encoding, and the other two steps of the induction circuit can be classified as decoding. Intuitively, since the decoding operations require the encoding results as input, unless the encoding operation converges to a stable output, the decoding can not be trained since the input-output mapping is noised, causing unstable gradients to interfere with the training (Liu et al., 2020).

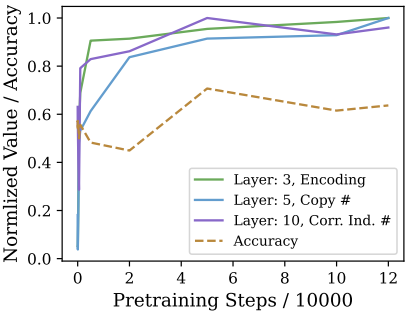


Figure A.3: Operating magnitude for each inference step (normalized) and ICL accuracy against pre-training steps on Pythia-6.9B and SST-2.

We confirm such a hypothesis by measurements in Pythia-6.9B (Biderman et al., 2023) as shown in Fig A.3, where: (1) The magnitude of the 3 operations emerge in the early phase of pre-training (less than 10k steps), and is monotonically increasing, while the encoding operation has the fastest growth rate. (2) ICL capacity appearance after all three operations reaches a high level (around 50k steps, notice that the random accuracy is 0.5), while the curve morphologies of the operating magnitudes against the layer numbers (shown in Fig. A.4) are convergence to the last training step. Such results suggest that: LMs start to produce the inner encoding in the very early steps of the pre-training, and can be an important fundamental in building the subsequent induction circuits, as explained in the previous works mentioned in §3.2.1.

A.2.1 Data Distribution Requirement Explained by Hidden State Multiplex

Moreover, the hidden space multiplexing in the induction operation observed in this chapter can give a prototypical and phenomenological conjecture for the data distribution requirement found in the previous works (Olsson et al., 2022; Reddy, 2024; Singh et al., 2024b), where data with a large label space and various tasks can promote ICL and suppress **I**n-**w**eight **L**earning (IWL), and vice versa. Intuitively, suppose the

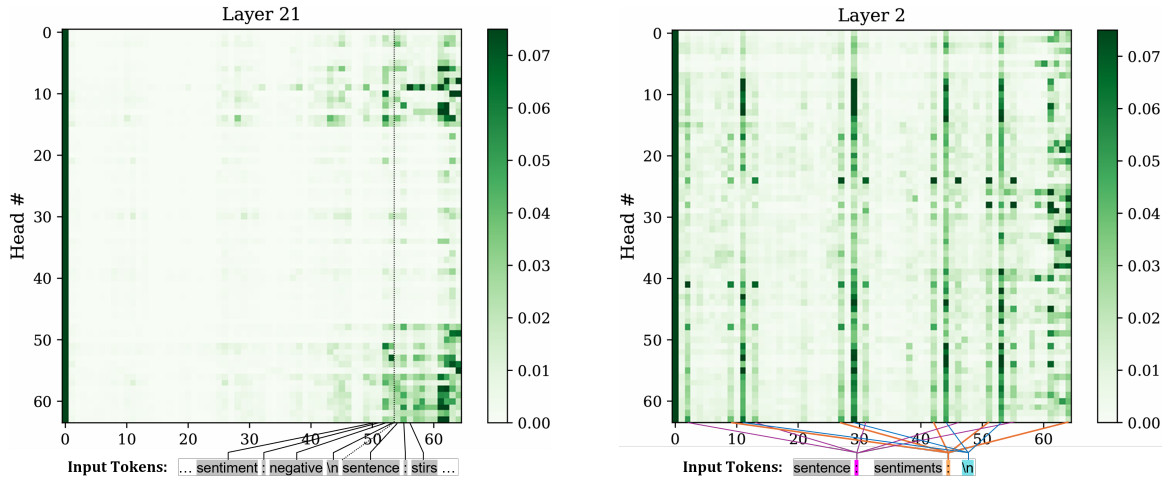


Figure A.5: Attention score visualized with the forerunner token of the ICL query as the attention query, from (**Left**) layer 21 and (**Right**) layer 2 in Llama 3-70B from an input case.

encoding inputted into the later layers is clustered by their labels (similar to the well-embedded input styles in the works above, confirmed in Fig. 3.2). In that case, we can say a cluster center is the **eigen-subspace of the corresponding label**. Since attention only conducts dot-multiplication operations, let us assume that these eigen-subspaces are radially distributed.

During the training, (1) **When the label space is small**, the trained attention heads only need to extract the projected length of the query on each label's eigen-subspace. For each label, such operation has a parameters' analytical solution with (*encoding kernel*) $W_Q^\top W_K = \mathrm{I}$ and (*decoding transformation*) $W_O W_V = o_y^\top e_y$, where o_y is the label token's output embedding, and e_y is the label's eigen-subspace. From such an operation, theoretically, one layer can handle at most $|\mathcal{H}|$ labels, where $|\mathcal{H}|$ is the head amounts. While, considering the sparsity of these eigen-subspaces, such an upper bound can be increased to $d'|\mathcal{H}|$ by multiplex one head to decode an orthogonal group of labels with orthogonal eigen-subspaces of $\mathcal{E} = [e_{y_1}; e_{y_2}; \ldots; e_{y_{d'}}]$ and orthogonal output embedding $\mathcal{O} = [o_{y_1}; o_{y_2}; \ldots; o_{y_{d'}}]$, where d' is the inner dimension of attention head, with decoding transformation $W_O W_V = \mathcal{O}^\top \mathcal{E}$. (2) **When the label space expands**[24], the decoding transformation can not distinguish all the clusters since the $W_O W_V$ is low-rank. Driven by the training loss, the model can choose to transform the encoding kernel to focus on catching the most similar label tokens with the query, and copy the label tokens' information back to the query. As a result, one attention head can catch at most d' groups of label tokens mapped collinearly by the encoding kernel (note that this set of labels may not appear simultaneously in the context, so confusion can be avoided), and the common space of these label words become the induction subspace shown in §3.4.2.

Our other conjecture is that the ICL training endpoint is thermodynamically stable (with a lower loss), and in contrast, the IWL training endpoint is kinetically stable (with a more accessible training trajectory). Moreover, the IWL training object can be a precursor of ICL training, since the total number of labels fed into the model gradually increases with the data. So, we can hypothesize that: When the metastable state is disturbed by some condition, such as the appearance of rare or noisy labels, the training can show a phase transition toward a thermodynamically stable state.

Notice that this section is our hypothesis based on the results of this chapter, the detailed dynamics are still unknown, which should be empirically validated in future work. One can start by decomposing pre-training targets into implicit tasks, and examine how these tasks can evoke the occurrence of the 3-step inference operations. A possible beginning is: finding implicit input-label tuples in wild data.

[24] Notice that such a situation can also occur when the orthogonality between eigen-subspaces or output embeddings is lost. A common situation is that the variance of the cluster increases, creating confusion within the decoding space of the attention head.

A.3 Can LMs Segment ICL Inputs?

The experiments in §3.3.1 imply that LMs conduct effective segmentation on ICL-styled inputs, enabling LMs to block the interference of preceding demonstrations in the input text encoding operation. Here, as a prototypical discussion, we confirm the existence of the segmentation operation and then reveal that such segmentation can be done in very early layers from an attention operation focusing on some specific segmentation tokens in the inputs.

As a preliminary observation, we visualize the attention scores with the last forerunner token as the attention queue at layer 21 (a layer with high encoding magnitude, refer to Fig. 3.3) from an input case, as shown in Fig. A.5 (Left), where most of the attention heads are focused on the query text. The visualization suggests that encoding operations are localized into the query tokens.

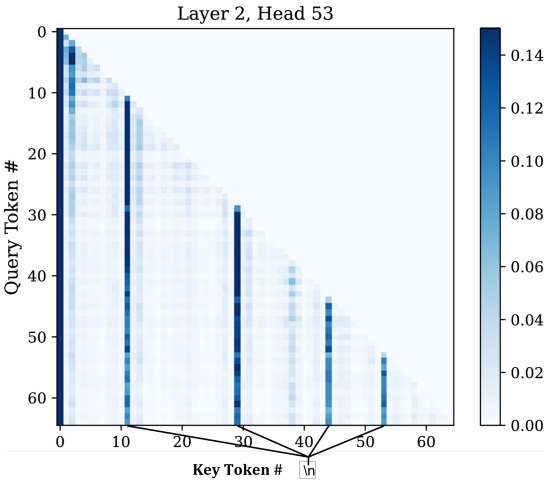


Figure A.6: Attention matrix visualized on layer 2, head 53 in Llama 3-70B from an input case.

Although position embedding inserts sufficient positional information to hidden states and enables attention heads to identify nearby tokens, we believe that position embedding is insufficient to accurately segment the various input parts of uncertain lengths. We hypothesize that LM focused on natural delimiters (e.g. "`\n`", "`: `") in the input during the early stages of inference, and visualization in Fig. A.5 (Right) supports such hypothesis: In layer 2, most of the attention heads focus on the natural delimiters, and another visualization in Fig. A.6 shows that all attention queries (not only the forerunner token) exhibit similar separator-focusing behavior, suggesting that: Some attention heads merge all the preceding delimiters' representation into every token as *delimiter-based positional encoding*, making the representations of tokens with the same number of preceding delimiters similar, while differentiating the representations of tokens with different numbers of preceding delimiters. In the subsequent inference process, LM can utilize these delimiter-based positional encodings for localization operations. Such observation is also consistent with Fig. 3.4.

Furthermore, we empirically demonstrate that delimiters have significant saliency towards ICL accuracies in Table A.2 (upper), experimented by removing them from prompt templates. Interestingly, the trial to completely remove these delimiters from the inputs yielded almost random results, even though these inputs still conform to the primary form of ICL. A reliable reason can be that: The hiatus of the delimiter interferes with the encoding operation (Step 1) on both demonstrations and queries, which completely disrupts the ICL process.

Table A.2: Accuracy drop with delimiters removed/modified from prompts on SST-2.

Template Modification	**Acc.** (%)
None (Table A.1)	91.60
- w/o "`\n`"	93.36
- w/o "`: `"	85.74
- w/o "`sentence: `","`sentiment: `"	79.10
- w/o all above	50.98
- "`: `" → "`hello `"	78.91
- "`: `" → "`@ `"	91.41
- "`: `" → "`positive `"	71.48
(Random)	50.00

The scale of such a segmentation operation can surprise one, since more than half of the heads focus on the segmenting operation as shown in Fig. A.5 (Right). However, as an assumption, we want to argue that dividing the input text into local segmentation is a crucial step in language modeling, so, functionally, LM has sufficient "motivation" to focus on segmenting by the pre-training objective. Moreover, based on the above principles, as long as the delimiter appears periodically at appropriate positions and can be captured by attention heads (only in the structured parts of the prompt template), as shown in Table A.2 (lower), the delimiter can be designed to any token. While we still recommend natural delimiters without semantics in the template design.

A.4 Semantics Merge is Non-selective on Location

To investigate whether the copy processing described in §3.4.1 has selectivity on the forerunner token, on every attention head of each layer, for a given token position i, we measure the **N**ormalized **C**opy **M**agnitude shown below:

$$\mathrm{NCM}_{n_t}(\alpha) = n_t \alpha_{(i-1)\to i}, \tag{A.14}$$

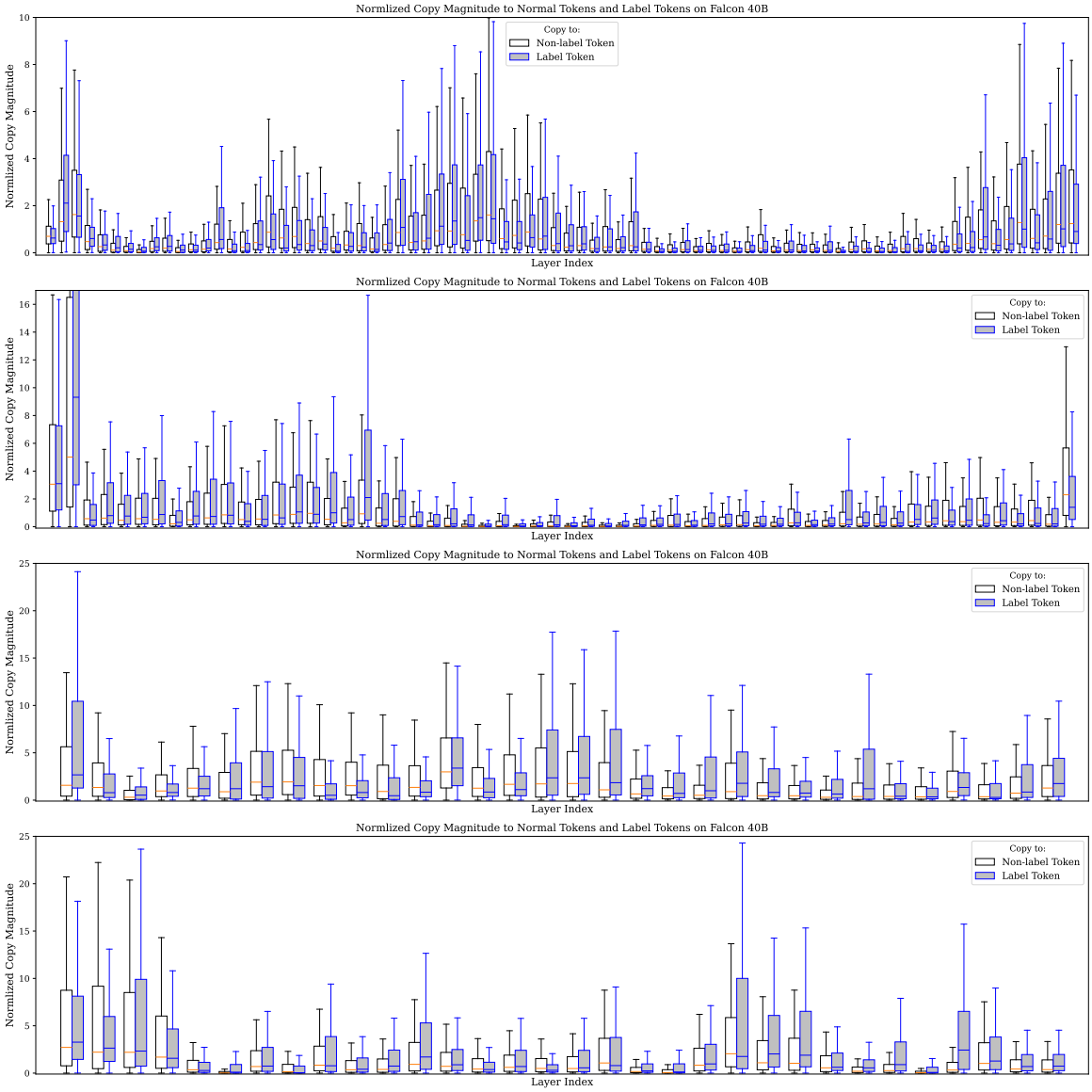


Figure A.7: Copy magnitude normalized by the sequence length from the previous token to the non-label token and label token, for every token and head in each layer. Significant statistical differences cannot be observed.

where the $\alpha_{(i-1)\to i}$ is the attention score with i-th token serving as the attention query and $(i-1)$-th token serving as the attention key. For each layer, we export the NCM at all positions and on all attention heads, and separately statistics the cases where the i-th token is a label token or a non-label token. The results for 4 models on SST-2 are shown in Fig. A.7.

From the results, no significant statistical differences between these two types of tokens can be observed, suggesting that the semantics copy process, which is identified as Step 2 of our circuit, is not selective on the token types. However, even if we demonstrate that the model cannot exhibit selectivity in the copy positions within the current input, a potential direction for future research is to investigate whether any forerunner tokens in the inputs can enhance or weaken this copy process. Given that this copying is known to be related to label denoising (§3.4.1), exploring and carefully designing these forerunner tokens could significantly benefit the control of ICL behavior.

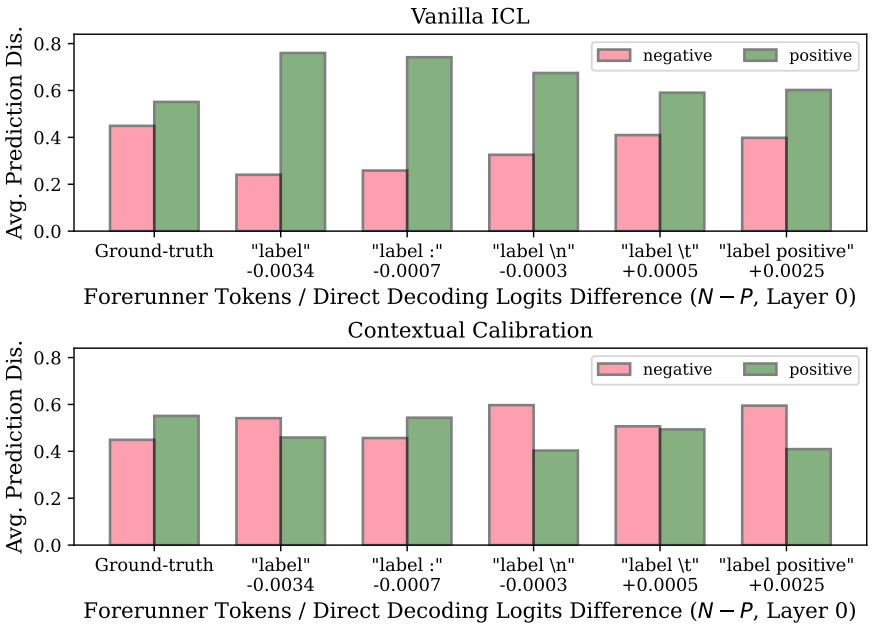


Figure A.9: Predicting distributions on different forerunner tokens with various direct decoding logits. Inference process used: **Upper**: Vanilla ICL, **Lower**: Biased removed inference by Contextual Calibration proposed by Zhao et al. (2021).

A.5 Measurement on Direct Decoding

This section measures the direct decoding bypass, suggesting that: (1) Direct decoding on well-processed hidden states with some later layer skipped can get satisfactory accuracy even better than the full inference process. (2) Direct decoding on insufficient processed hidden states adds bias towards the predicting distribution.

We examine the first claim by applying the language model head on each layer's hidden state on SST-2, Llama 3-8B, and conduct a standard ICL process on the decoded token prediction distribution. The results are shown in Fig. A.8, where direct decoding accuracy emerges from random to near 1 around layer 18. Refer to Fig. 3.8 and results in Appendix A.8.2, we can confirm: Accuracy emerges after all three steps are executed. Moreover, the accuracies on the intermediate hidden states are even higher than the last hidden states, which is aligned with the discussion in Table 3.3. So, we can conclude: Direct decoding on well-processed hidden states can classify well.

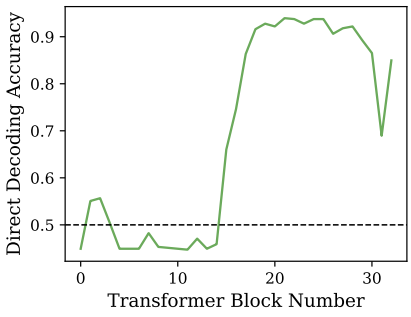


Figure A.8: Direct decoding accuracies on various layers.

Moreover, we infer that direct decoding from lower layers, where hidden states are not sufficiently processed, causes prediction bias. We investigate the influence of the direct decoding result of layer 0, by the relationship between direct decoded distribution and final prediction distribution. In detail, on SST-2 and Llama 3-8B, we use various forerunner tokens with different direct decoding distributions on the label tokens "positive" and "negative", and calculate their ICL prediction probability distributions respectively, as shown in Fig. A.9 (Upper), where forerunner tokens with biased direct decoding distribution produce prediction biases with the same tendencies. While, when we apply contextual calibration (Zhao et al., 2021), which removes the background value without a query from the prediction, such similar tendencies disappear (Fig. A.9 (Lower)).

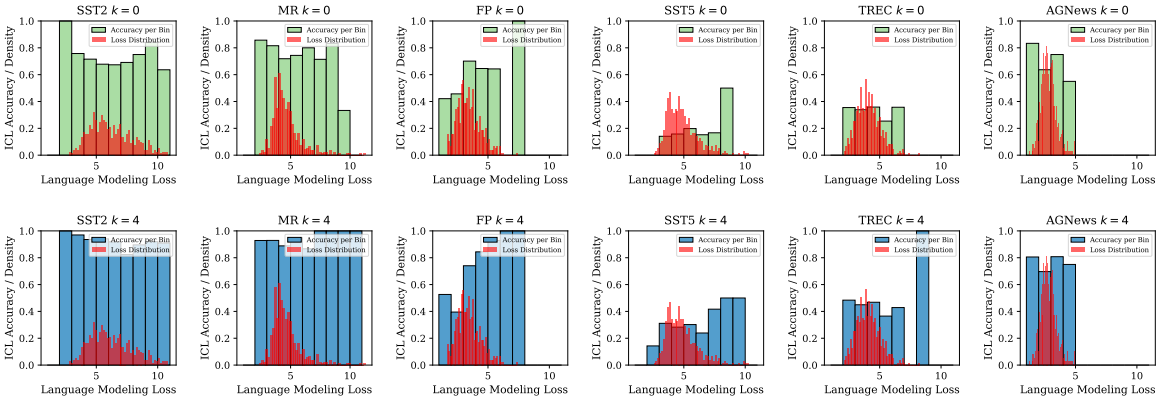


Figure A.10: The correlations between language modeling loss and ICL prediction accuracies. **Upper**: 0-shot results; **Lower**: 4-shot results.

A.6 Demonstrations Enhance the Inference of Perplexed Queries

We investigate the correlation between the queries' perplexities and the classification accuracies with and without demonstrations, as a supplement of results in Fig. 3.3 (Right). We divide the queries into 10 bins w.r.t. the language modeling loss, and calculate the prediction accuracy in each bin, shown in Fig. A.10. In these results, although a unified correlation can not be observed, we can confirm that: Compared to the 0-shot results, the 4-shot inference shows better accuracies, especially on queries with high language modeling loss. So, we can conclude that: Demonstrations enhance the inference accuracy of perplexed queries, consistent with the results in Fig. 3.3 (Right).

A.7 Degradation Analysis: In-weight Learning without Ground-truth Label in Context

Given the circuit proposed in this chapter, it is intuitive that some label tokens, especially the ground-truth label token of the query should be presented in the demonstrations, which is the typical in-context learning setting compared to the **I**n-**W**eight **L**earning (IWL) setting (Chan et al., 2022; Reddy, 2024) where the ground-truth label is not offered in the demonstration. When the ground-truth label is missing, the behavior of Steps 1 and 2 remains constant as they are independent of the query (due to the causal attention mask), while the behavior of Step 3 should be re-discussed.

In this section, we demonstrate that our circuit, especially the induction processing of Step 3, can still explain the inference behavior in the IWL scenario, but is not as robust as in the ICL scenario. Moreover, our experiments demonstrate that in the IWL scenario, induction heads are detrimental to the model's predictions, which is consistent with the conclusion drawn from the results in the main text. Also, some inference phenomena in the IWL scenario can be explained by our conclusions. While, we also illustrate through a counterexample that under the condition of IWL, even if the induction heads work negatively, the demonstrations can still enhance the inference.

Experimental results in this section require us to consider new inference dynamics for IWL scenarios, although we believe the discussion for ICL scenarios in this chapter is robust. We demonstrate that such separate consideration of ICL and IWL is reasonable and responsible by highlighting the essential differences between ICL and IWL settings, as also shown in previous works (Chan et al., 2022; Reddy, 2024).

A.7.1 How Accurate can Induction Heads Explain IWL Inference?

Our main concern is: Whether the induction heads in our circuit for ICL can explain or predict the inference behavior in the IWL scenario. We design a metric to investigate the accuracy of such an explanation: We calculate the divergence between the real output probability of the model and the predicted output probability through the attention assignment of the induction heads. In detail, given a label space $\mathbb{Y}$, for each label $l \in \mathbb{Y}$,

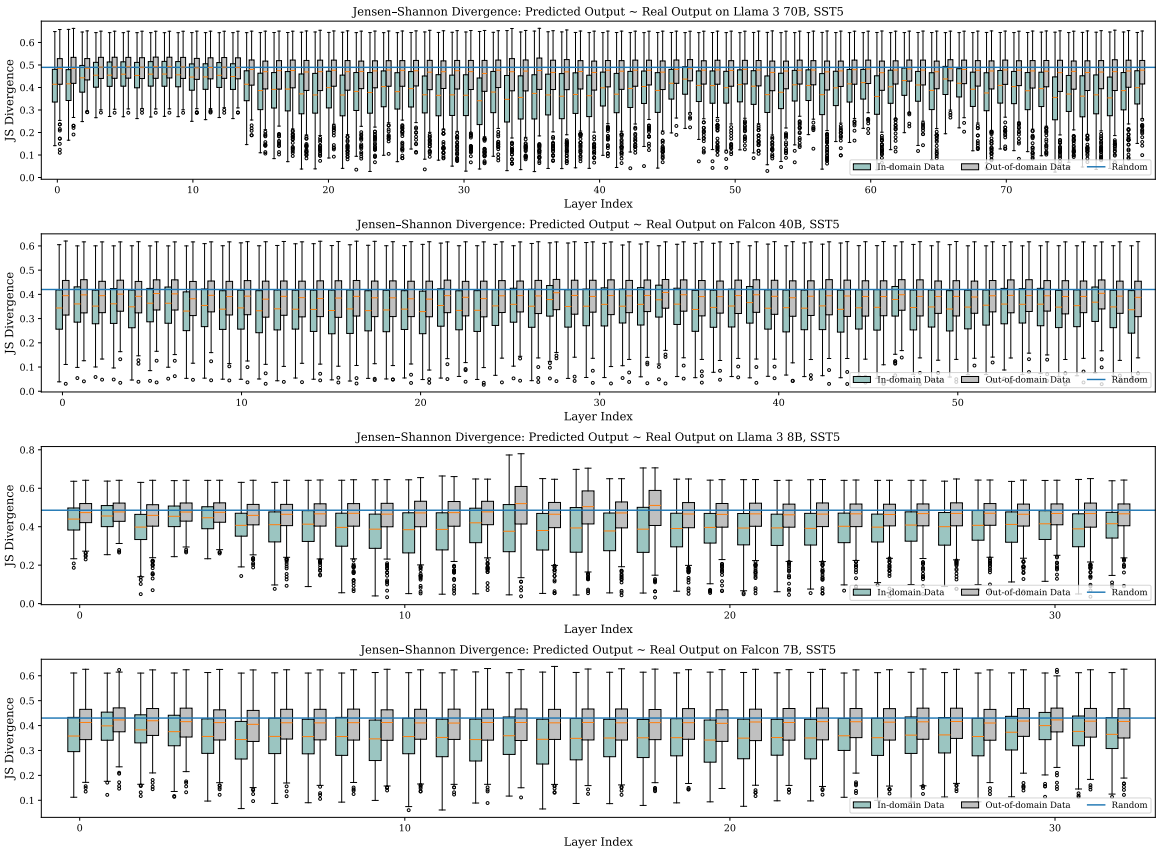


Figure A.11: The JS divergence between the predicted output by the induction attention score (Eq. A.15) and real output on ICL and IWL settings. Since not all attention heads are induction heads, the lowest 5 data of each layer is selected.

we denote the token-index set of l as $\mathcal{L}_l$. Then, we calculate the predicted output probability $\hat{o}$ as:

$$\hat{o} = \text{softmax}\left(\left[o_1, o_2, \ldots, o_{|\mathbb{Y}|}\right]\right), \tag{A.15}$$

$$o_l = n_t \sum_{i \in \mathcal{L}_l} \alpha^h_{i \to q}, \tag{A.16}$$

where the $\alpha^h_{i \to q}$ is the attention score of head h where the query's forerunner token serves as the attention query, and the i-th token as the attention key. Given the real output distribution of ICL as o, we calculate the Jensen–Shannon divergence $D_{\text{JS}}[\hat{o}, o]$ as the metric for the accuracy of predicting o from $\hat{o}$. For each Transformer layer, we select the results of the 5 attention heads with the lowest divergence, and the results on SST-5[25] are shown in Fig. A.11.

In the results, statistically, compared to the ICL setting, induction heads provide poorer explanations of model predictions in the IWL setting. However, these divergences consistently remain below the random baseline, indicating that model outputs in the IWL setting can still be weakly explained by induction heads. This indicates that the circuit utilized in this chapter aligns with the IWL scenario, albeit slightly weaker.

A.7.2 Do Induction Heads Contribute to IWL Processing?

To test the generalizability of our conclusion towards IWL scenarios, we conduct more ablation experiments similar to §3.5.1, but with filtered inputs with no ground-truth query label presented in the context. The results are shown in Table A.3. Generally speaking, these experiments confirm that: Our theoretical framework is still able to explain inference phenomena in IWL scenarios, thus strengthening our framework.

[25] We select a more-way task to reduce the influence of frequency bias of ICL (introduced in §3.5.3).

Table A.3: Accuracy variation (%) with each inference step ablated on **Left**: Llama 3-8B, **Right**: Falcon-7B **in IWL scenarios**. Notations are the same with Table 3.1, significant comparisons are highlighted with arrows.

#	Attention Disconnected Key → Query	Affected Layers Ratio (from layer 1) 25%	50%	75%	100%
1	**None** (4-shot IWL baseline)	±0 (Acc. 33.43 ↓)			
	– Step1: Input Text Encode –				
2	**Demo. Texts** x_i → **Forerunner** s_i	−2.31 +0.03	−17.90 +0.78	−28.58 +0.49	−31.22 −0.10
3	**Query Texts** x_q → **Forerunner** s_q	−2.67 +0.23	−12.96 +0.23	−17.87 −0.46	−13.41 +0.26
	– Step2: Semantics Merge –				
4	**Label** y_i → **Query Forerunner** s_q	−1.20 +0.29	+0.13 ↑ −0.10	+0.52 ↑ −0.06	+0.88 ↑ −0.06
	– Step3: Feature Retrieval & Copy –				
5	**Label** y_i → **Query Forerunner** s_q	+2.08 ↑ −0.23	+2.05 ↑ −0.06	+5.40 ↑ −0.36	+22.95 ↑ +0.00
	Reference Value				
6	**Zero-shot**	+17.22 (Acc. 50.65)			
7	**Random Prediction**	−0.93 (Acc. 32.50)			

#	Attention Disconnected Key → Query	Affected Layers Ratio (from layer 1) 25%	50%	75%	100%
1	**None** (4-shot IWL baseline)	±0 (Acc. 27.41 ↓)			
	– Step1: Input Text Encode –				
2	**Demo. Texts** x_i → **Forerunner** s_i	−17.64 −0.13	−24.32 −0.42	−26.17 −1.24	−26.43 +0.91
3	**Query Texts** x_q → **Forerunner** s_q	−3.19 −0.33	−15.33 −0.23	−21.48 +0.03	−22.53 +0.32
	– Step2: Semantics Merge –				
4	**Demo. Forerunner** s_i → **Label** y_i	+2.93 ↑ −0.00	+1.10 ↑ +0.32	+6.38 ↑ 0.13	+5.86 ↑ −0.13
	– Step3: Feature Retrieval & Copy –				
5	**Label** y_i → **Query Forerunner** s_q	+8.20 ↑ +0.13	+5.63 ↑ 0.23	+9.28 ↑ +0.00	+29.26 ↑ +0.10
	Reference Value				
6	**Zero-shot**	+33.58 (Acc. 60.99)			
7	**Random Prediction**	+5.09 (Acc. 32.50)			

In detail, compared to the results of unfiltered inputs (Table 3.1), we highlight 4 major phenomena, and explain them within our framework: (1) The baseline accuracies (line 1) in IWL settings are lower than in normal settings, even worse than random prediction. This is easy to understand given the frequency bias introduced in §3.5.3. (2) As we gradually suppress the induction heads (line 5), the accuracies increase, which is consistent with the conclusion in the main text of this chapter: In IWL setting, induction heads can not find and copy any correct label-related information in the context, and only copy the label information presented in the context noisily. Therefore, after removing it from the inference process, the accuracies increase and approaches a 0-shot level. These results confirm our main claim of this section: Induction heads are detrimental to IWL inference. (3) Interestingly, removing the copying processing (line 4) unexpectedly enhances the accuracy. We infer that the input texts of different labels under the same task still have a certain background-value similarity of the encoding at the forerunner token. Such similarities are propagated into the label tokens by Step 2, enhancing the attention flow from labels to the query's forerunner token in Step 3, causing a decrease in accuracy. (4) Removing the demonstration text encoding (line 2) decreases the accuracy to around 0. According to our discussion about Step 1, in this case, forerunner tokens cannot receive information from the demonstration text, leading an equivalent sequence for the later layers without demonstration text like $[s_1][y_1][s_2][y_2]\dots[s_k][y_k][x_q][s_q][y_q]$. Given that the forerunner token $[s.]$ is consistent throughout the input, later layers are likely to conduct induction only on the forerunner token, whose subsequent tokens do not contain the correct label, leading to a 0 accuracy. This is also a critical rebuttal to previous work (Wang et al., 2023), which believes the label tokens directly collect the information of input text: If that is the case, then disconnecting the links from demonstration texts to the forerunner tokens will not have such a significant disturbance on the accuracy.

In summary, these results indicate: Even in the IWL setting, this chapter's conclusions can explain the inference phenomena, and the findings in our work are also significantly enhanced. This extends the applicability of our conclusions, while, given the existence of some counterexamples shown below, we still hope that future work can provide a more detailed discussion of the inference dynamics under the IWL scenario.

A.7.3 The Counterexample

Given the demonstration shown in Fig. A.12, we use the query "Geoffrey Hinton", whose ground-truth label is presented in the context as "Researcher" as the ICL example, and the query "Michael Jordan", whose ground-truth label "Athlete" is NOT presented in the context as an IWL example. Here, to investigate the behavior of the induction heads, we input both examples to Llama 3-70B, and visualize the attention scores from the query's forerunner token (which serves as the attention query) on layer 31, which is identified as the layer with the highest induction magnitude, as shown in Fig. A.12. In the right part of the figure for the IWL scenario, the attention magnitude directed toward

Table A.4: Label probabilities from the model predictions of the ICL (query: "Geoffrey Hinton" (Researcher)) and IWL (query: "Michael Jordan" (Athlete)) scenario shown in Fig. A.12. In this case, IWL predictions also benefit from demonstrations.

Label Token		" Ath#"	" Research#"	" Singer"	" Politician"
IWL	3-shot	**1.00**	0.00	0.00	0.00
	0-shot	**0.89**	0.04	0.01	0.06
ICL	3-shot	0.00	**1.00**	0.00	0.00

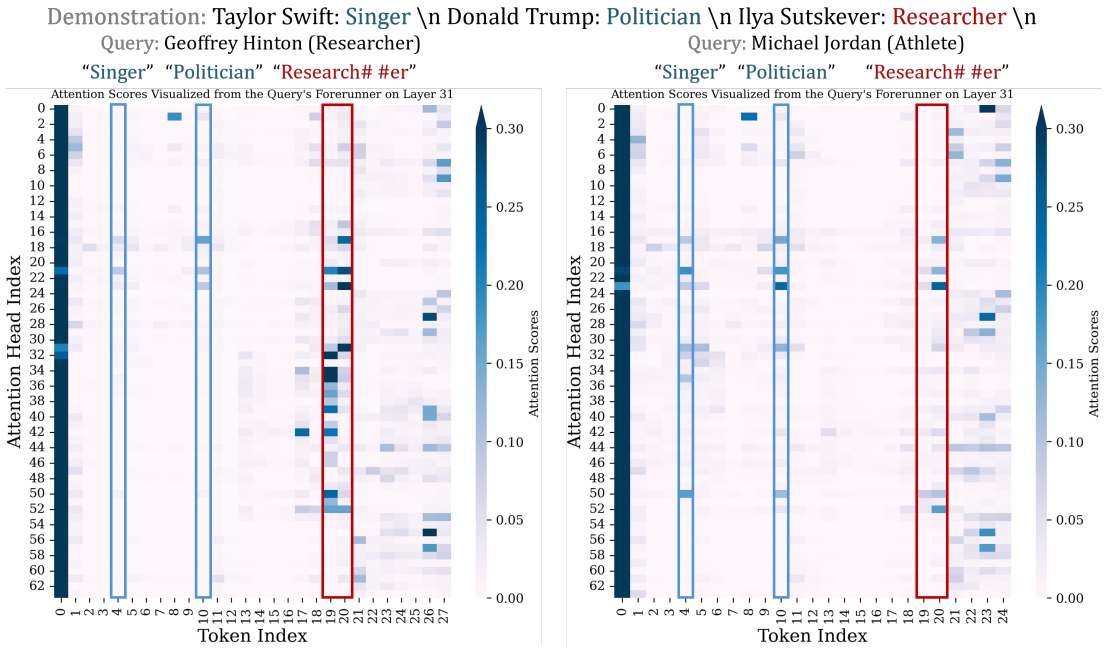


Figure A.12: A counterexample when the induction head's behavior cannot predict the LM's inference behavior. Given the demonstration shown in the figure, the attention scores from the label's forerunner token are visualized on layer 31 of Llama 3-70B. The **left** part is a standard ICL scenario where the ground-truth label of the query can be accessed in the demonstrations. The **right** part is the IWL scenario where the ground-truth label of the query is not presented in the demonstrations. A clear induction pattern can not be observed in the IWL scenario.

the labeled tokens is significantly weak, with most of the attention scores being absorbed by the Attention Sink (Xiao et al., 2024) of the first token. A comparison with the left part, where a ground-truth label is given in the context can particularly highlight such an observation. Such an observation is currently aligned with our expectations since no label features similar to the query's forerunner token can be accessed in the IWL input.

In other words, the induction heads are almost not writing demonstration-relevant information to the query in the IWL scenario. It is intuitive to infer that the model cannot predict the label for "Michael Jordan" well, and the demonstrations cannot help the prediction either. However, as shown in Table A.4, the model produces good predictions and benefits from the demonstration, which contradicts our expectations. Such a contradiction indicates that even if our inference circuit can robustly explain the inference behavior in the ICL scenario, it can not generalize to the IWL scenario generalizatively.

A.7.4 Discussions: The Possibility of Different Inference Dynamics in IWL Scenarios

Our explanation for this counterexample is: It can be considered that in the IWL settings, some different inference dynamics contribute to the prediction. Such a difference is natural, since even if the ICL and IWL input data share a consistent format, they are fundamentally distinct, and sometimes even antagonistic, as shown in previous works (Chan et al., 2022; Reddy, 2024), which find that toy Transformers are difficult to perform well on both types of data with the same set of parameters. While, large models may allow for the coexistence of multiple inference dynamics, as discussed in §3.5.2, making LLMs able to yield better performance on both ICL and IWL inference scenarios.

Therefore, although we claim that our circuit can explain inference behavior under some of the IWL scenarios, it is reasonable to consider ICL data and IWL data separately, and this work conducts a robust analysis under ICL conditions, leaving the IWL scenario for future works.

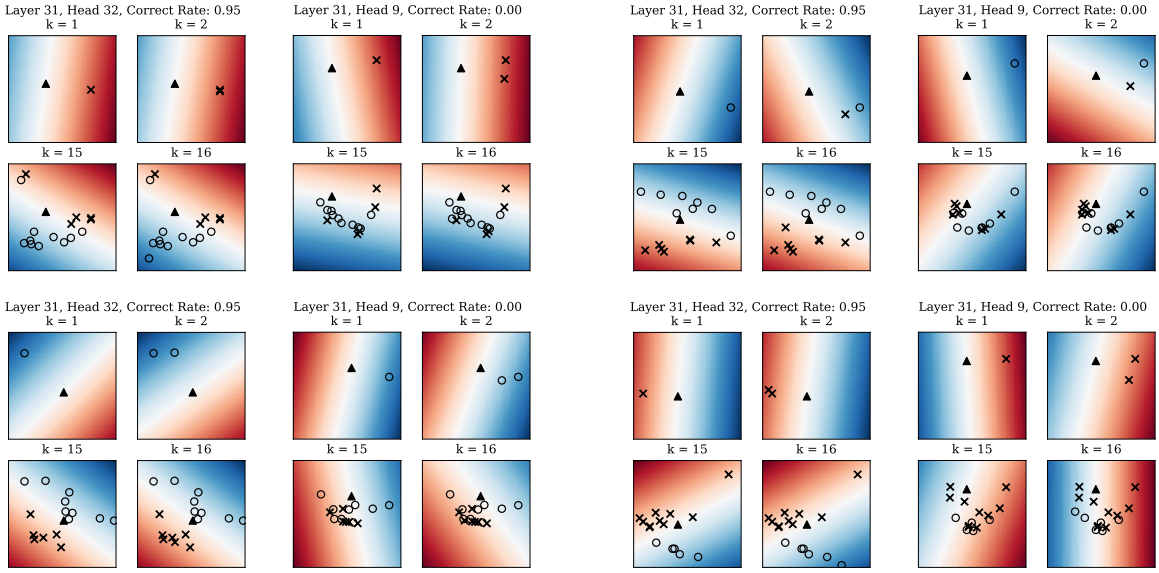


Figure A.13: Supplemental experiment result on more samples for Fig. 3.7.

A.8 Augmented Experiment Results

A.8.1 Attention Head Statistics

We count the marked times of each attention head as Forerunner Token Head (Fig. A.14, A.15, A.16, A.17) / Correct Induction Head (Fig. A.18, A.19, A.20, A.21) by all data samples from each dataset on each model.

Forerunner Token Head Statistics. We plot the distributions of the marked Forerunner Token Heads towards correct and wrong labels: There are observable morphological differences in figures across different datasets, while the forerunner token heads marked on the correct and incorrect labels of the same dataset are almost identical. The detailed data confirms our conclusion in §3.4.1.

Induction Head Statistics. We plot the distributions of the marked Correct Induction Heads, where there are observable morphological differences in figures across different datasets, but also significant overlaps, which confirms our conclusion in §3.4.2.

A.8.2 Other LMs' Experiment Results

The results of most experiments in the main text on Llama 3-8B are shown in Fig. A.22, A.25, A.28, and A.31; The results of most experiments in the main text on Falcon-40B are shown in Fig. A.23, A.26, A.29, and A.32; The results of most experiments in the main text on Falcon-7B are shown in Fig. A.24, A.27, A.30, A.33, and Table A.5.

From these results, we can conclude consistently with the main text. However, as discussed in §3.5.2, inference dynamics on these models are delocalized, thus clear serialization of the 3 steps can not be observed in these results.

Table A.5: Results of Table 3.1 on Falcon-7B.

#	**Attention Disconnected** Key → Query	**Affected Layers Ratio** (from layer 1) 25%	50%	75%	100%
1	**None** (4-shot baseline)	±0 (Acc. 65.27)			
	– Step1: Input Text Encode –				
2	**Demo. Texts** x_i → **Forerunner** s_i	−7.65 −0.68 ± 0.07	−15.69 −0.62 ± 0.07	−27.15 +0.08 ± 0.03	−29.10 −0.36 ± 0.07
3	**Query Texts** x_q → **Forerunner** s_q	−8.30 −0.16 ± 0.00	−21.13 −0.15 ± 0.00	−28.84 +0.11 ± 0.01	−31.74 −0.15 ± 0.02
	– Step2: Semantics Merge –				
4	**Demo. Forerunner** s_i → **Label** y_i	−1.01 +0.36 ± 0.10	−1.92 −0.00 ± 0.00	−1.04 +0.06 ± 0.00	−1.27 +0.06 ± 0.02
	– Step3: Feature Retrieval & Copy –				
5	**Label** y_i → **Query Forerunner** s_q	+3.32 +1.72 ± 3.94	−3.61 −0.03 ± 0.00	−7.91 −0.00 ± 0.00	−5.92 +0.10 ± 0.00
	Reference Value				
6	**Zero-shot**	−4.28 (Acc. 60.99)			
7	**Random Prediction**	−32.77 (Acc. 32.50)			

A.8.3 More Results of Fig. 3.7

To enhance the persuasiveness, we additionally and randomly try 4 input samples as supplements to Fig. 3.7 on SST-2 and Llama 3-70B as shown in Fig. A.13. From these results, we can observe similar phenomena to Fig. A.13 and conclude consistently.

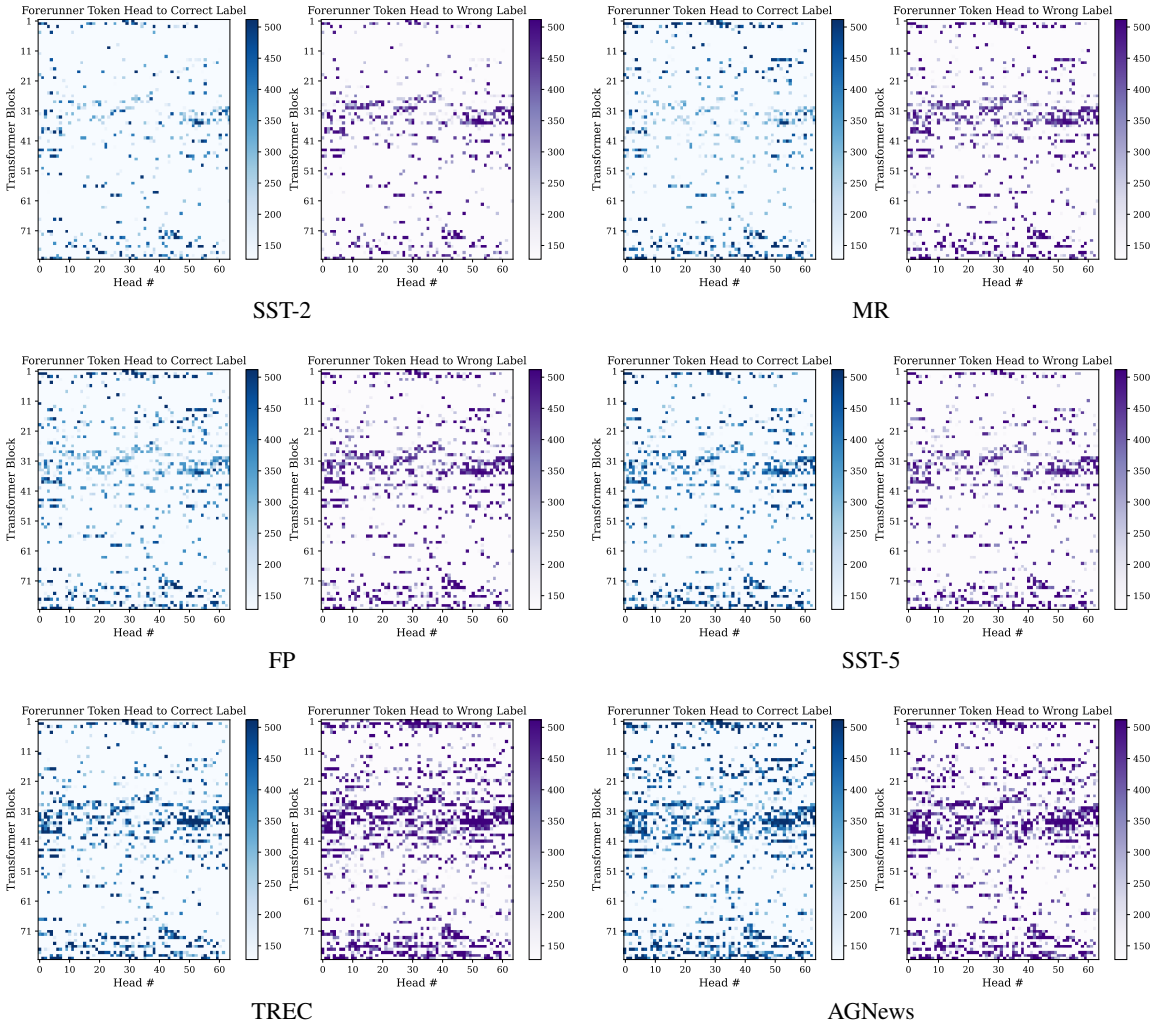


Figure A.14: Forerunner Token Head marked on Llama 3-70B.

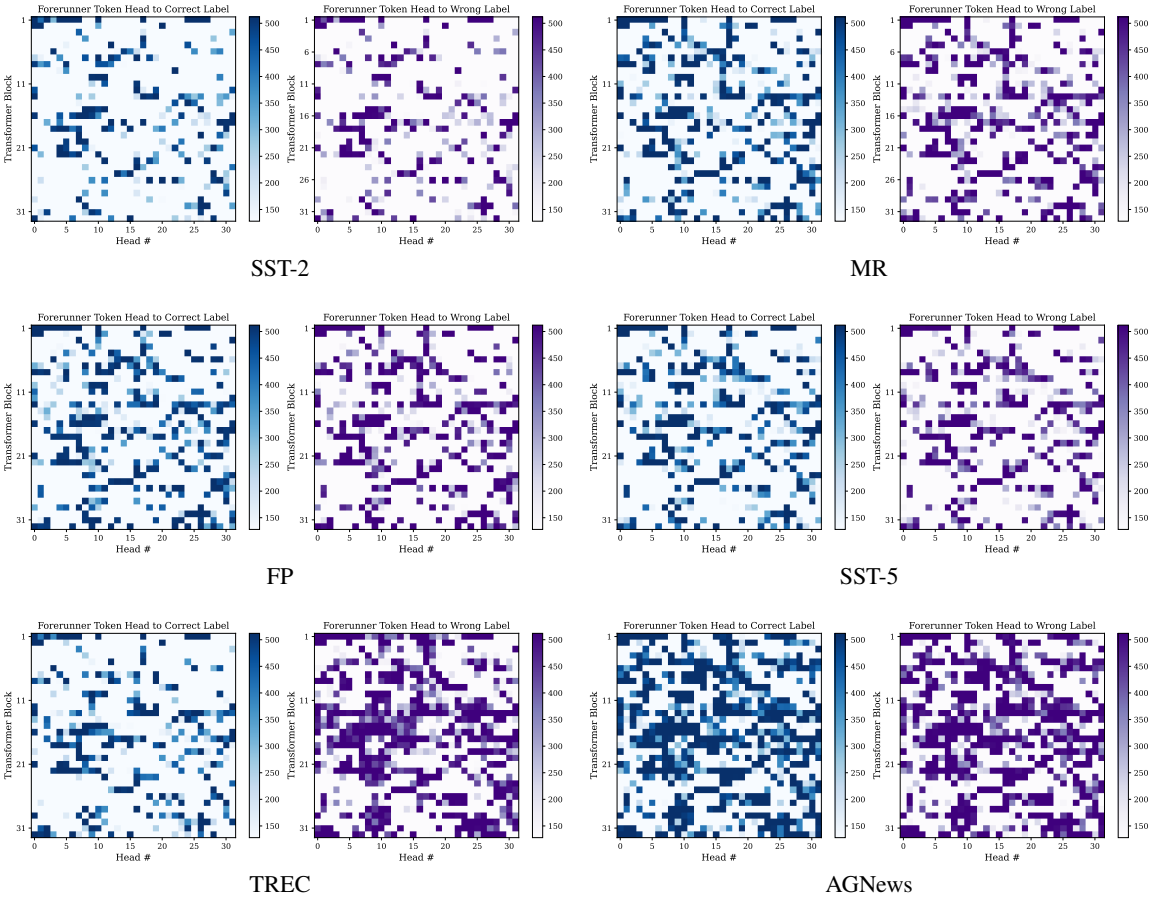
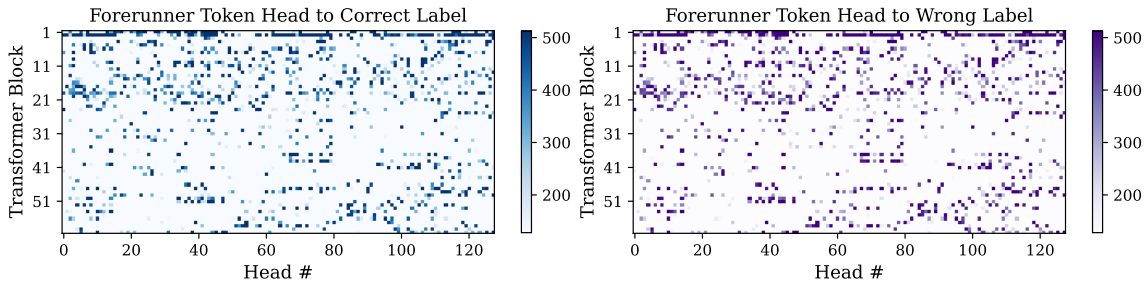
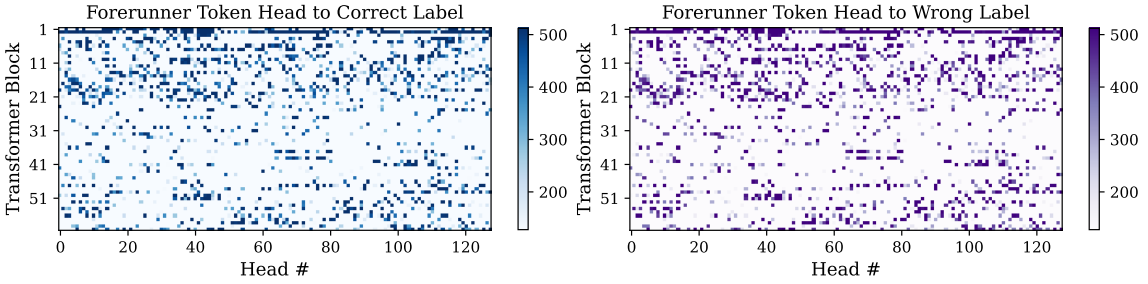


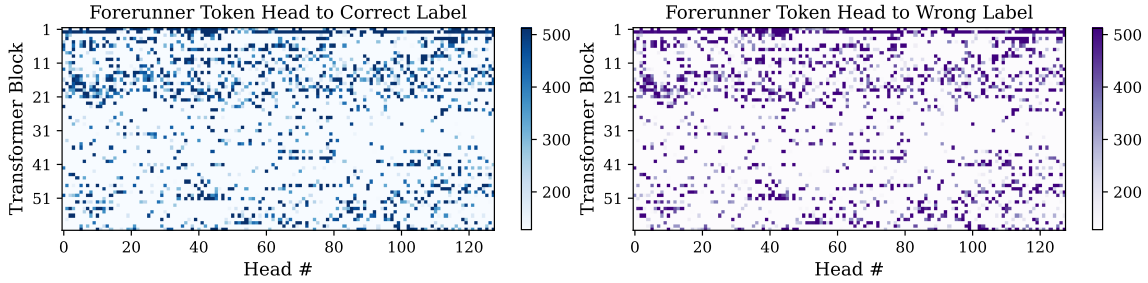
Figure A.15: Forerunner Token Head marked on Llama 3-8B.



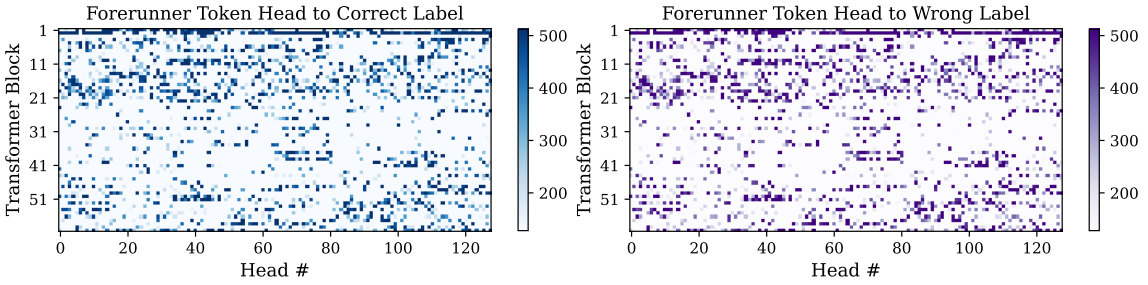
SST-2



MR



FP



SST-5

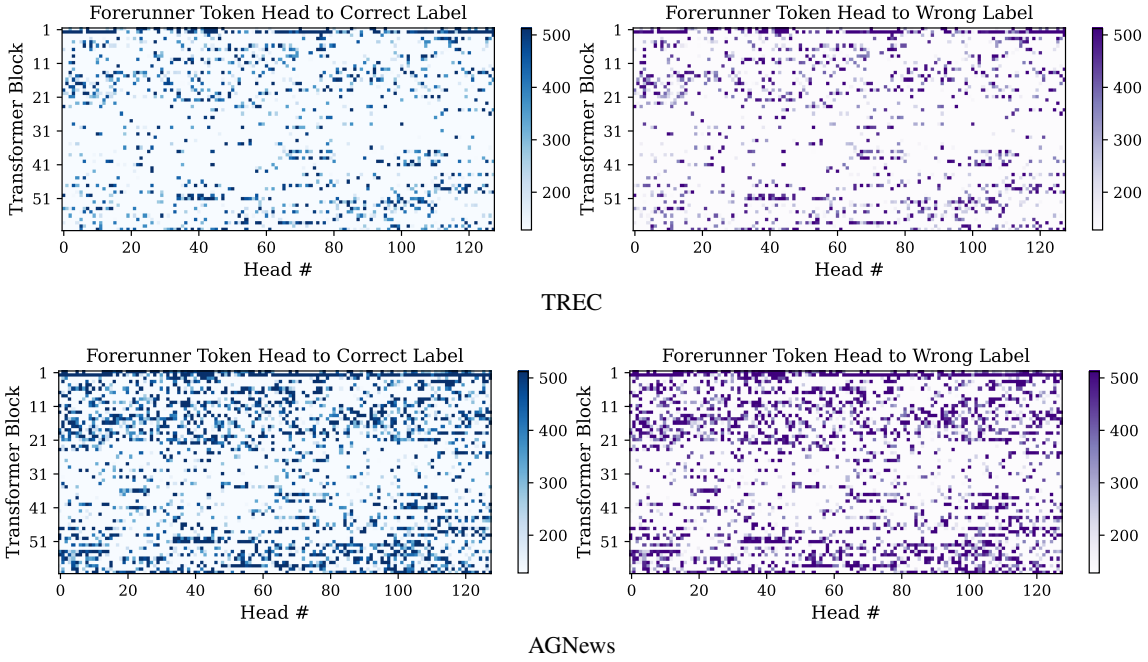
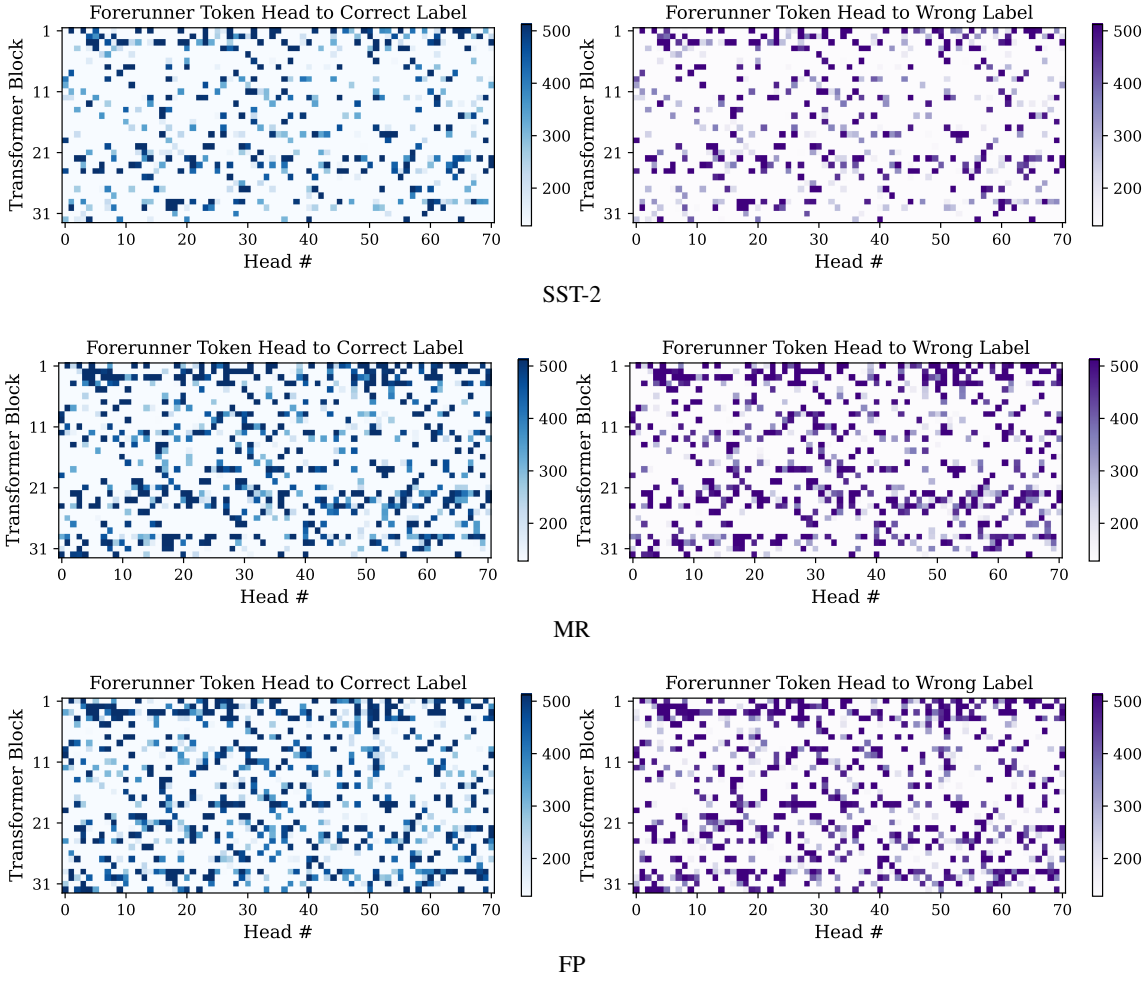


Figure A.16: Forerunner Token Head marked on Falcon-40B.



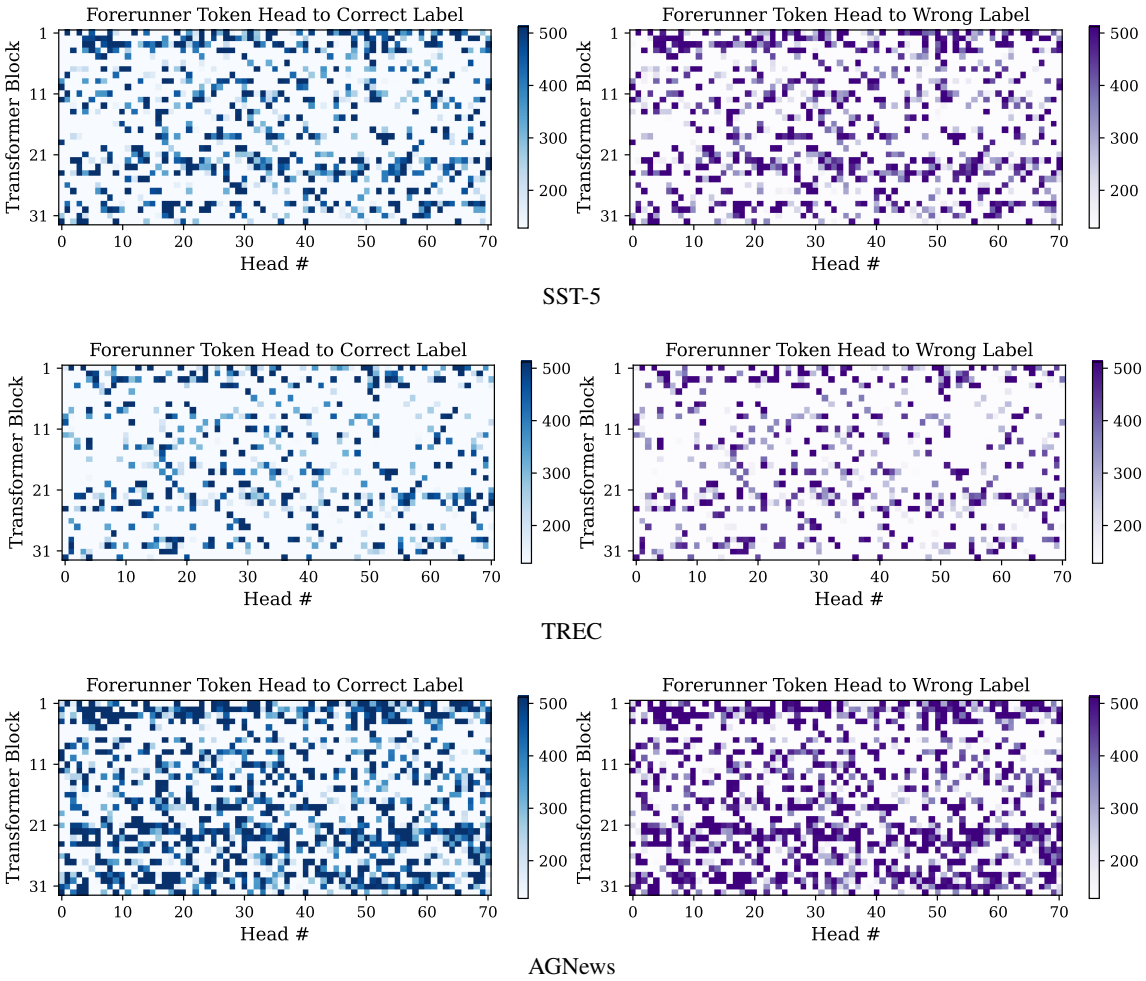


Figure A.17: Forerunner Token Head marked on Falcon-7B.

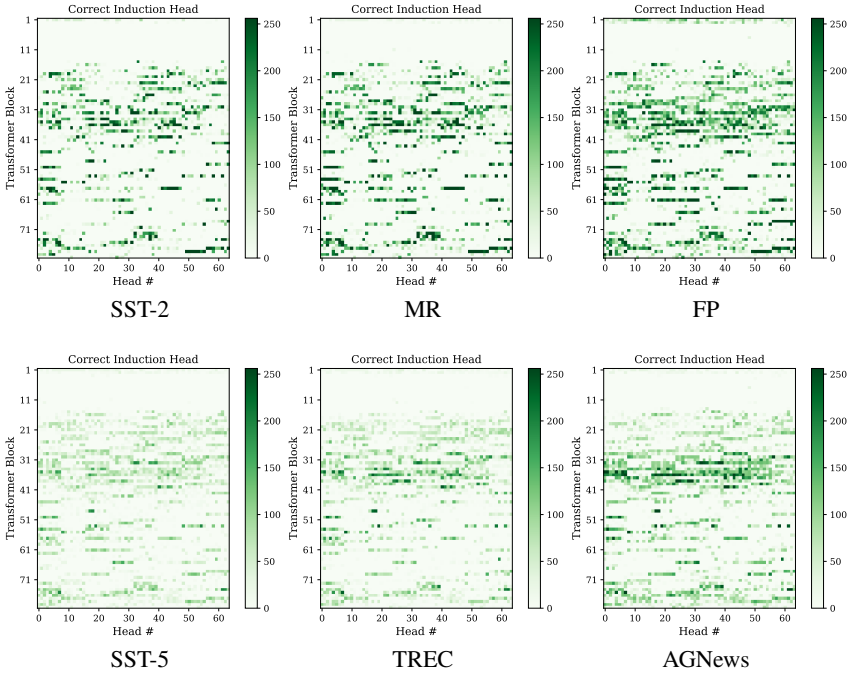


Figure A.18: Correct Induction Head marked on Llama 3-70B.

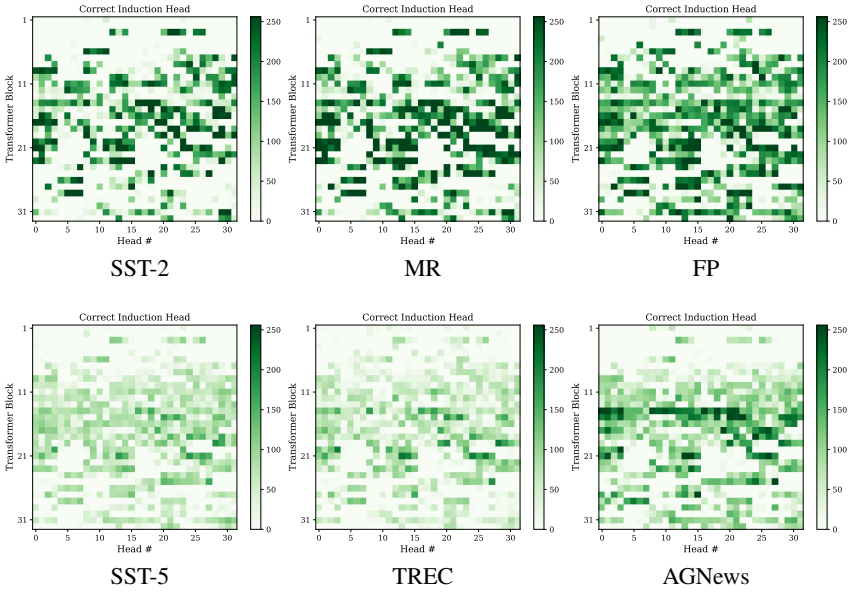


Figure A.19: Correct Induction Head marked on Llama 3-8B.

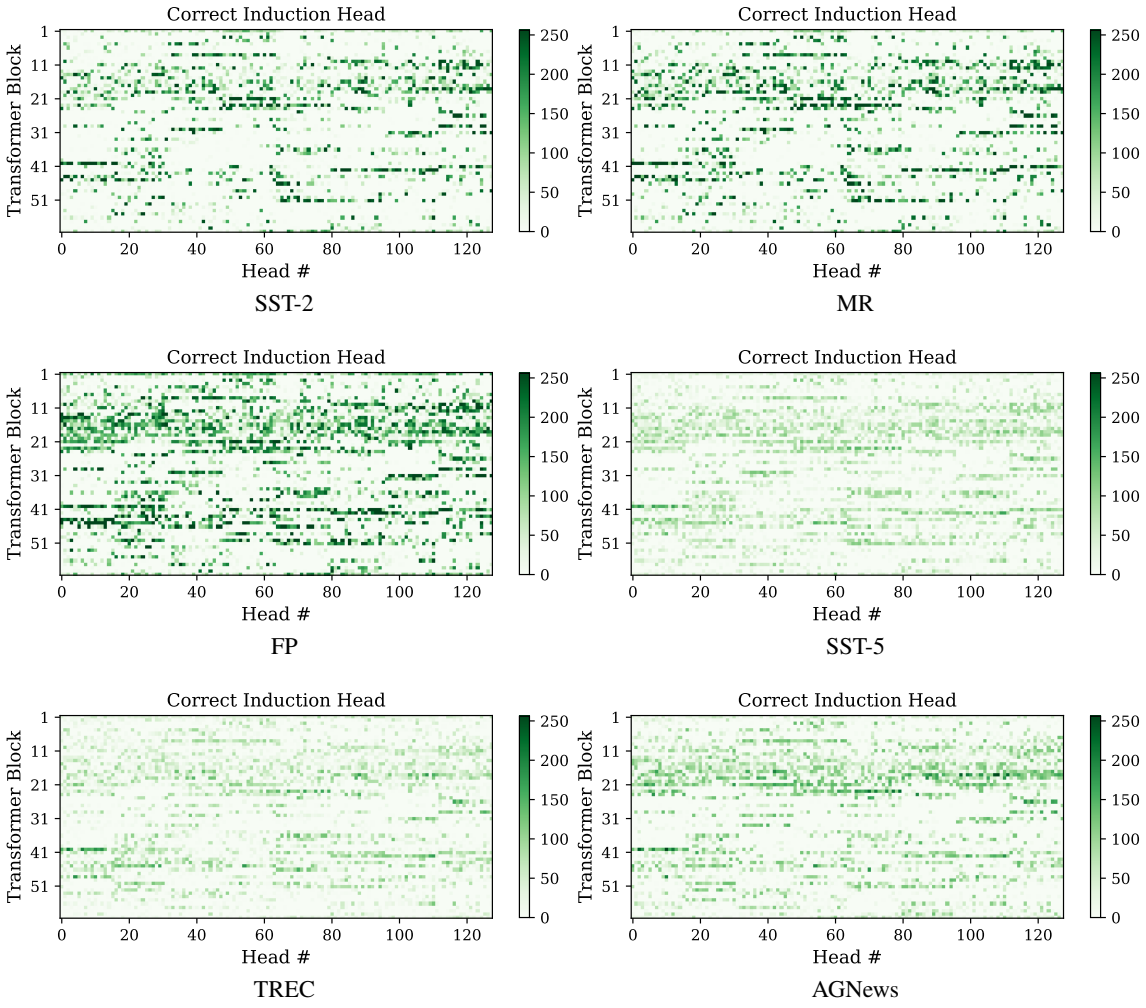


Figure A.20: Correct Induction Head marked on Falcon-40B.

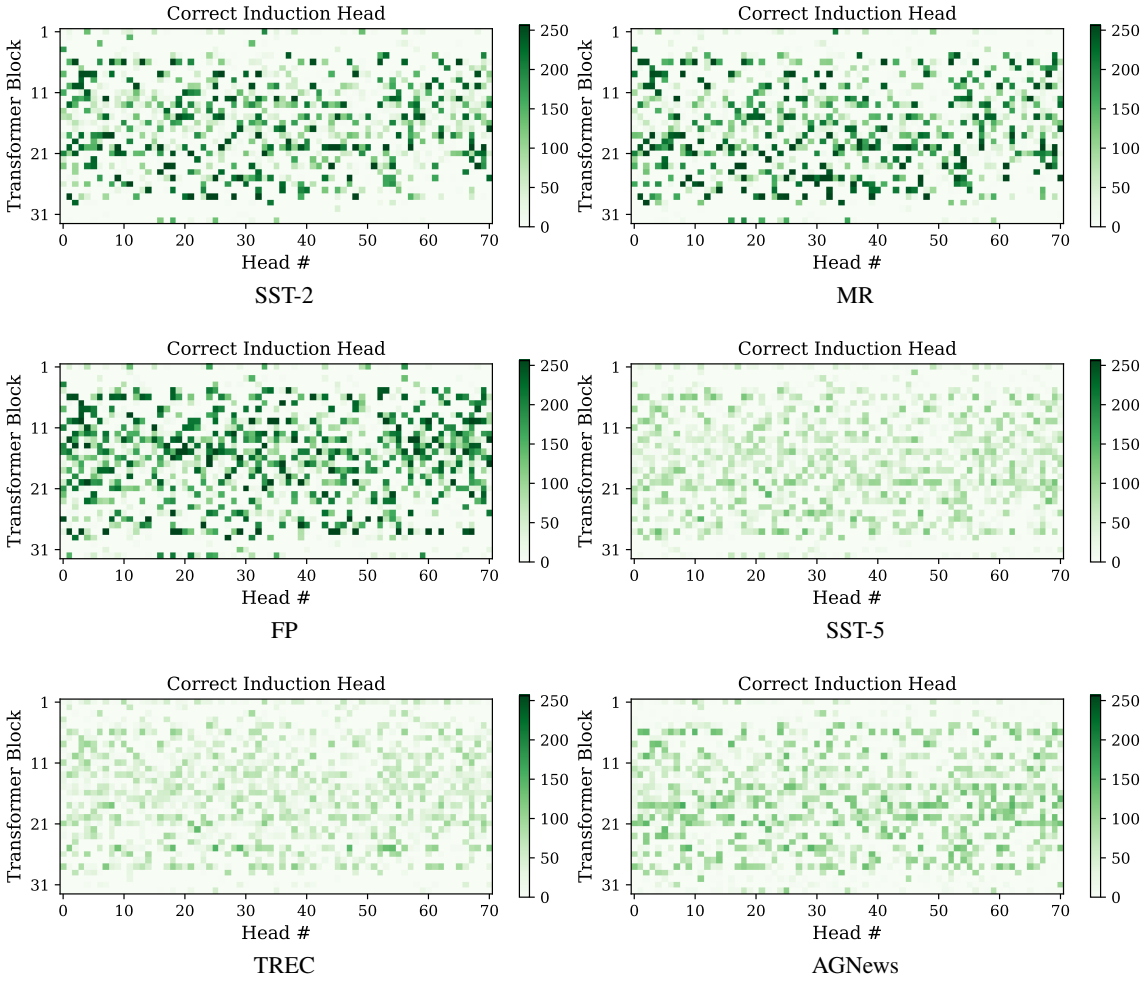


Figure A.21: Correct Induction Head marked on Falcon-7B.

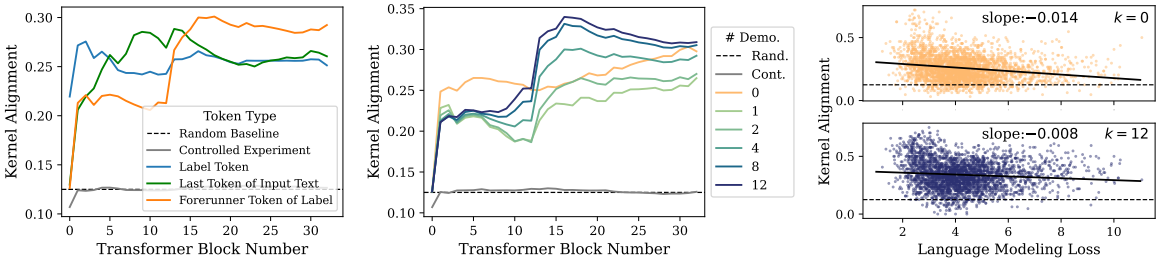


Figure A.22: Augmentated results towards Fig. 3.3 on Llama 3-8B.

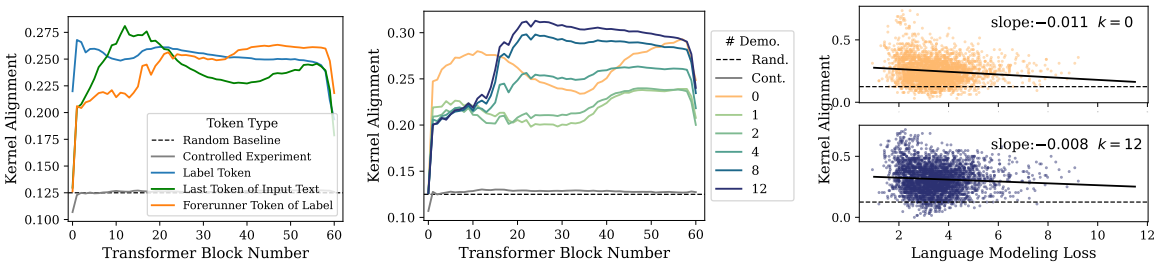


Figure A.23: Augmentated results towards Fig. 3.3 on Falcon-40B.

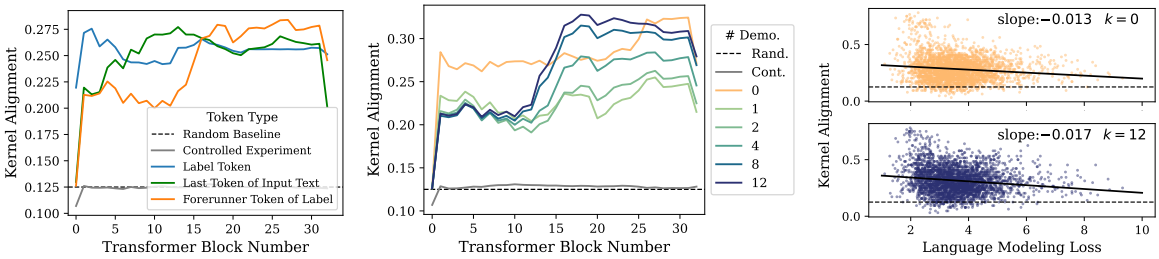


Figure A.24: Augmentated results towards Fig. 3.3 on Falcon-7B.

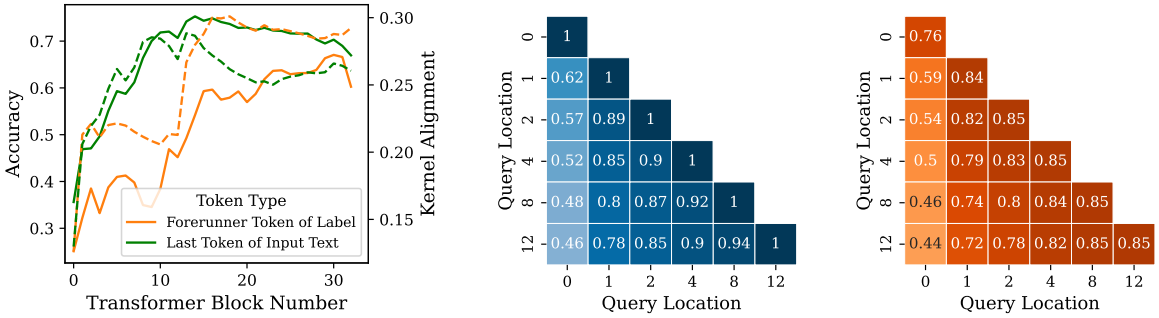


Figure A.25: Augmentated results towards Fig. 3.2 and 3.4 (on Layer 16) on Llama 3-8B.

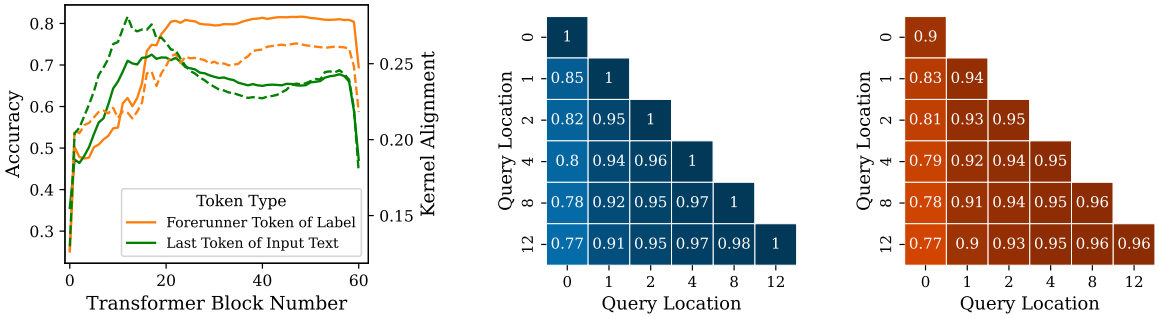


Figure A.26: Augmentated results towards Fig. 3.2 and 3.4 (on Layer 24) on Falcon-40B.

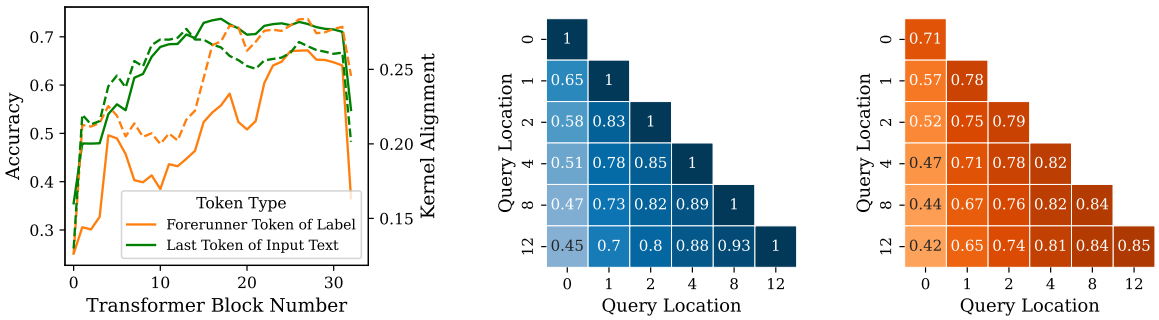


Figure A.27: Augmentated results towards Fig. 3.2 and 3.4 (on Layer 16) on Falcon-7B.

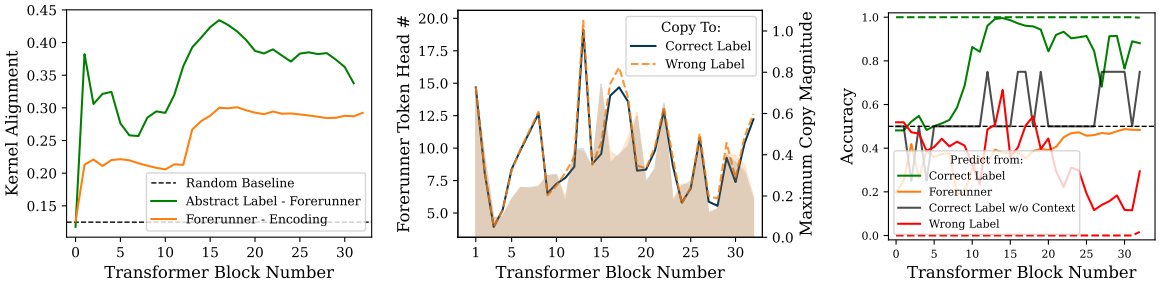


Figure A.28: Augmentated results towards Fig. 3.5 on Llama 3-8B.

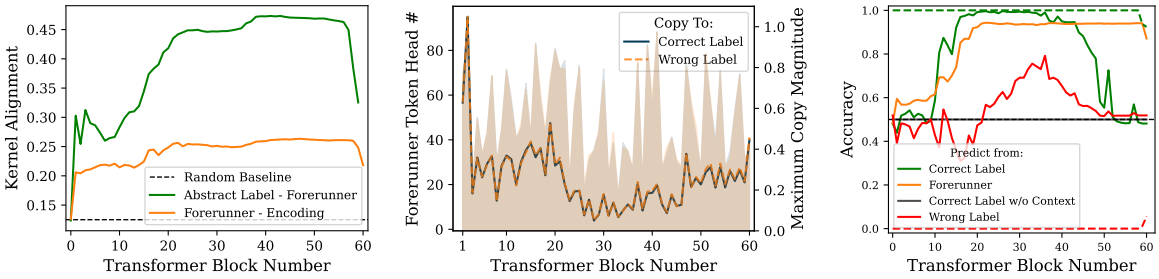


Figure A.29: Augmentated results towards Fig. 3.5 on Falcon-40B.

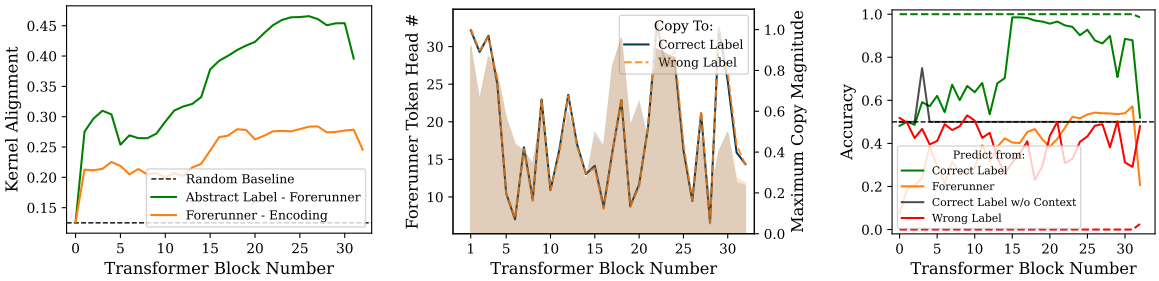


Figure A.30: Augmentated results towards Fig. 3.5 on Falcon-7B.

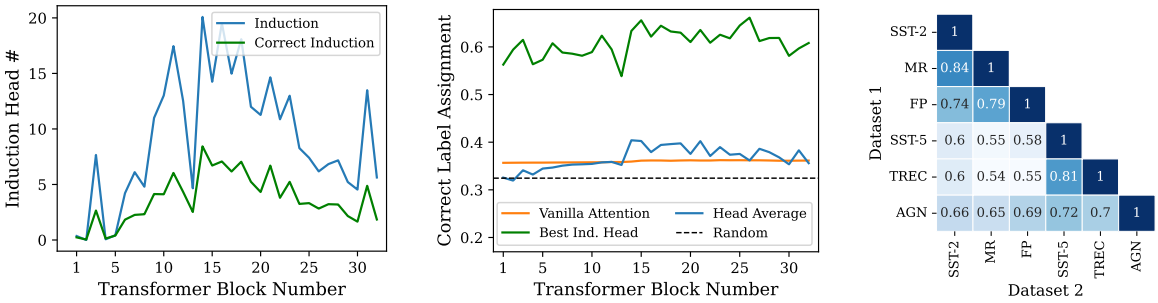


Figure A.31: Augmentated results towards Fig. 3.6 on Llama 3-8B.

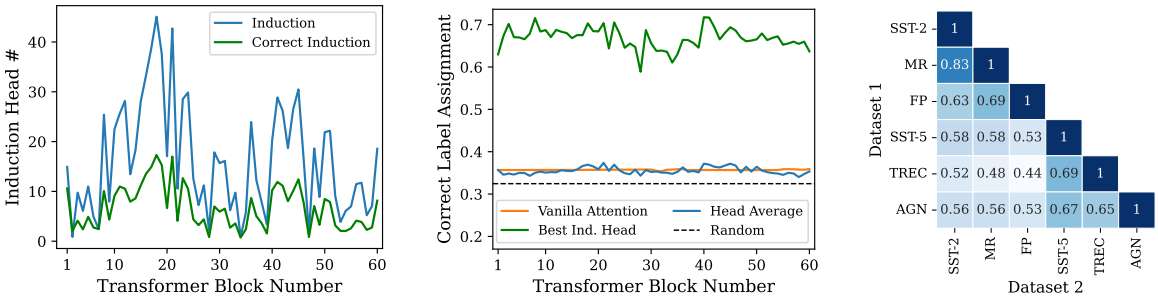


Figure A.32: Augmentated results towards Fig. 3.6 on Falcon-40B.

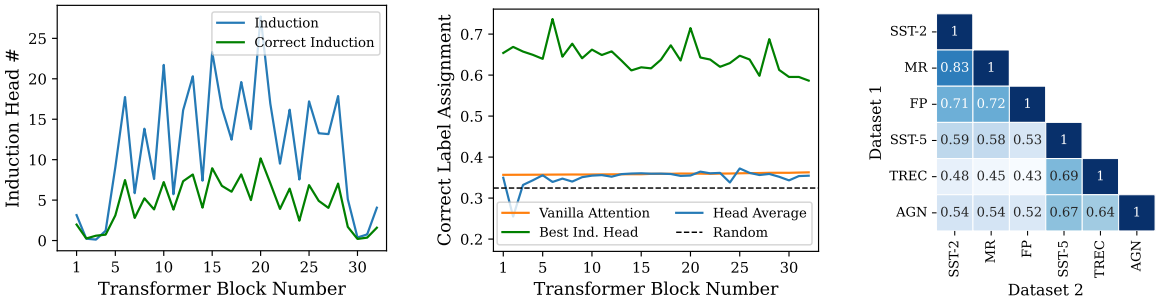


Figure A.33: Augmentated results towards Fig. 3.6 on Falcon-7B.

Appendix B

Supplementary Materials for Chapter 4

Table B.1: Prompt templates used in this work.

Dataset	Prompt Template (Unit)	Label Tokens
SST-2	`sentence: [x] sentiment: [y] \n`	negative, positive
MR	`review: [x] sentiment: [y] \n`	negative, positive
FP	`sentence: [x] sentiment: [y] \n`	negative, neutral, positive
SST-5	`sentence: [x] sentiment: [y] \n`	poor, bad, neutral, good, great
AGNews	`news: [x] topic: [y] \n`	world, sports, business, science
Subjective	`review: [x] subjectiveness: [y] \n`	objective, subjective
country-capitals	`country: [x], label: [y] \n`	-
people-profession	`name: [x], label: [y] \n`	-
opus-100	`sentence: [x], translation: [y] \n`	-

Table B.2: Instruction used in Fig. 4.5.

	Dataset	Instruction
Basic	SST-2	You are a helpful assistant. Please predict the sentiment of the following sentence:
	MR	You are a helpful assistant. Please predict the sentiment of the following sentence:
	FP	You are a helpful assistant. Please predict the sentiment of the following sentence:
	SST-5	You are a helpful assistant. Please predict the sentiment of the following sentence:
	AGNews	You are a helpful assistant. Please predict the category of this news:
	Subjective	You are a helpful assistant. Please predict the subjectivity of this sentence:
w/ Label Space	SST-2	You are a helpful assistant. Please predict the sentiment of the following sentence in positive and negative:
	MR	You are a helpful assistant. Please predict the sentiment of the following sentence in positive and negative:
	FP	You are a helpful assistant. Please predict the sentiment of the following sentence in positive, neutral, and negative:
	SST-5	You are a helpful assistant. Please predict the sentiment of the following sentence in poor, bad, neutral, good, and great:
	AGNews	You are a helpful assistant. Please predict the category of this news in world, sports, business, science:
	Subjective	You are a helpful assistant. Please predict the subjectivity of this sentence in objective and subjective:

B.1 Experiment Details

All the models and datasets are loaded from HuggingFace, with checkpoint names listed in Table B.3. The Llama 3-13B Instruct is quantized to `INT4`.

Table B.3: Models and corresponding checkpoint names used in this work.

Model	Checkpoint
Llama 3.2-1B	`meta-llama/Llama-3.2-1B`
Llama 3-8B	`meta-llama/Meta-Llama-3-8B`
Llama 3-13B Instruct	`elinas/Llama-3-13B-Instruct`
Qwen 2.5-3B	`Qwen/Qwen 2.5-3B`
Qwen 2.5-3B Instruct	`Qwen/Qwen2.5-3B-Instruct`
Qwen 2.5-7B	`Qwen/Qwen 2.5-7B`

B.1.1 Prompt Template

In this chapter, we build the input prompt from (Cho & Inoue, 2025), detailed in Table B.1. Moreover, in the experiments shown in Fig. 4.5, we utilize the instructions as shown in Table B.2, with and without label spaces provided in the instruction.

We illustrate seen vs. unseen label scenarios in Fig. B.1. In the seen case, the query's ground-truth answer appears at least once in the demonstration, activating induction heads for direct copying. In the unseen case, the answer is absent, and induction heads lose their functionality (Cho et al., 2025a).

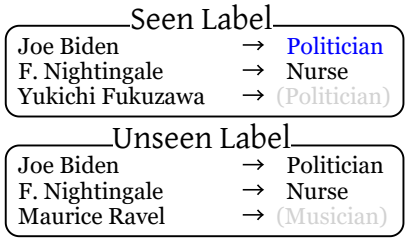


Figure B.1: Examples of seen label demonstration and unseen label demonstration.

Also, a random label prompt utilized in §4.4.3 is built by randomly sampling the labels presented in the demonstrations in the label space. For example, for the prompt shown in Fig. B.2, we randomly (with probability 50%) flip the "positive" to "negative" and "negative" to "positive".

B.1.2 Training Details of $W_{\text{enc}}W_{\text{dec}}$ (§4.4.2)

As shown in Fig. 4.1 (C), in the experiment of §4.4.2, we train two contiguous linear layers in the residual stream of the LMs, with only the first linear layer (parameterized by W_{enc}) attached with a bias term. During the training, only the two linear layers are unlocked for parameter update. We sample 2048 zero-shot training examples, and train the filter on Adam (Kingma & Ba, 2014) with learning rate 10^{-4}, momentum factor $\beta_1 = 0.9$, $\beta_2 = 0.999$. After the gradient calculation of every 32 training samples, we update the parameter once, i.e., we utilize a pseudo batch size of 32. The filters are trained for 4 epochs. After the training, we test the model on a 512-size hold-out test set by strict token matching across the entire vocabulary space.

B.1.3 Details for Eccentricity / Covariance Flux and Experiments of §4.4.3

We intuitively illustrate the eccentricity and covariance flux metrics in Fig. B.10.

The Calculation of Covariance Flux. Given the last token hidden state set $H^{l,k} = \{h_i^{l,k}\}_{i=1}^N$ from layer l and k-shot ICL prompts, we recall the low-rank filter obtained in §4.4.2 (independently trained from 0-shot setting) as W_{enc} and W_{dec}. Then, we calculate the mapped[26] hidden state set as $H_1^{l,k} = \{h_i^{l,k}W_{\text{enc}}W_{\text{dec}}\}_{i=1}^N$, and calculate the covariance matrix of the mapped set as $D_1^{l,k} = \text{Cov}[H_1^{l,k}]$, also the covariance matrix of the original set as $D^{l,k} = \text{Cov}[H^{l,k}]$. Then, we calculate the covariance flux as:

$$\text{Covariance Flux} = \frac{\|D_1^{l,k}\|_*}{\|D^{l,k}\|_*}, \qquad \text{(B.1)}$$

where the $\|\cdot\|_*$ is the nuclear norm, i.e., the sum of the singular values.

Experiment Details. Based on the same 512-size test set with Appendix B.1.2, we sample 2 demonstration sequence for each test sample with k demonstrations. Then, on the specified layer, we extract the hidden states from these 1024 test samples to measure the 2 metrics.

Accuracy with Instruction and Label Configurations. As supplementary information, we list the accuracies in the input configurations shown in Fig. 4.5 and 4.6 in Table B.4.

Table B.4: Accuracies in all configurations shown in Fig. 4.5 and 4.6. **Ins.**: with basic instruction; **Ins. (LS)**: with instructions and label space prompt; **Unseen**: with unseen label settings; **Rand** y: with randomly sampled demonstrations label tokens in the original label space.

	Dataset	0-shot	0-shot Ins.	0-shot Ins. (LS)	8-shot	8-shot Unseen	8-shot Rand y
Llama 3.2-1B	SST-2	4.49	22.27	48.83	87.40	15.14	77.44
	MR	0.00	42.19	46.68	90.23	44.92	88.77
	FP	1.17	12.30	15.82	73.14	0.39	65.04
	SST-5	0.00	0.78	17.97	42.19	2.93	37.60
	AGNews	0.20	2.34	15.23	71.78	23.83	66.21
	Subjective	0.00	0.00	0.00	61.43	0.00	50.78
	Average	0.98	13.31	24.09	71.03	14.53	64.31
Llama 3-8B	SST-2	17.38	66.60	24.61	91.31	27.15	82.71
	MR	0.00	65.04	65.23	93.07	52.83	83.89
	FP	27.34	25.98	22.66	82.71	9.57	57.03
	SST-5	0.20	0.00	23.63	45.31	6.74	40.23
	AGNews	0.98	12.11	9.77	75.59	33.69	48.73
	Subjective	0.00	0.00	0.00	77.83	0.00	53.81
	Average	7.65	28.29	24.32	77.64	21.66	61.07
Llama 3-13B	SST-2	63.48	5.66	58.79	76.86	29.59	80.08
	MR	11.13	1.56	49.80	79.79	35.64	79.39
	FP	70.70	6.25	21.29	83.11	65.43	75.10
	SST-5	4.30	0.00	42.58	47.85	7.03	44.34
	AGNews	13.67	24.22	14.65	68.75	58.79	60.16
	Subjective	0.00	0.00	0.00	79.79	0.00	53.81
	Average	27.21	6.28	31.18	72.69	32.75	65.48
Qwen 2.5-3B	SST-2	43.55	47.66	77.34	92.58	56.05	81.84
	MR	28.13	61.52	84.38	91.31	50.98	83.01
	FP	8.40	31.05	72.85	78.81	25.00	55.86
	SST-5	0.00	0.00	41.99	48.14	5.37	41.41
	AGNews	0.59	26.95	67.19	75.39	41.50	58.20
	Subjective	0.00	0.00	0.00	62.79	0.00	52.54
	Average	13.44	27.86	57.29	74.84	29.82	62.14
Qwen 2.5-7B	SST-2	48.05	20.12	84.77	93.07	54.98	75.20
	MR	69.53	0.59	4.30	92.48	65.72	83.40
	FP	0.78	14.65	71.88	68.65	26.07	41.80
	SST-5	0.00	0.20	50.59	50.20	7.42	43.07
	AGNews	0.20	0.20	35.74	75.68	46.09	44.82
	Subjective	0.00	0.00	0.00	66.41	0.00	55.47
	Average	19.76	5.96	41.21	74.41	33.38	57.29

B.1.4 Details for Attention Visualization in Fig. 4.10

In the attention visualization, we input an 8-shot ICL prompt as shown in Fig. B.2 into Llama 3.2-1B, and extract the attention scores from the specified attention head.

B.2 The Information Removing Magnitude of Low-rank Filter

Principle of Covariance Loaded as the Lower Bound of Information Removal. Given that information theory suggests the covariance of a distribution can approximate its differential entropy (i.e., information content), we can compute the upper bound of the covariance remained after the filtering as a lower bound of

[26] Notice that the bias terms in the linear layer do not affect the covariance, so we omit them here.

sentence: 's a rollicking adventure for you and all your mateys , regardless of their ages . sentiment: positive\n
sentence: playing a role of almost bergmanesque intensity ... bisset is both convincing and radiant sentiment: positive\n
sentence: robust and scary sentiment: positive\n
sentence: addictive guilty pleasure sentiment: positive\n
sentence: chance to find love in the most unlikely place sentiment: positive\n
sentence: but because of the startling intimacy sentiment: positive\n
sentence: the movie 's seams may show ... but sentiment: positive\n
sentence: if tragic) sentiment: negative\n
sentence: likable story sentiment:

Figure B.2: The input for attention visualization.

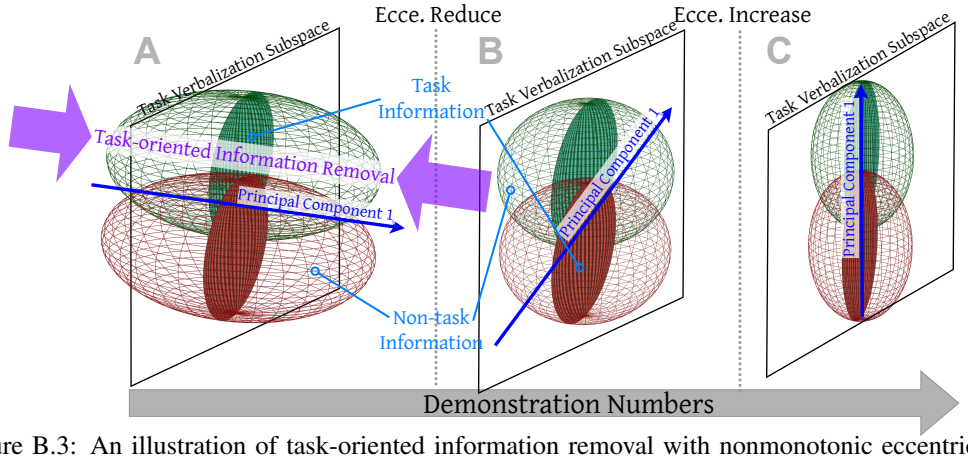


Figure B.3: An illustration of task-oriented information removal with nonmonotonic eccentricities against the number of demonstrations. The ellipsoids refer to the hidden state point cloud with two ground-truth labels (distinguished in colors), which is separated even no demonstrations are given, according to previous works (Cho et al., 2025a; Yang et al., 2025c).

the information removed by the filter. Evidently, this covariance upper bound can be taken as the variance captured by its top-r principal components.

We utilize such a principle to estimate the gap between the real information removal to the lower bound: if the eigenvectors of W_{enc} fall within the top principal component subspace of the hidden states point cloud, thus we judge that W_{enc} can better preserve the information of point clouds, and vice versa. Therefore, we map every eigenvector of W_{enc} to the orthonormal basis spanned by the top-64 principal components of the hidden state point cloud, and calculate the ratio of vector norms before and after the mapping, as shown in Fig. B.12 - B.17 for all the datasets on Llama 3.2-1B. In the results, the norm ratios remain low, suggesting that not all the covariance on the top-64 principal components of the hidden state point cloud can be preserved through the W_{enc} filter, i.e., the W_{enc} filter is an effective information removal towards the hidden states.

Although the TVS is composed of two parts W_{enc} and W_{dec}, merely confirming that W_{enc} performs significant information removal is sufficient to demonstrate that projecting the hidden state onto the $W_{\text{enc}}W_{\text{dec}}$ substantially reduces the amount of information contained in the hidden state point cloud.

B.3 Visualization: the Direction of Information Reduction

In §4.4.3, we observe that the eccentricities have a nonmonotonic pattern against the number of demonstrations on SST-2 and Llama 3.2-1B, while in some of augmentation results of Appendix B.9, e.g., results on MR and Llama 3.2-1B (Fig. B.34), such nonmonotonic pattern can not be observed.

Our explanation is that: as shown in Fig. B.3 (A), in some datasets, the distribution of the hidden states (as sentence embeddings (Cho et al., 2025a)) of zero-shot queries does not dominate on the task subspace, i.e., the

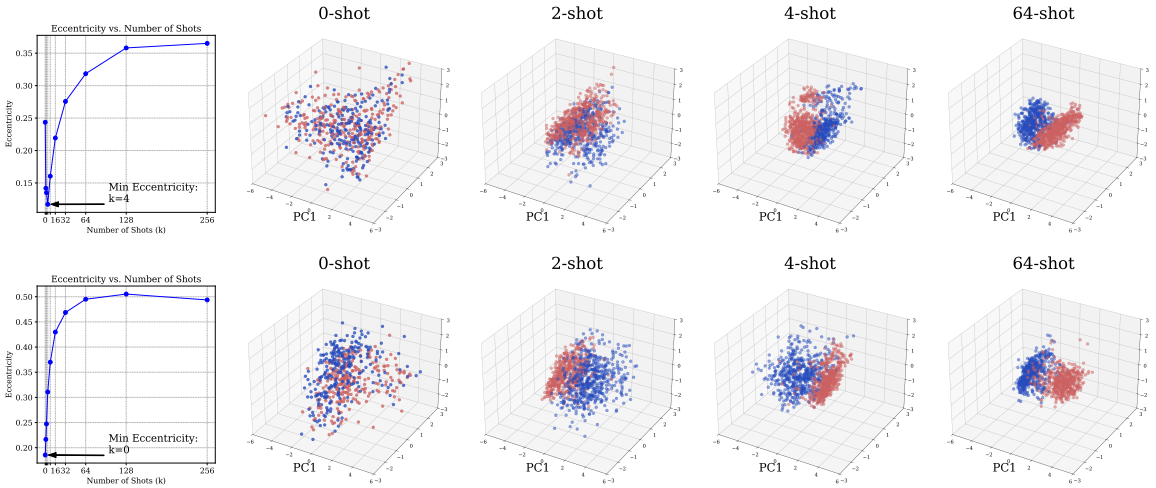


Figure B.4: Visualization of the last token's hidden states from Layer 13 of Llama 3.2-1B, (**upper**) on SST-2 with a nonmonotonic eccentricity, and (**lower**) on MR with a monotonic eccentricity. The colors refer to the queries' ground-truth label.

first principal direction of such hidden state point clouds is orthogonal to the TVS[27]. So, the early information removal when a few demonstrations are given is from a direction near the first principal direction, which reduces the covariance loaded on the first principal direction, i.e., reduces the eccentricity. Such removal gradually reduces the dominance of task-irrelevant information, causing the first principal component of the hidden state point cloud to progressively approach the TVS, as shown in Fig. B.3 (B). At this point, the eccentricity reaches its minimum and then begins to increase (Fig. B.3 (C)) gradually. While, in other datasets, the distribution of the hidden states of zero-shot queries naturally dominates on or near the TVS, that is, the information removal begins from Fig. B.3 (B), causing a monotonic increase in the eccentricity.

We prove our aforementioned inference by visualizing the last token hidden state point clouds from Layer 13 of Llama 3.2-1B on SST-2 (nonmonotonic) and MR (monotonic) with various demonstration numbers, as shown in Fig. B.4. In this visualization, on SST-2, the separability occurs on the 3rd principal component at the beginning of demonstrating, and turns to the 1st principal component on $k = 4$, where the eccentricity reaches the minimum. However, on MR, the separability occurs on the 1st principal component at the beginning, causing a monotonic eccentricity. Such observations corroborate the process illustrated in Fig. B.4 and confirm an implicit yet important property: task-oriented information removal preserves the distribution along the separable directions of the hidden state point cloud, while compressing the distribution along the non-separable directions.

Moreover, we consider the reason for such a distribution difference on SST-2 and MR as the difference in the inputs. We observe some inputs in both datasets, as shown in Fig. B.6.

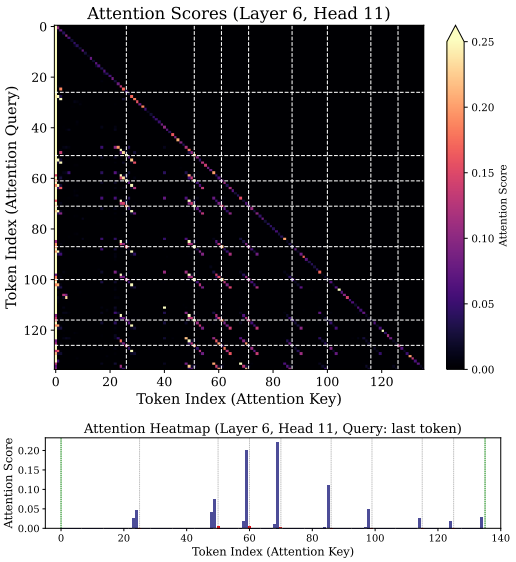


Figure B.5: (Upper) attention map; (lower) attention scores to the last token (":", as the attention query) of Layer 6 Head 11.

MR examples show clear sentiment tendencies, absent in SST-2, likely explaining the hidden state distribution difference between the two datasets. Moreover, such a distributional property, i.e., signal-noise-ratio describing whether the zero-shot hidden states are in the status of Fig. B.3 (A) or (C), can be designed into a metric to characterize the task difficulty. We leave such a discussion to future work.

[27] In this visualization and the experiments of this section, we implicitly make the assumption that TVS maximizes, or at least preserves, the separability of clusters produced by queries of different labels. This assumption is acceptable: if TVS were to confound this separability, the mapping results on TVS would fail to effectively distinguish different labels, which would contradict the high accuracy observed in Fig. 4.3.

(Test samples from SST-2)
- video, and.
- stirs us as well.
- all of Dean's mannerisms and self-indulgence,

(Test samples from MR.)
- Audiard successfully maintains suspense on different levels throughout a film that is both gripping and compelling.
- The problem with the mayhem in Formula 51 is not that it's offensive, but that it's boring.
- Doesn't deliver a great story, nor is the action as gripping as in past Seagal films.

Figure B.6: Examples for test samples from (upper) SST-2 and (lower) MR.

B.4 Case Analysis for DHs' Mechanism

In this section, we provide prototypical evidence for the mechanism of DH, i.e., locating the important query tokens with task-related information by the attention scores, where the hidden states processed by the $W_Q^\top W_K$ serve as a detector of such tokens. In detail, we visualize the attention scores of Llama 3.2-1B Layer 9 Head 15 on the last tokens in some cases from SST-2, as shown in Fig. B.8. In the visualization, it is obvious that the attention scores are concentrated on some sentiment-related tokens, which contain the task-related information for the sentiment analysis task defined on SST-2. This result confirms that DHs can leverage attention scores to correctly filter hidden states generated by tokens containing task-relevant information, thereby amplifying task-related representations, so as to relatively reduce task-irrelevant information. In this process, we can equivalently interpret $W_Q^\top W_K$ as extracting information from a certain subspace of the last-token hidden state, which encodes the criterion for determining task information, enabling the subsequent dot-product operations with each hidden state vector to select task-relevant information.

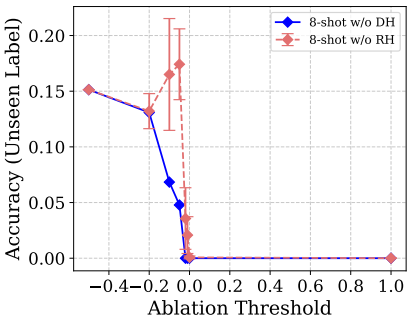


Figure B.7: DH / RH ablation results against ablation threshold.

B.5 Discussion on the Selection of Ablation Threshold

The ablation experiments in §4.4.4 utilize a fixed threshold of -5%, which might be suspected to be arbitrary. Therefore, this section aims to demonstrate that the ablation results are insensitive to this threshold, showing that any *mild* threshold is sufficient to distinguish DH ablation from random ablation. In detail, we vary this threshold and retest the results of ablating DHs and random heads with the same amount of DH. The results on Llama 3.2-1B SST-2 unseen setting are shown in Fig. B.7, where thresholds between -0.2 and 0 (note that we use the relative change rate of covariance flux as the metric, so this is a fairly broad range) all successfully identify the causal effect of DHs against RHs. While, if the threshold is too strict (extremely small values), no head will be ablated, and an overly loose threshold will cause many or even all heads to be ablated. Therefore, we select -0.05 within this range as the experimental setting.

B.6 "Implicit Steering Heads" Discovered in §4.4.4

Since we test the ablation effect on the accuracy of all the attention heads in §4.4.4, we can detect more important heads rather than induction heads or denoising heads as an interesting supplement. As shown in Fig. B.46 - B.51, ablating Head 11 in Layer 6 causes a very significant drop in accuracy; therefore, we examine its attention map under the aforementioned setting shown in Appendix B.1.4, as shown in Fig. B.5. We find that the attention map exhibits a localized diagonal pattern that recurs periodically with the location of label tokens. Ignoring the attention sink (Gu et al., 2025), this pattern reflects the behavior of the head: locating tokens in the preceding context that are identical to the current token (more evident in Fig. B.5

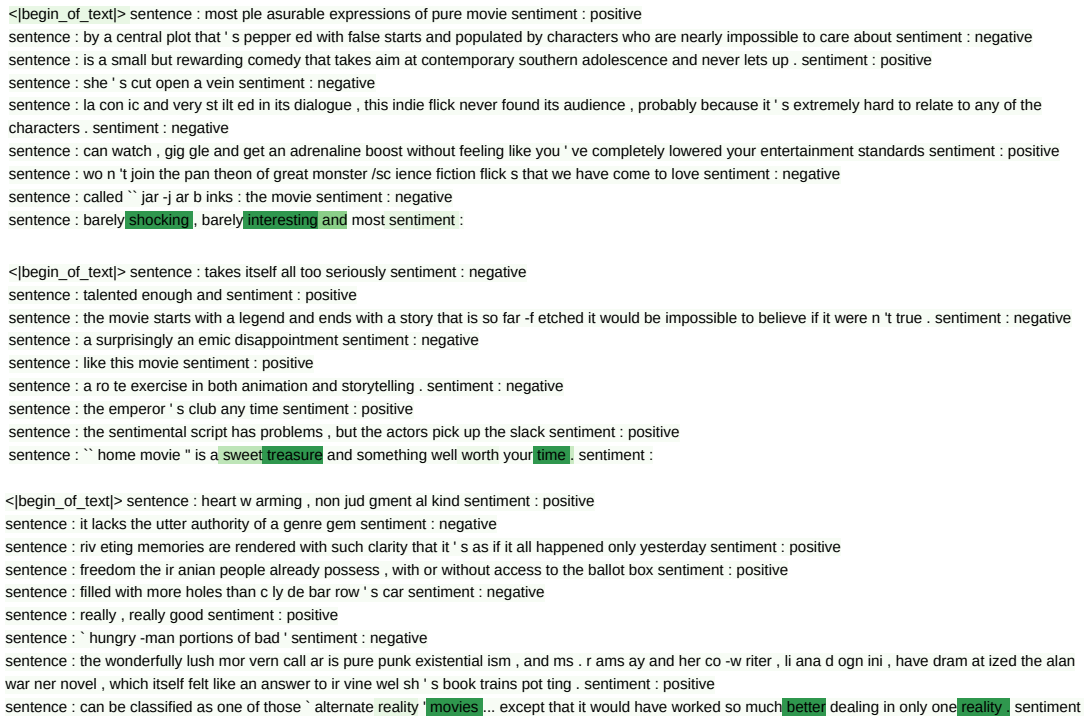


Figure B.8: Attention score visualization of DH (Llama 3.2-1B Layer 9 Head 15), where the last token serves the attention query.

(lower), where nonzero attention scores appear almost exclusively at the ":" positions, identical to the current token). Consequently, the attention map produces a periodic diagonal pattern around each label token, since the tokens at these positions are exactly the same.

Intuitively, at the final token (":"), this attention head copies information from all preceding ":" hidden states to the current position, where, each ":" stores the inference result of the previous shot round (e.g., in an 8-shot input, due to the causal mask, the 8th ":" position is effectively the output position of all preceding tokens, i.e., a 7-shot ICL input, and thus stores the previous round's inference result). In other words, this attention head functions similarly to "task vector steering": it implicitly steers the hidden state of the previous ":" (commonly used as a task vector in prior work (Hendel et al., 2023)) to the current ":" and aggregates it into an updated task vector. This may partly explain why more demonstrations can yield better accuracy, requiring further discussion.

B.7 Grounding the Metrics to Information Measurement

Since we utilize covariance-based metrics to measure the information-based quantity, which is usually measured by entropy, here we prove that the covariance is a linear-scale surrogate of entropy.

Notice that we need to handle the scenario with low-rank inputs (e.g., the point set mapped by the low-rank filter $W_{\text{enc}}W_{\text{dec}}$), we define entropy on Hausdorff measure, instead of the typical differential entropy defined on Lebesgue measure, to avoid the entropy calculated being $-\infty$.

Definition 1 (Hausdorff differential entropy on d-dimensional r-rank subspace)**.** *Let $x \in \mathbb{R}^d$ be a random vector sampled from Gaussian $X \sim \mathcal{N}(\mu, \Sigma)$, where the rank of Σ is $r \leqslant d$. Let $\mathcal{H}^r$ denote the r-dimensional Hausdorff measure, embedded on the subspace $\mathcal{M} = \mu + \mathrm{im}(\Sigma)$. If p is the density of X on $\mathcal{H}^r$, the **Hausdorff differential entropy** of X is defined as*

$$h_{\mathcal{H}^r}(X) = -\int_{\mathcal{M}} p(x) \log p(x) \, \mathrm{d}\mathcal{H}^r(x). \tag{B.2}$$

On such an entropy measurement, we can build the link to the covariance.

Theorem 1 (Non-zero eigenvalues as Hausdorff entropy estimator)**.** *Let $x \in \mathbb{R}^d$ be a random vector sampled from Gaussian $X \sim \mathcal{N}(\mu, \Sigma)$, where the rank of Σ is $r \leqslant d$. Let $\lambda_1, \ldots, \lambda_r > 0$ as the non-zero eigenvalues*

Table B.5: Fine-grained data for Table 4.2.

		Random Sample			Unseen Labels			Seen Labels		
		8-shot	**8-shot** w/o DH	**8-shot** w/o RH	**8-shot**	**8-shot** w/o DH	**8-shot** w/o RH	**8-shot**	**8-shot** w/o DH	**8-shot** w/o RH
Llama 3.2-1B	SST-2	87.40	61.42	$84.04_{1.61}$	15.14	4.78	$17.42_{3.18}$	87.79	62.10	$68.73_{1.84}$
	MR	90.23	81.25	$88.60_{4.67}$	44.92	1.27	$28.01_{11.81}$	90.53	81.64	$65.54_{2.10}$
	FP	73.11	67.38	$73.26_{3.69}$	0.39	0.00	$0.68_{0.71}$	75.00	67.97	$59.42_{4.41}$
	SST-5	42.19	30.08	$39.28_{2.08}$	2.93	0.29	$3.46_{1.34}$	48.63	33.98	$61.79_{7.77}$
	AGNews	71.78	36.72	$63.87_{9.17}$	23.83	0.00	$14.32_{10.83}$	76.95	39.94	$77.32_{2.25}$
	Subjective	61.42	53.42	$54.71_{1.81}$	0.00	0.00	$0.00_{0.00}$	61.52	53.51	$81.30_{3.93}$
	Average	71.02	55.05	$67.29_{3.84}$	14.54	1.06	$10.65_{4.65}$	73.40	56.52	$69.02_{3.72}$
Llama 3-8B	SST-2	91.31	86.23	$90.94_{0.76}$	27.14	2.83	$26.46_{4.39}$	91.89	86.91	$91.50_{0.45}$
	MR	93.07	91.31	$92.80_{0.48}$	52.83	33.69	$40.67_{14.61}$	93.46	91.60	$93.12_{0.54}$
	FP	82.71	79.59	$82.35_{0.55}$	9.57	4.69	$8.84_{0.76}$	84.96	82.13	$85.47_{0.46}$
	SST-5	45.31	43.16	$45.73_{0.48}$	6.73	3.91	$6.01_{1.29}$	54.39	51.95	$54.83_{0.42}$
	AGNews	75.58	60.74	$74.95_{0.96}$	33.69	0.00	$31.67_{2.83}$	80.27	66.70	$78.30_{1.32}$
	Subjective	77.83	71.29	$71.56_{2.54}$	0.00	0.00	$0.00_{0.00}$	77.83	71.39	$74.97_{4.61}$
	Average	77.64	72.05	$76.38_{0.96}$	21.66	7.52	$18.94_{3.98}$	80.47	75.11	$79.70_{1.30}$
Qwen 2.5-3B	SST-2	91.80	70.60	$92.30_{0.49}$	21.88	0.68	$43.49_{6.79}$	92.09	71.09	$92.65_{0.54}$
	MR	89.06	81.05	$89.94_{1.14}$	46.00	0.88	$48.24_{10.51}$	91.11	81.44	$90.33_{1.43}$
	FP	77.73	79.00	$79.45_{1.01}$	23.92	27.25	$25.31_{1.19}$	79.49	80.37	$80.35_{1.07}$
	SST-5	49.71	40.53	$47.13_{1.81}$	5.47	1.07	$5.43_{2.19}$	55.76	46.09	$53.44_{2.04}$
	AGNews	74.80	74.90	$74.22_{1.63}$	43.65	38.96	$38.46_{2.21}$	78.02	78.52	$77.42_{1.80}$
	Subjective	60.74	64.36	$62.81_{1.68}$	0.00	0.68	$0.00_{0.00}$	62.01	64.36	$63.94_{2.52}$
	Average	73.97	68.41	$74.31_{1.30}$	23.49	11.59	$26.82_{3.82}$	76.41	70.31	$76.36_{1.57}$
Qwen 2.5-3B Ins	SST-2	90.33	89.94	$90.78_{0.61}$	69.04	38.28	$66.95_{7.40}$	90.43	90.04	$89.96_{0.65}$
	MR	89.84	85.53	$88.65_{1.08}$	64.84	29.49	$66.54_{6.87}$	89.94	86.13	$89.26_{0.52}$
	FP	87.89	85.94	$88.61_{0.39}$	77.73	73.82	$73.65_{3.01}$	88.57	87.89	$88.24_{0.86}$
	SST-5	50.78	48.54	$49.96_{0.55}$	16.50	10.06	$13.97_{2.32}$	56.44	54.59	$56.33_{0.34}$
	AGNews	75.29	75.29	$74.31_{1.20}$	48.92	43.26	$43.65_{4.06}$	78.32	78.22	$78.09_{1.28}$
	Subjective	69.33	68.16	$71.35_{7.54}$	15.14	30.17	$13.18_{3.43}$	69.43	68.26	$66.44_{1.16}$
	Average	77.24	75.57	$77.28_{1.90}$	48.70	37.51	$46.32_{4.51}$	78.86	77.52	$78.05_{0.80}$

of Σ. *Then the Hausdorff differential entropy* $h_{\mathcal{H}^r}(X)$ *is given by*

$$h_{\mathcal{H}^r}(X) = \frac{r}{2}\left(\log 2\pi + 1\right) + \frac{1}{2}\sum_{i=1}^{r} \log \lambda_i \tag{B.3}$$

The Theorem 1 shows that: the covariance loaded on each principal direction (λ_i) can be a positive correlated measurement of the information (entropy) contribution ($\log \lambda_i$) on such direction, which is the theoretical grounding of our metrics design.

Proof. Since Σ has rank r, we write its eigen-decomposition as

$$\Sigma = Q \begin{pmatrix} \Lambda_r & 0 \\ 0 & 0 \end{pmatrix} Q^\top, \qquad \Lambda_r = \operatorname{diag}(\lambda_1, \ldots, \lambda_r). \tag{B.4}$$

On the Hausdorff measure $\mathcal{H}^r$ on $\mathcal{M}$, the probability density function of X is

$$p(x) = \frac{1}{(2\pi)^{r/2} \det(\Lambda_r)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^+ (x-\mu)\right), \tag{B.5}$$

where Σ^+ is the Moore-Penrose pseudoinverse of Σ.

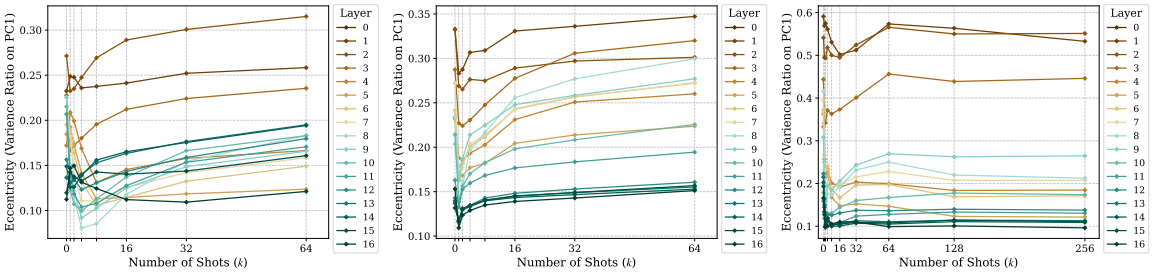


Figure B.9: Eccentricity results of 3 non-classification tasks. **(Left)** people-profession; **(Middle)** opus-100 translation: English to Chinese; **(Right)** country-capital.

The Hausdorff differential entropy is therefore

$$h_{\mathcal{H}^r}(X) = -\mathbb{E}[\log p(X)] \tag{B.6}$$

$$= \frac{r}{2}\log(2\pi) + \frac{1}{2}\log\det(\Lambda_r) + \frac{1}{2}\mathop{\mathbb{E}}_{x\sim X}\left[(x-\mu)^\top \Sigma^+ (x-\mu)\right]. \tag{B.7}$$

Since Σ^+ inverts Λ_r on the image of Σ and annihilates the orthogonal complement, we obtain

$$\mathop{\mathbb{E}}_{x\sim X}\left[(x-\mu)^\top \Sigma^+ (x-\mu)\right] = \mathrm{tr}(\Sigma^+\Sigma) = \mathrm{tr}(I_r) = r. \tag{B.8}$$

Therefore,

$$h_{\mathcal{H}^k}(X) = \frac{r}{2}\log(2\pi) + \frac{1}{2}\log\det(\Lambda_r) + \frac{r}{2} \tag{B.9}$$

$$= \frac{r}{2}(\log 2\pi + 1) + \frac{1}{2}\sum_{i=1}^{r}\log\lambda_i \tag{B.10}$$

□

B.8 Prototype of Information Removal on Generative Scenario

In §4.5, we discuss some infinite label space settings, i.e., the country-capitals task and the people-profession task, but still on one-token or few-token labels. Therefore, in this section, we generalize these discussions to the generative tasks. In summary, we observe similar information removal dynamics on generative tasks with classification tasks (Fig. 4.4 (middle)). However, as acknowledged in the Limitation (4) (§4.5), we can not obtain a linear information bottleneck to evaluate the correctness of such information removal (i.e., Covariance Flux). Therefore, this experiment serves only as a prototypical observation intended to motivate subsequent research within our framework.

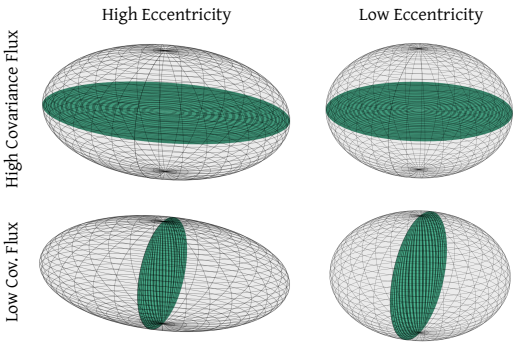


Figure B.10: Diagram for eccentricity and covariance flux metrics. The green plain refers to the TVS ($W_{\text{enc}}W_{\text{dec}}$) obtained in §4.4.2.

Specifically, we perform tests on the opus-100 (Zhang et al., 2020; Tiedemann, 2012) English–Chinese translation dataset and Llama 3.2-1B that follow the same procedure of eccentricity calculation of Step 2 described in §4.3, yielding the results shown in Fig. B.9 (middle). Note that translation is not a bijective task (i.e., a single translation may correspond to multiple possible source sentences). Therefore, the translation task exhibits a clear information-removal trend. This is intuitive: with the prompt template "sentence: [original text], translation: [translated text]", the target language is not explicitly specified in zero-shot inputs. Thus, at the last token, the model encodes verbalization patterns corresponding to all potential target languages, with the *default* pattern dominating. Once some demonstrations are provided, non-target verbalizations are removed. If the specified target language differs from the "default" pattern, this produces the decrease-and-increase pattern as observed, following the description in Appendix B.3.

Such results directly confirm the information removal dynamics in task recognition driven by ICL demonstrations for generative and non-bijective tasks. However, it is conceivable that the outputs of such generative tasks are distributed on a manifold (to be compared, the outputs of classification tasks are distributed in the space spanned by the unembedding vectors of their labels). Therefore, utilizing a simple linear filter to actively remove task-irrelevant information and locate denoising heads is difficult, as shown in Fig. B.11, which echoes our Limitation 4. Future work can intuitively utilize more complex information bottlenecks to extend our framework to the nonlinear tasks.

Moreover, we repeat the experiment on the people-profession task and country-capital shown in Fig. B.9 (left) and (right). In these results, we can observe a clear information removal dynamic in the people–profession task, while the country–capital task shows no visible interval of increasing eccentricity, which is consistent with our earlier discussion in §4.5.

B.9 Supplementary Materials Outline

Augmentation Results for Fig. 4.3 ("Filter Injection"). We repeat the experiment shown in Fig. 4.3 on all the mentioned settings, as shown in Fig. B.18 - B.23, which are consistent with Fig. 4.3.

Augmentation Results for the Scatter Size of Fig. 4.3 ("Filter Injection: Covariance out of Rank r"). We explicitly visualize the numerical results of the lower-bound of information removal (i.e., the covariance rate out of rank) shown as the scatter size in Fig. 4.3, as shown in Fig. B.29 - B.33[28].

Augmentation Results for Table 4.1 ("Symbolic Fine-tuning"). We repeat the experiment shown in Table 4.1 on Llama 3-8B and Qwen 2.5-3B, as shown in Fig. B.26, which are consistent with Table 4.1.

Augmentation Results for Fig. 4.4 ("Eccentricity and Covariance Flux against k"). We repeat the experiment shown in Fig. 4.4 on all the mentioned settings, as shown in Fig. B.34 - B.36. The results are consistent with Fig. 4.4.

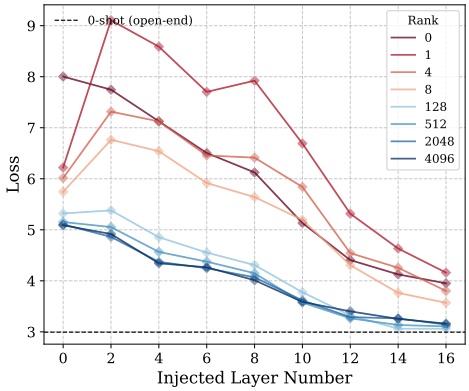


Figure B.11: Filter-injection evaluations on opus-100 English–Chinese translation task. **In cross-entropy loss.**

Augmentation Results for Fig. 4.5 and 4.6 ("Eccentricity and Covariance with Instruction and Various Labels"). We repeat the experiment shown in Fig. 4.5 and 4.6 on all the mentioned settings, as shown in Fig. B.40 - B.45. The results are consistent with Fig. 4.5 and 4.6.

Accuracy with Instruction and Label Configuration in Fig. 4.5 and 4.6. We report the test accuracies on the input configurations in Fig. 4.5 and 4.6 in Table B.4.

Augmentation Results for Fig. 4.7 ("Finding Denoising Heads"). We repeat the experiment shown in Fig. 4.7 on all the datasets with 8-shot inputs and Llama 3.2-1B, Llama 3-8B, Qwen 2.5-3B, Qwen 2.5-3B Instruct in Fig. B.46 - B.69.

Augmentation Results for Fig. 4.8 ("Layer Distribution of Denoising Heads"). We repeat the visualization shown in Fig. 4.8 on all the datasets and Llama 3.2-1B, Llama 3-8B, Qwen 2.5-3B, Qwen 2.5-3B Instruct in Fig. B.70 - B.73.

Augmentation Results for Fig. 4.11 ("Denoising Head Overlap among Datasets"). We repeat the visualization shown in Fig. 4.11 on Llama 3-8B and Qwen 2.5-3B in Fig. B.27.

Augmentation Results for Fig. 4.9 ("Overlap of Induction Heads and Denoising Heads"). We repeat the visualization shown in Fig. 4.9 on all the datasets and Llama 3.2-1B, Llama 3-8B, Qwen 2.5-3B, Qwen 2.5-3B Instruct in Fig. B.74 - B.77.

[28]The curve with $r = 8$ is usually plotted at a higher resolution, i.e., one point is drawn for each layer (if $r \neq 8$, we sometimes draw one point for each 2 layers). Therefore, in some cases, the $r = 8$ curve appears above curves with lower ranks since the scree plot is usually convex.

Augmentation Results for Fig. 4.10 ("Attention Map Visualization"). We repeat the visualization shown in Fig. 4.10 on more 2 denoising heads as shown in Fig. B.28.

Augmentation Results for Table 4.2 ("Ablation Experiment"). We list the ablation results on individual datasets rather than the average ones in Table B.5.

Augmentation Results for Fig. 4.12 and 4.13 ("Factor-recall Filter Injection"). We repeat the experiment shown in Fig. 4.12 and 4.13 on Llama 3-8B, Qwen 2.5-3B and 7B as shown in Fig. B.24 and B.25.

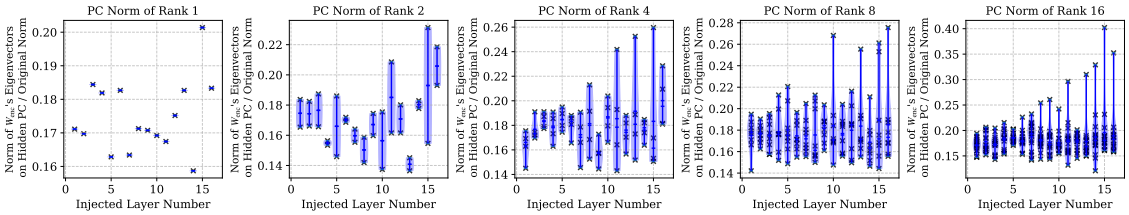


Figure B.12: The cosine similarity between all the eigenvectors (each for a scatter) of W_{enc} on specific layers and the subspace spanned by the top-64 principal components of the hidden state point cloud, measured by the ratio of vector norms before and after mapping to the principal subspace. On Llama 3.2-1B, SST-2, repeated on various inner ranks of TVS, i.e., the column dimensionality of W_{enc}.

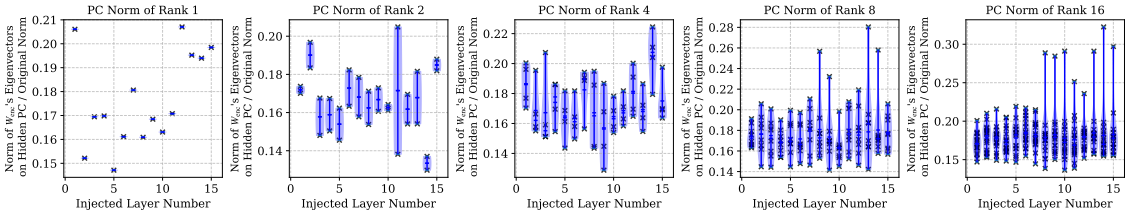


Figure B.13: The similar result with Fig. B.12 on Llama 3.2-1B, MR.

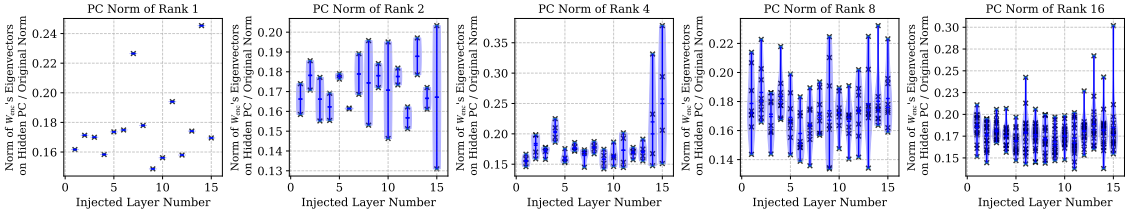


Figure B.14: The similar result with Fig. B.12 on Llama 3.2-1B, FP.

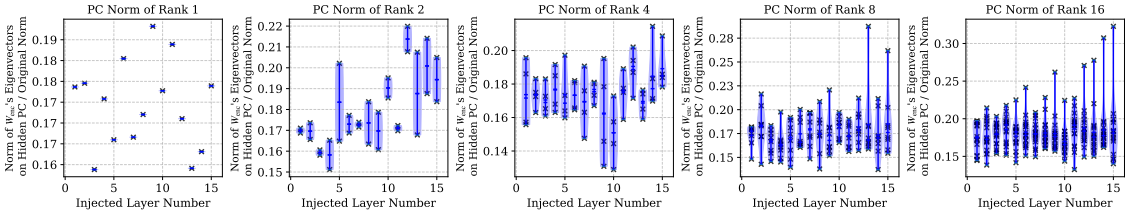


Figure B.15: The similar result with Fig. B.12 on Llama 3.2-1B, SST-5.

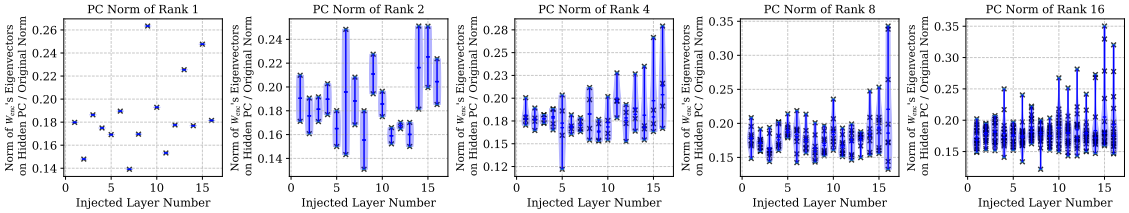


Figure B.16: The similar result with Fig. B.12 on Llama 3.2-1B, AGNews.

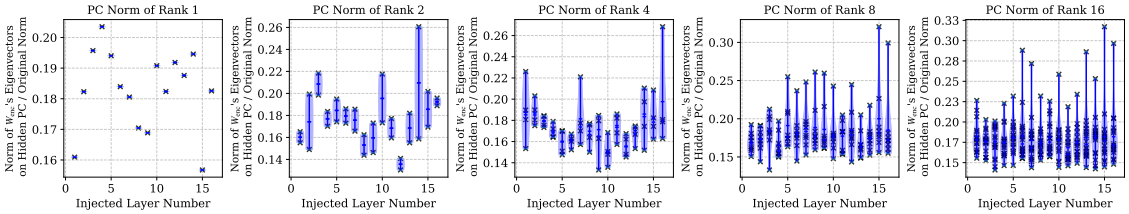


Figure B.17: The similar result with Fig. B.12 on Llama 3.2-1B, Subjective.

Supplementary Materials

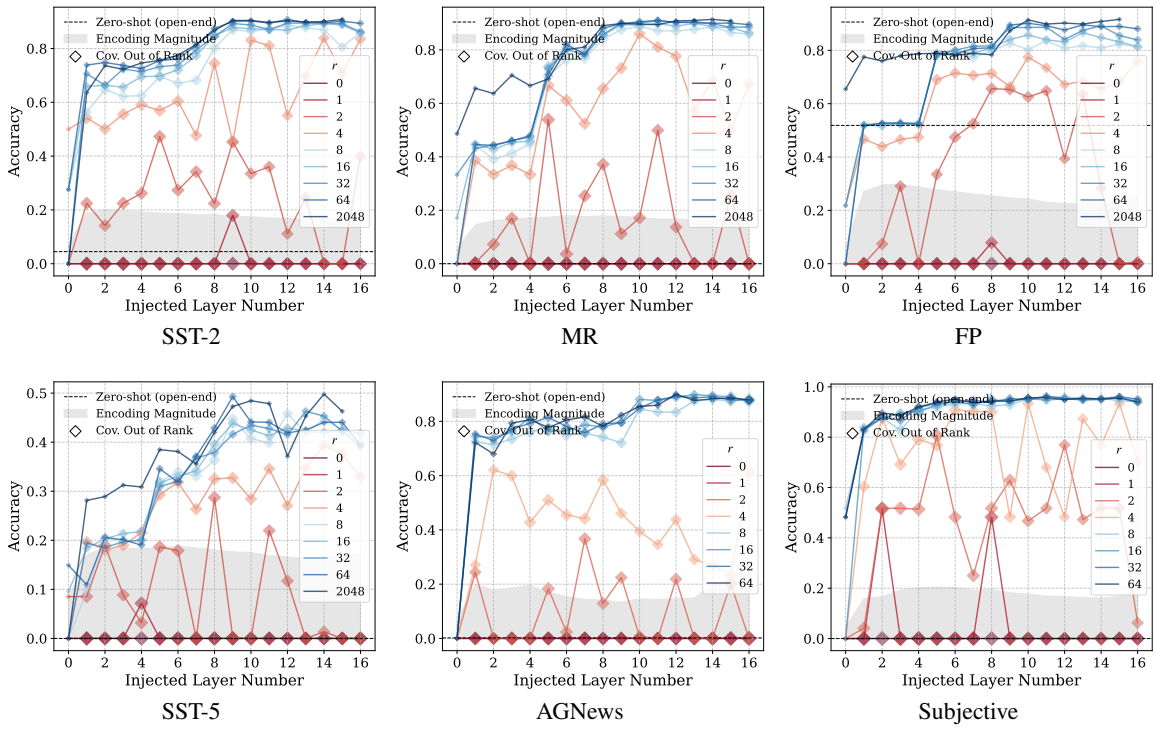


Figure B.18: Augmentation results for Fig. 4.3 on Llama 3.2-1B.

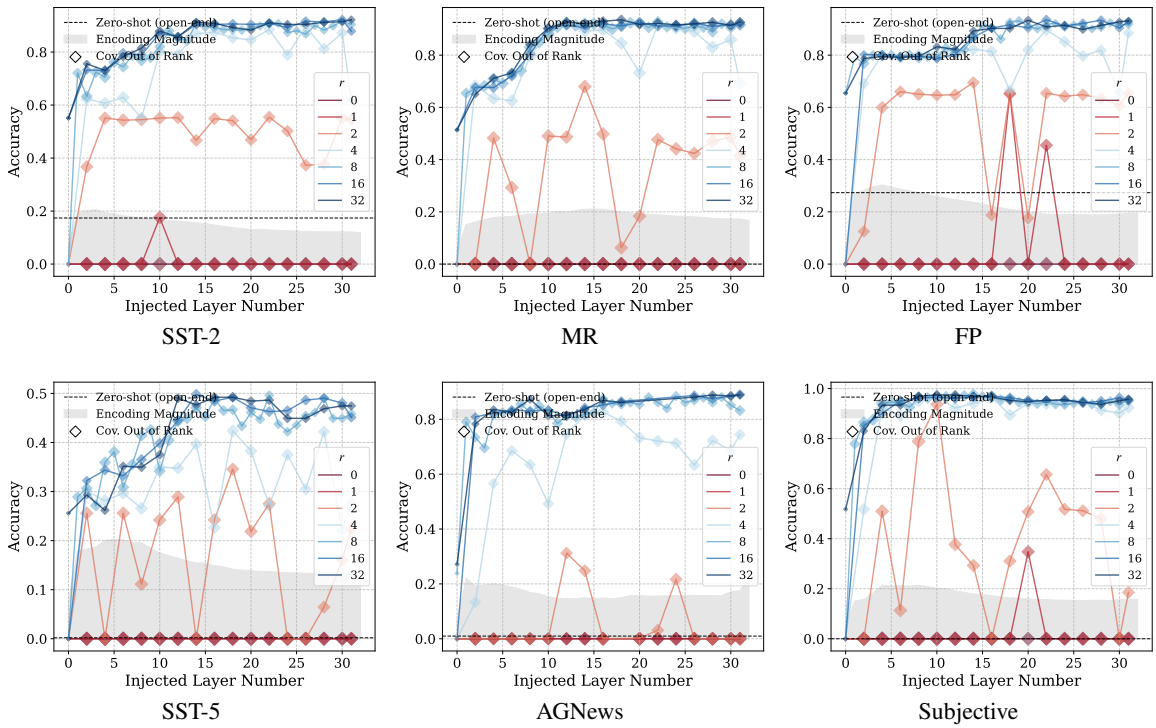


Figure B.19: Augmentation results for Fig. 4.3 on Llama 3-8B.

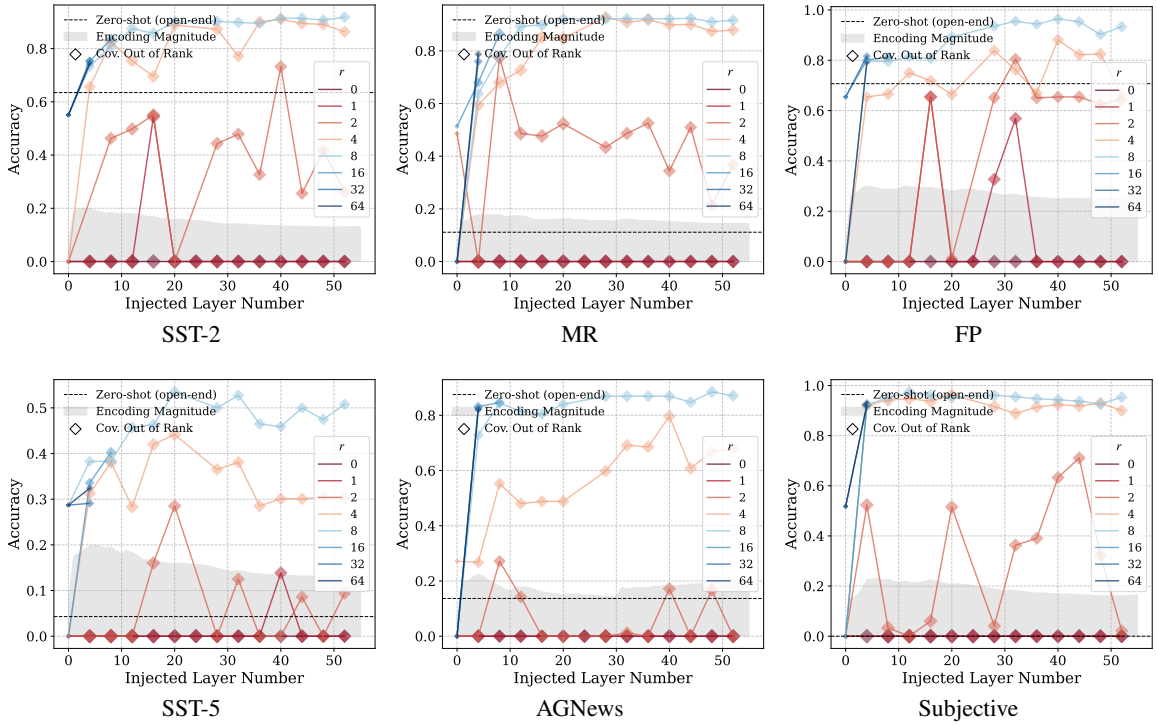


Figure B.20: Augmentation results for Fig. 4.3 on Llama 3-13B.

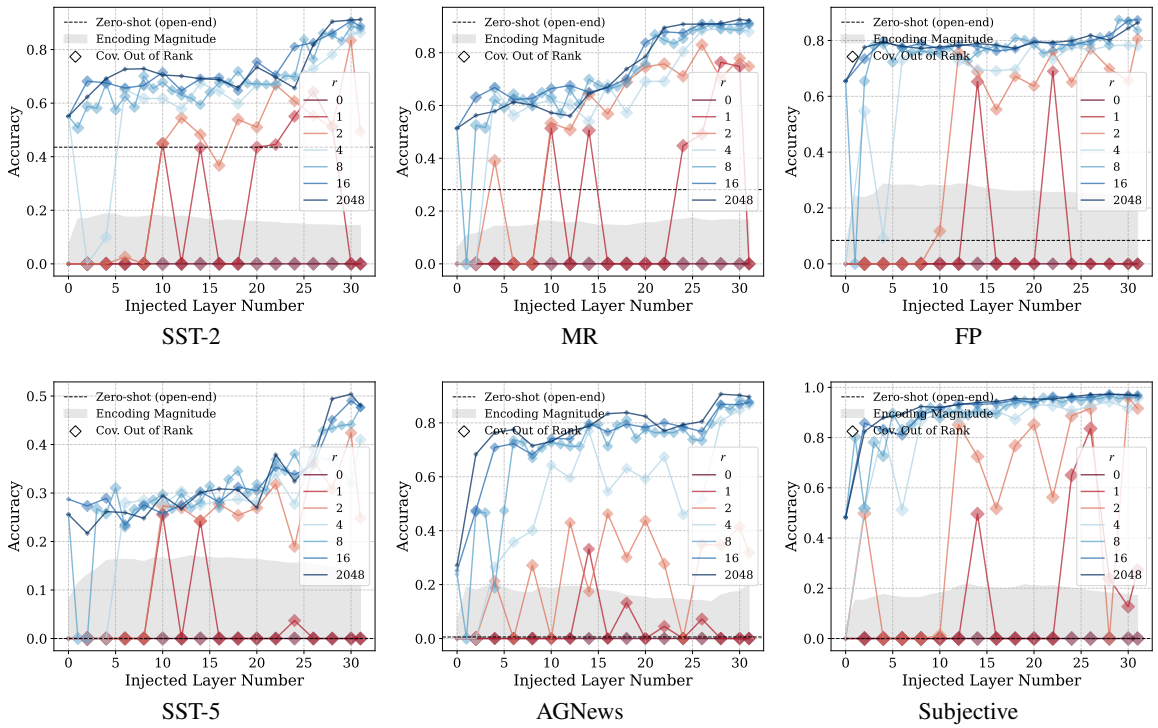


Figure B.21: Augmentation results for Fig. 4.3 on Qwen 2.5-3B.

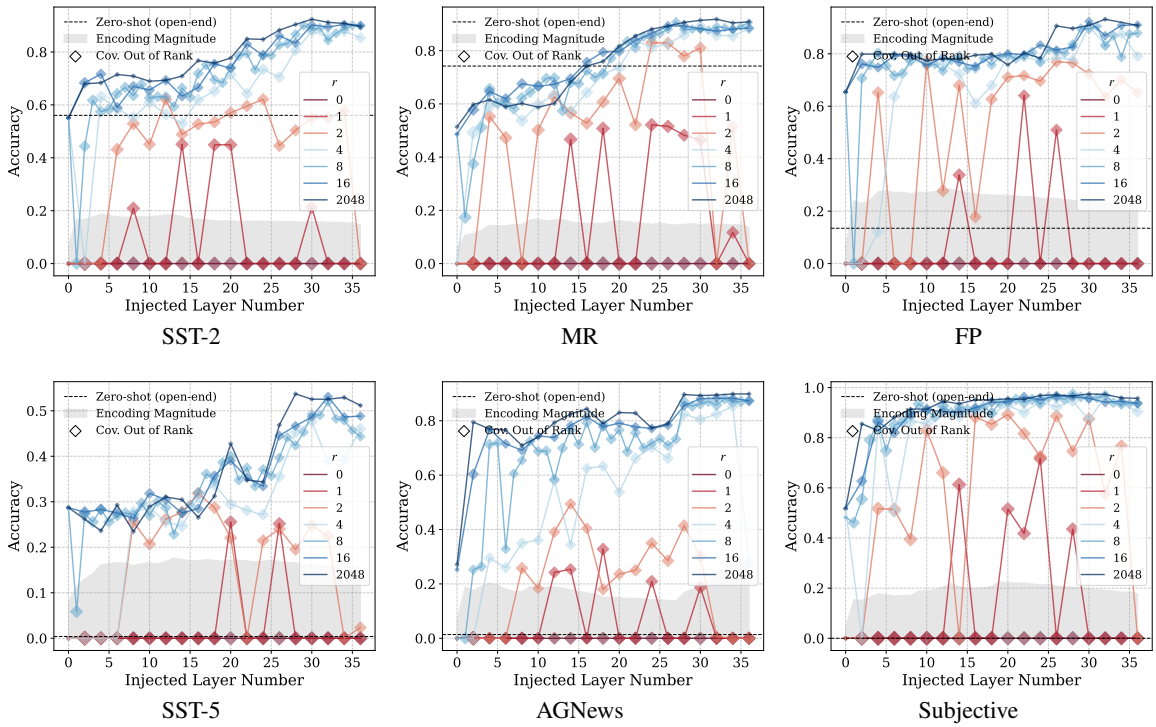


Figure B.22: Augmentation results for Fig. 4.3 on Qwen 2.5-3B Instruct.

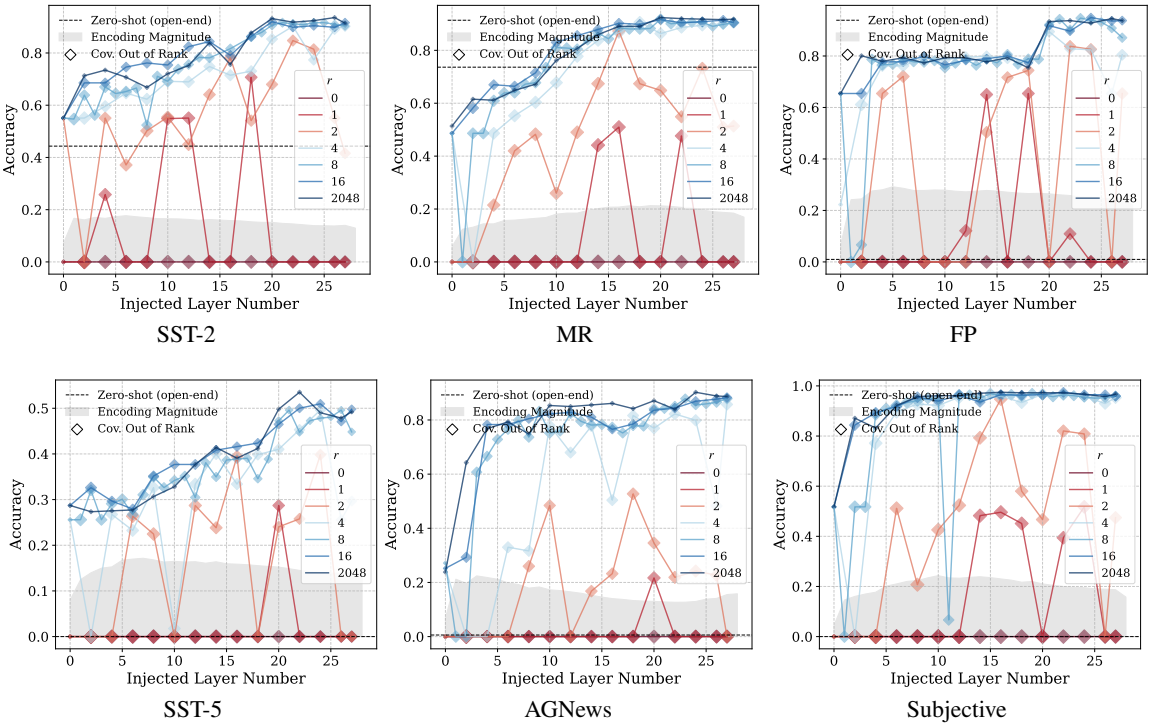
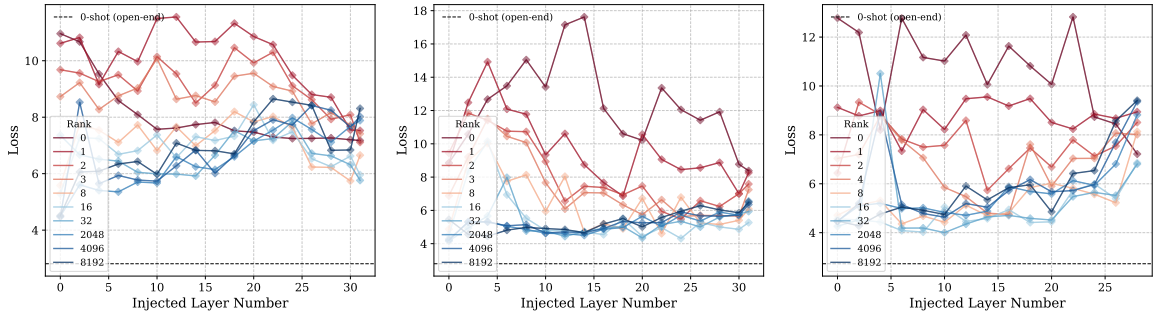
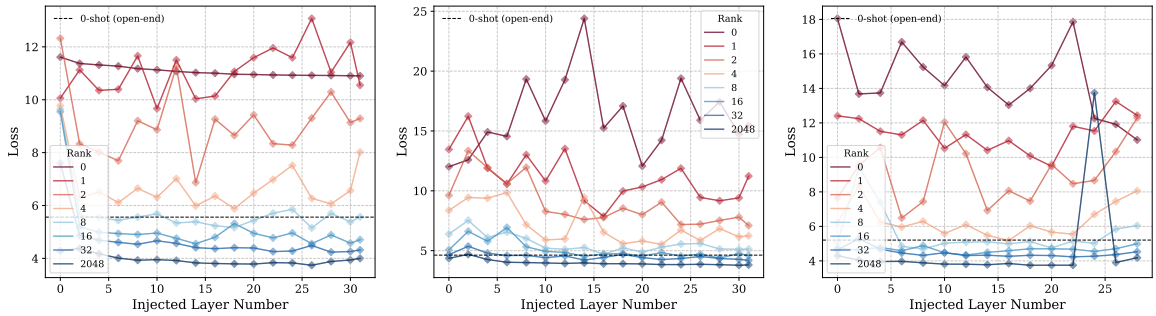


Figure B.23: Augmentation results for Fig. 4.3 on Qwen 2.5-7B.



Llama 3-8B Qwen 2.5-3B Qwen 2.5-7B

Figure B.24: Augmentation results for Fig. 4.12 on 3 models.



Llama 3-8B Qwen 2.5-3B Qwen 2.5-7B

Figure B.25: Augmentation results for Fig. 4.13 on 3 models.

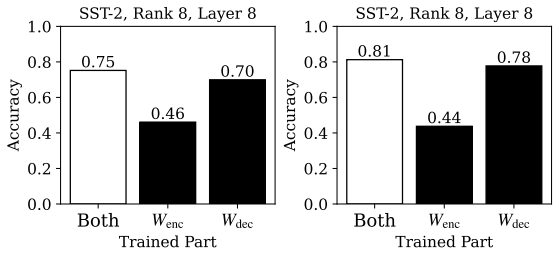


Figure B.26: Augmentation results for Table 4.1, (left) on Qwen 2.5-3B, (right) on Llama 3-8B.

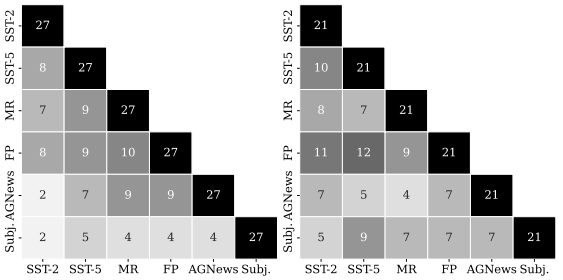


Figure B.27: Augmentation results for Fig. 4.11, (left) on Qwen 2.5-3B, (right) on Llama 3-8B.

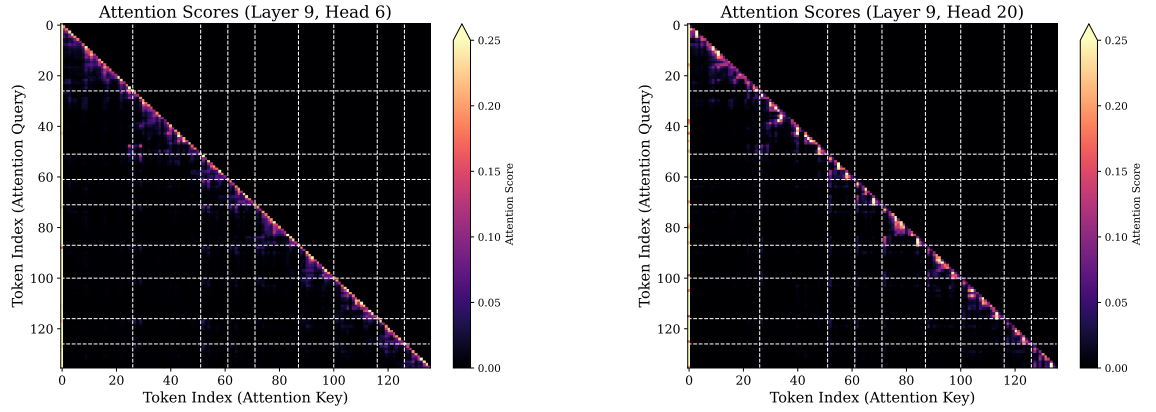


Figure B.28: Augmentation results for attention visualization on (left) Layer 9, Head 6, (right) Layer 9, Head 20, with the inputs shown in Fig. B.2.

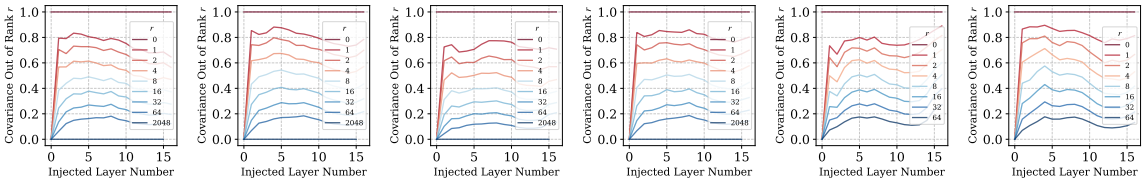


Figure B.29: Numerical results of covariance out of rank on Llama 3.2-1B.

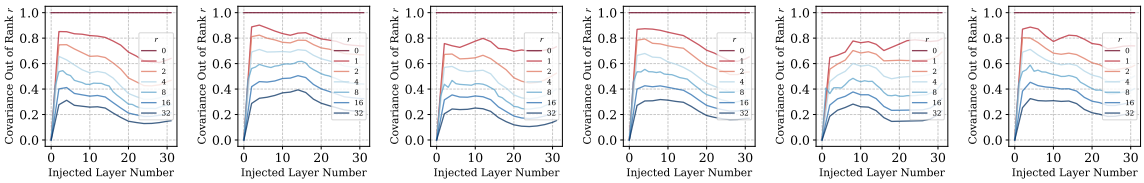


Figure B.30: Numerical results of covariance out of rank on Llama 3-8B[28].

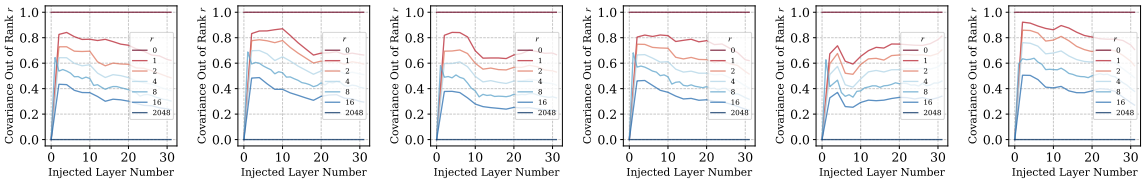


Figure B.31: Numerical results of covariance out of rank on Qwen 2.5-3B[28].

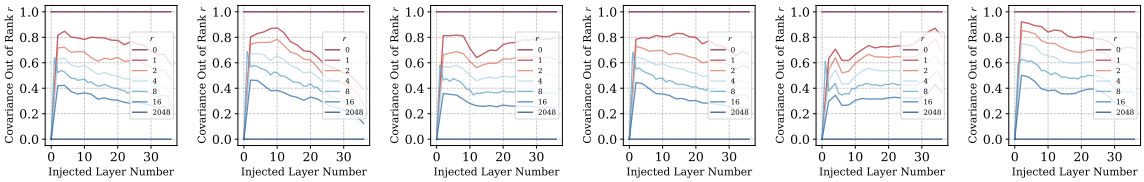


Figure B.32: Numerical results of covariance out of rank on Qwen 2.5-3B Instruct[28].

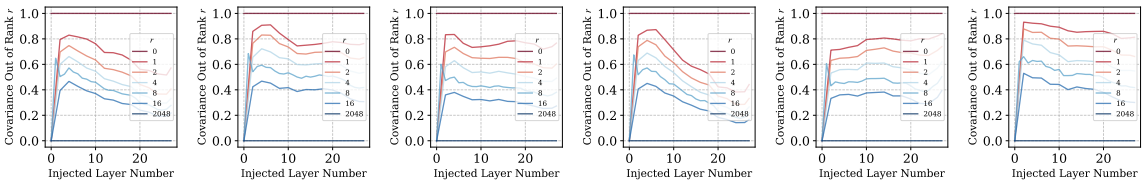


Figure B.33: Numerical results of covariance out of rank on Qwen 2.5-7B[28].

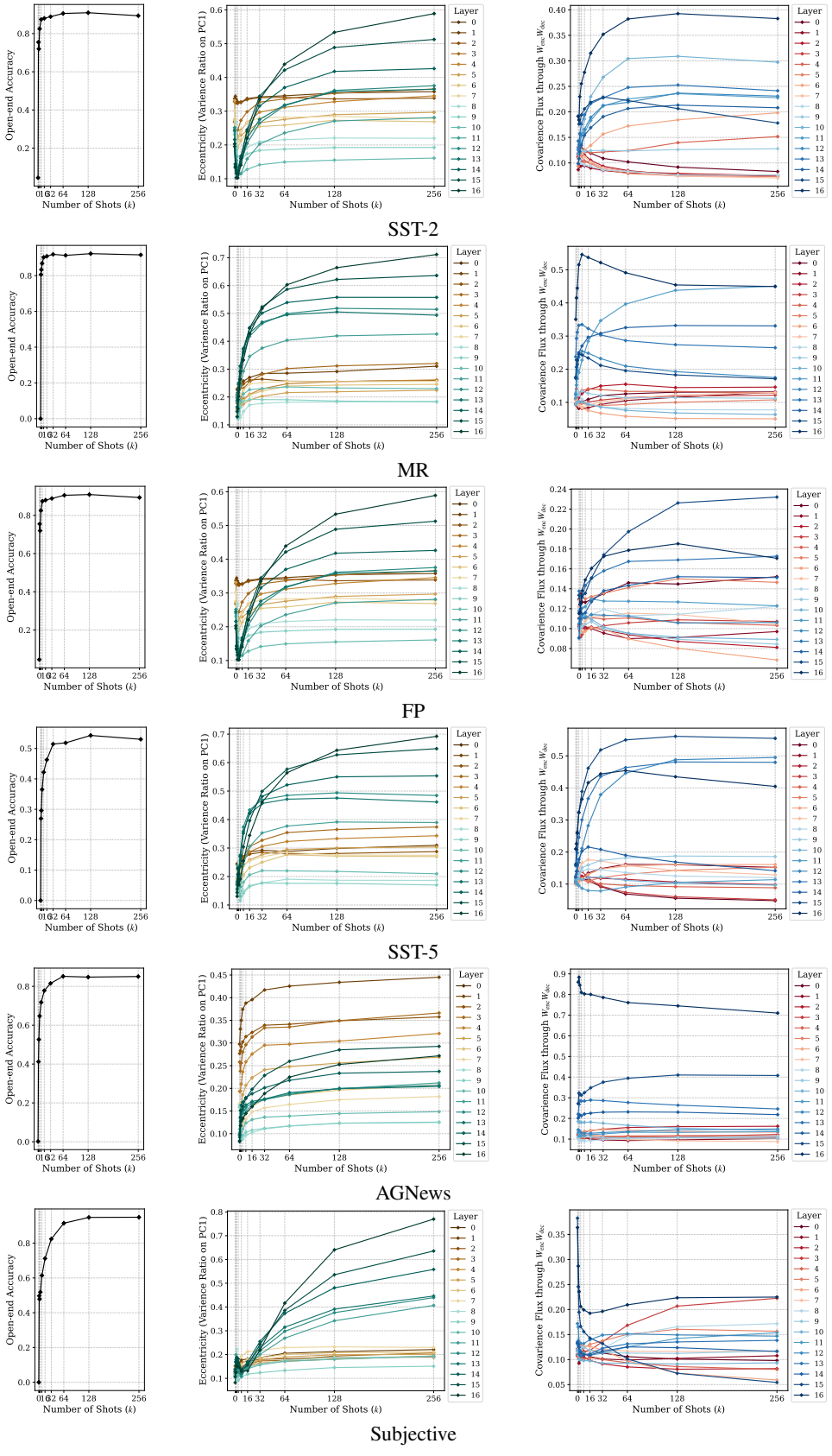


Figure B.34: Augmentation results for Fig. 4.4 on Llama 3.2-1B.

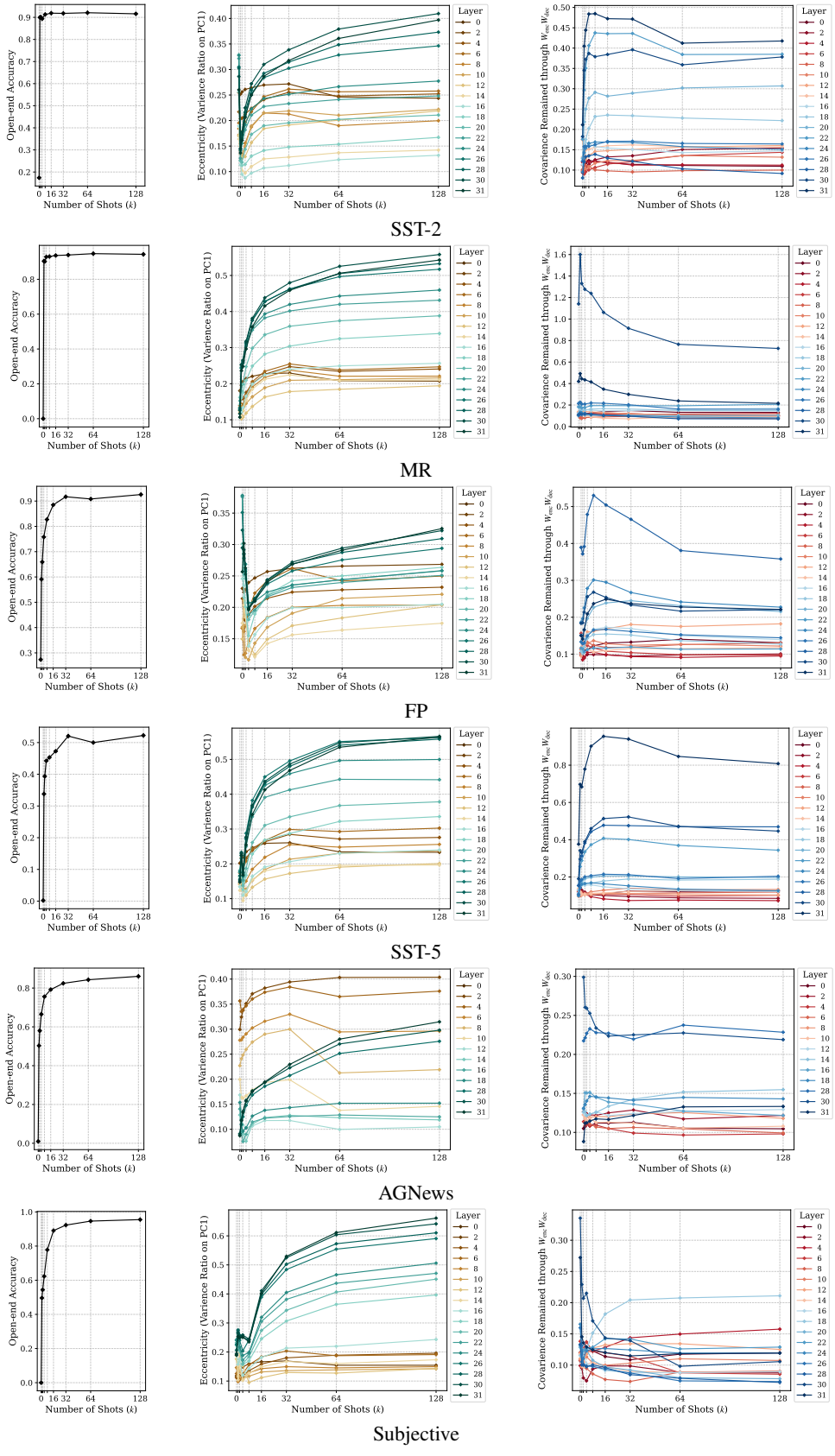


Figure B.35: Augmentation results for Fig. 4.4 on Llama 3-8B.

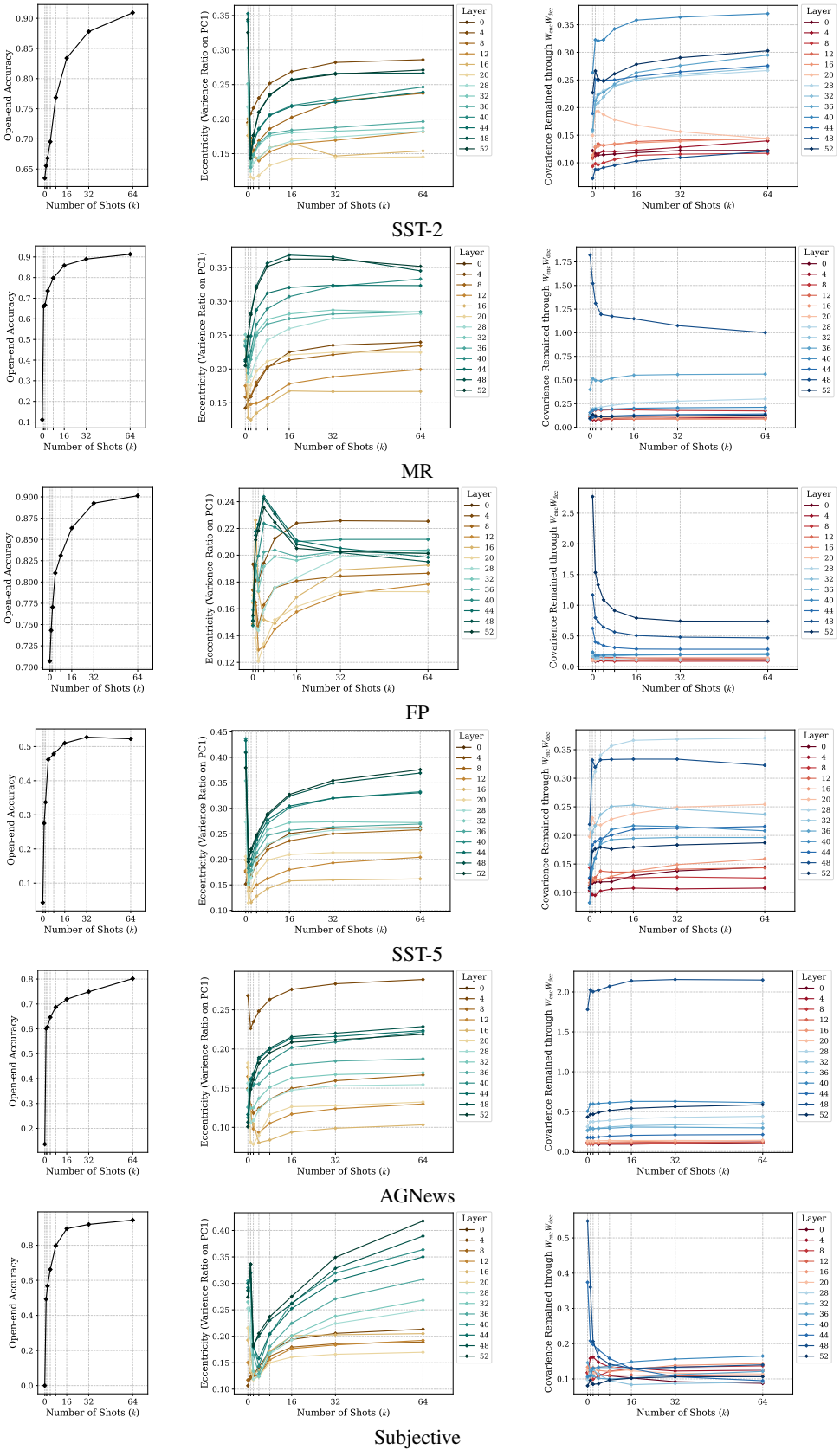


Figure B.36: Augmentation results for Fig. 4.4 on Llama 3-13B.

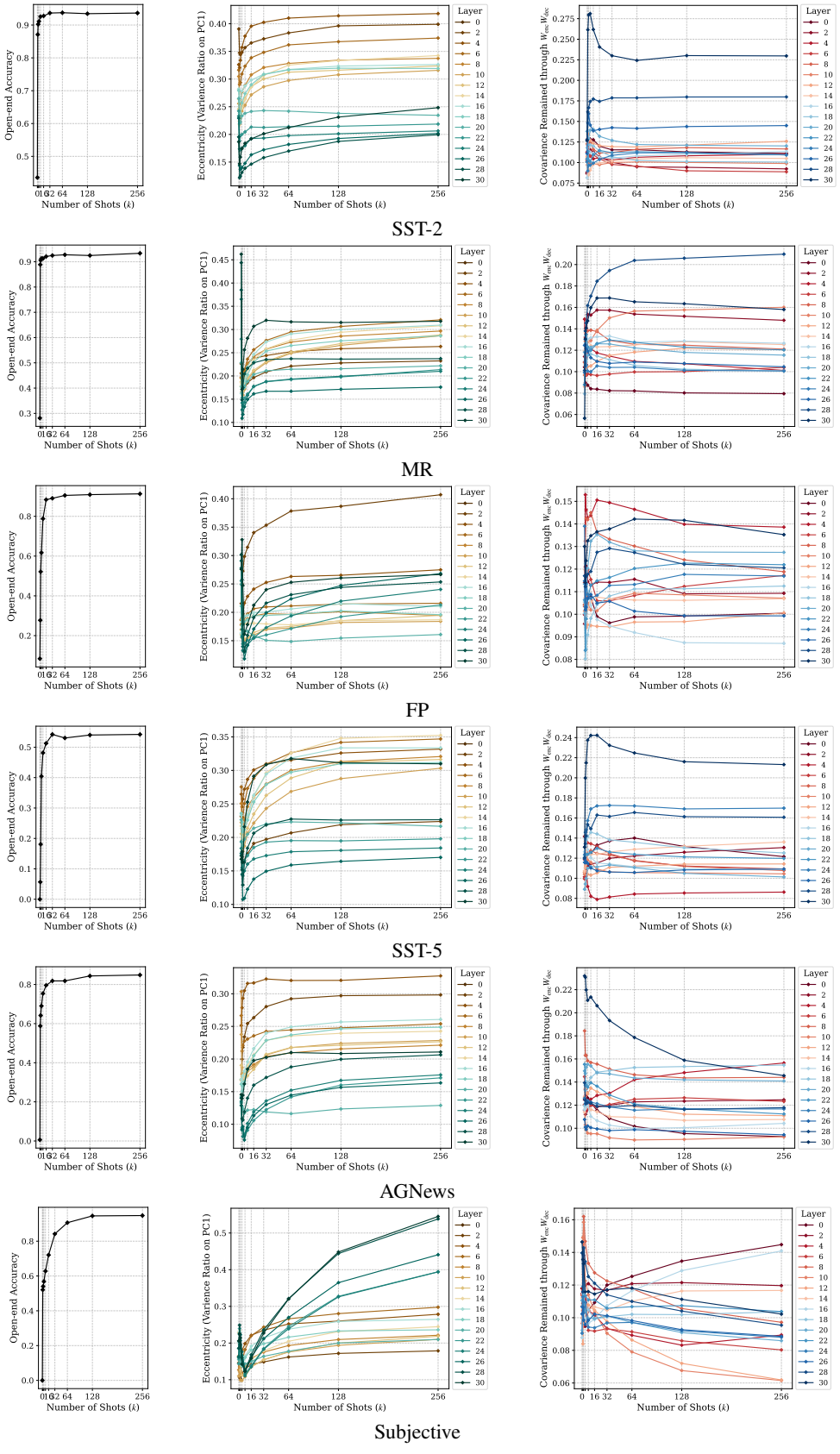


Figure B.37: Augmentation results for Fig. 4.4 on Qwen 2.5-3B.

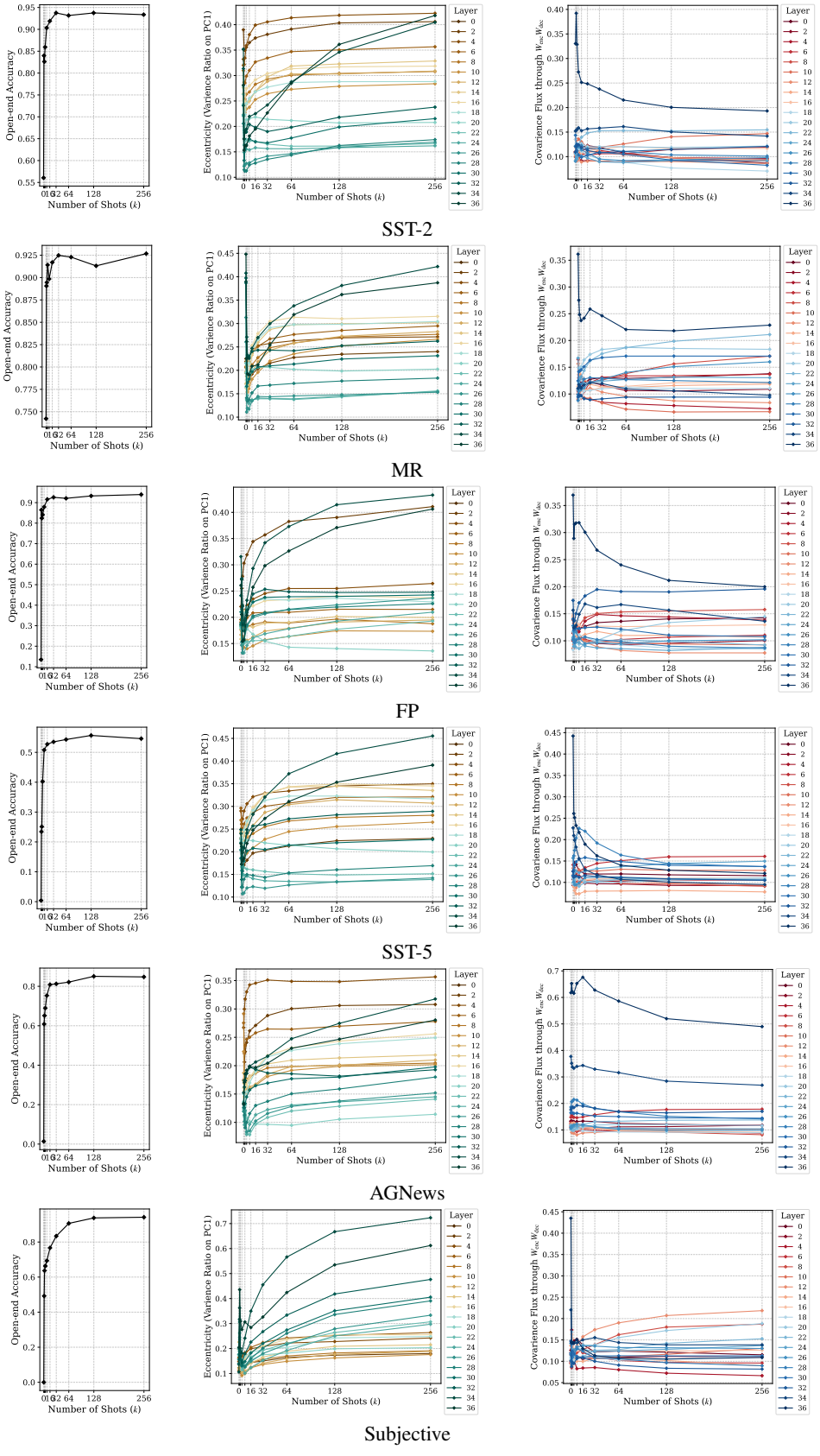


Figure B.38: Augmentation results for Fig. 4.4 on Qwen 2.5-3B Instruct.

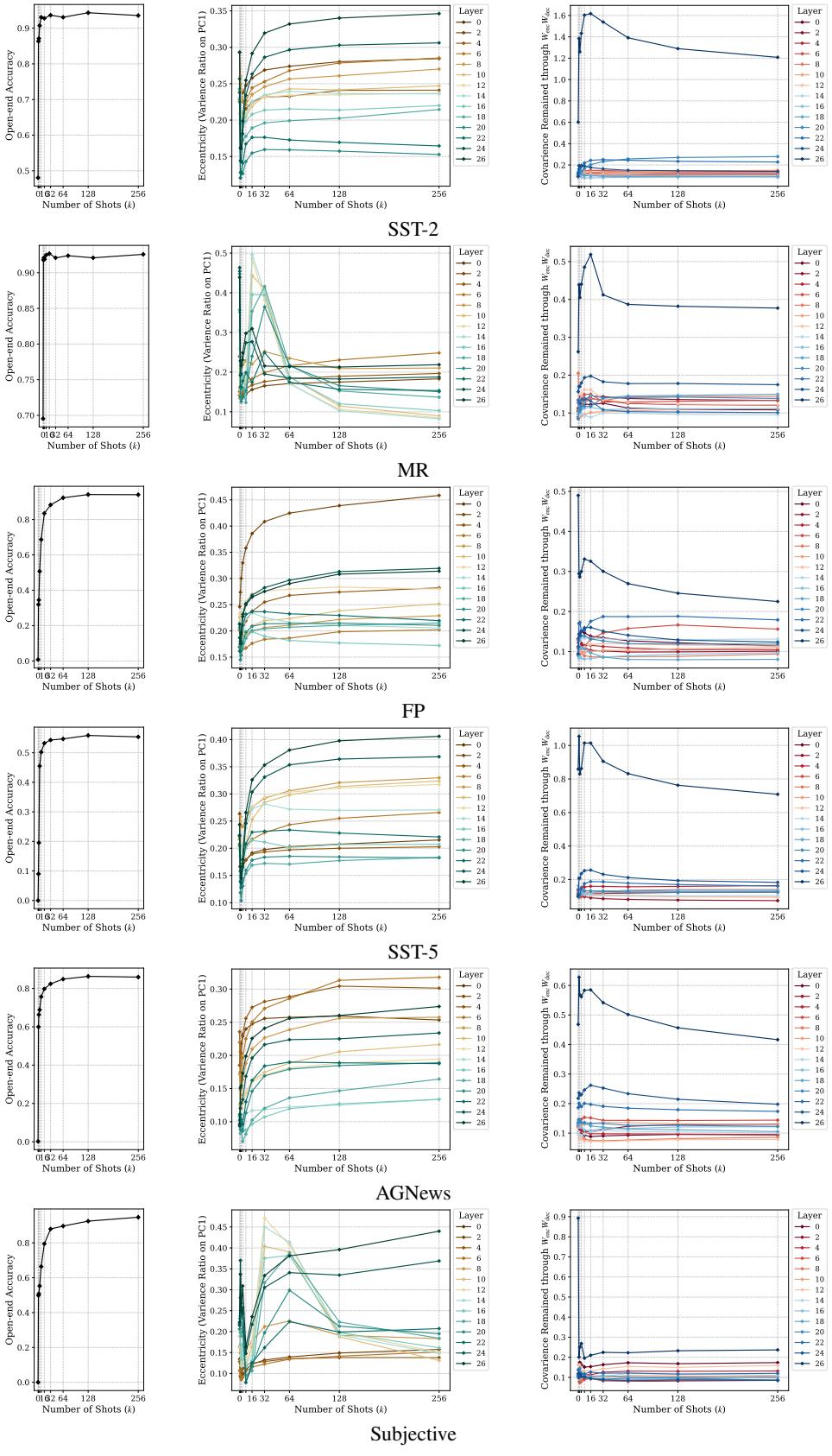


Figure B.39: Augmentation results for Fig. 4.4 on Qwen 2.5-7B.

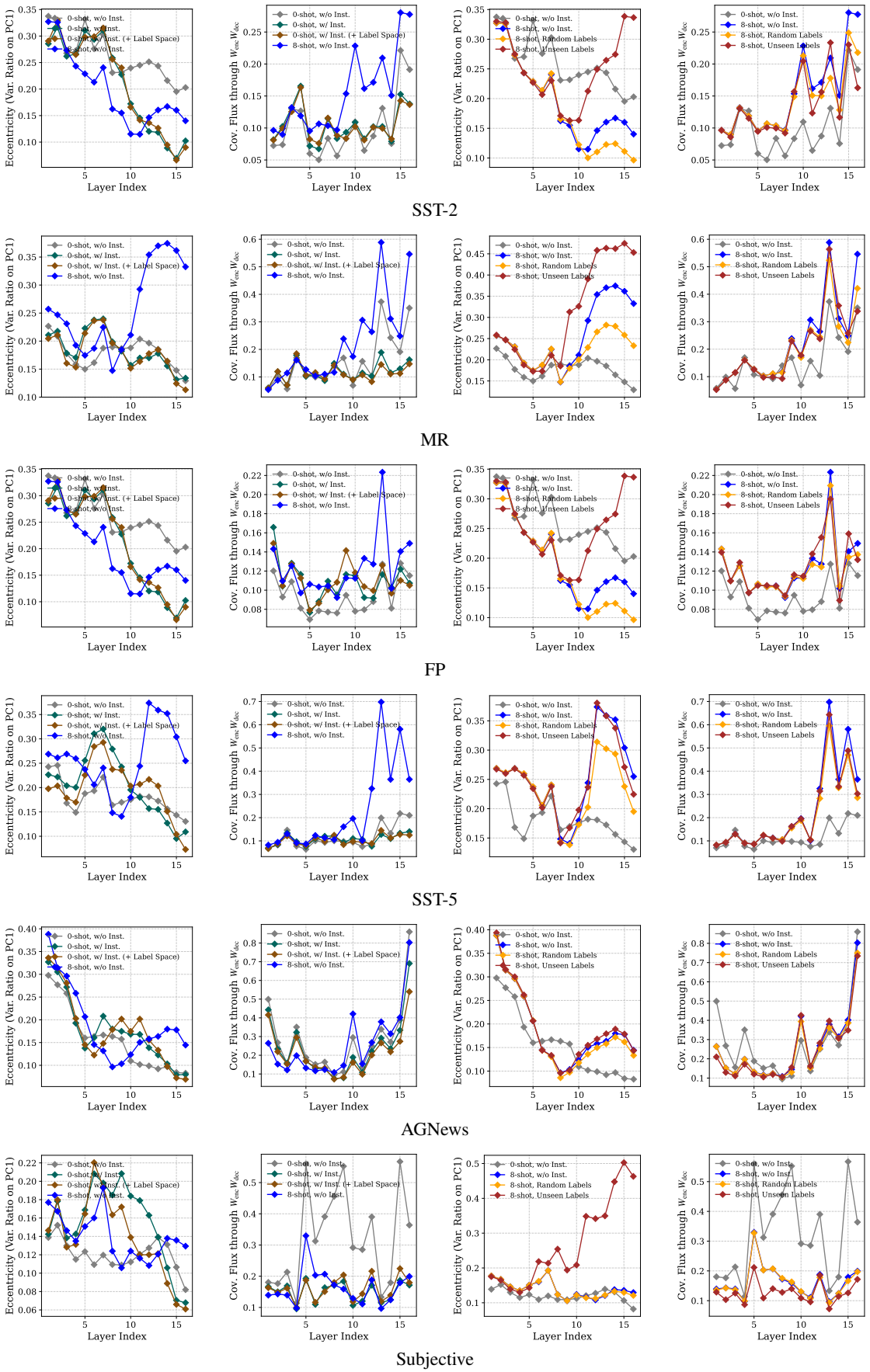
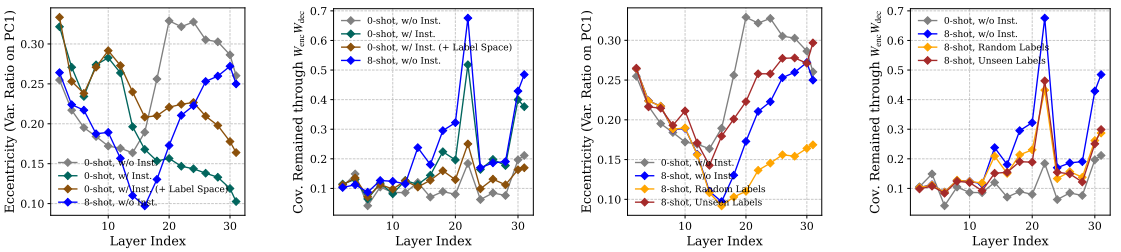
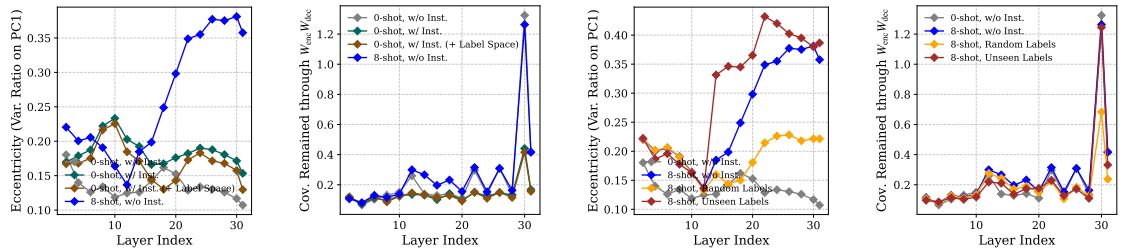


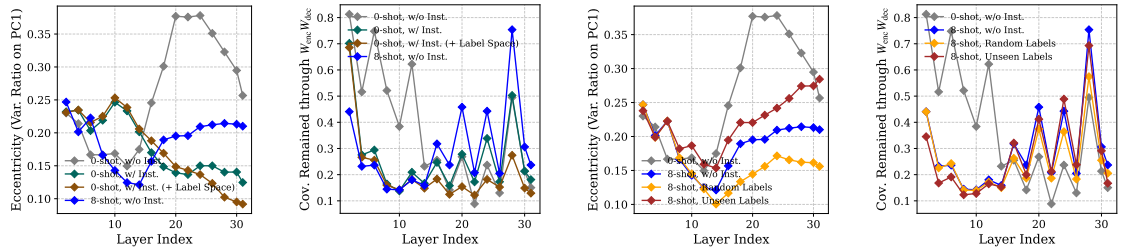
Figure B.40: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3.2-1B.



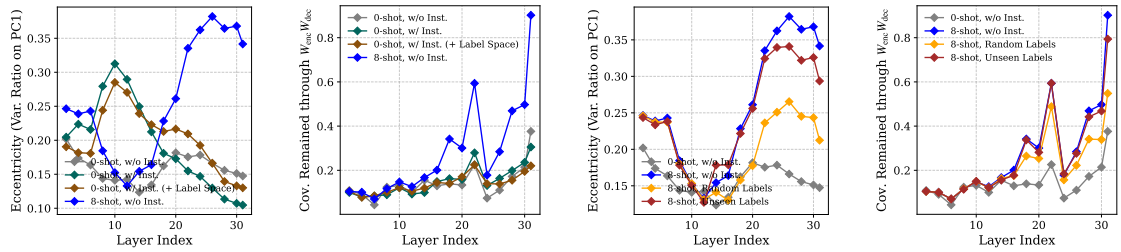
SST-2



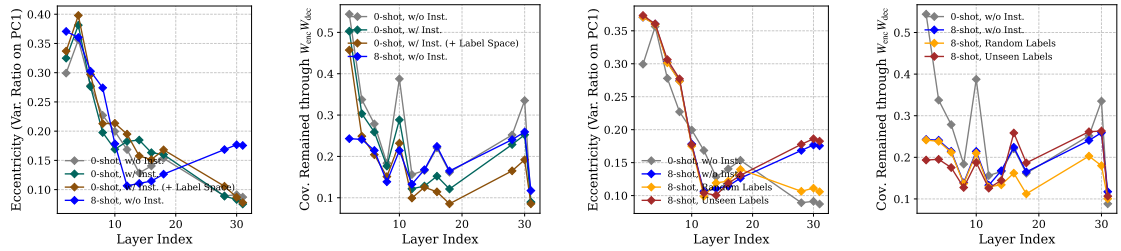
MR



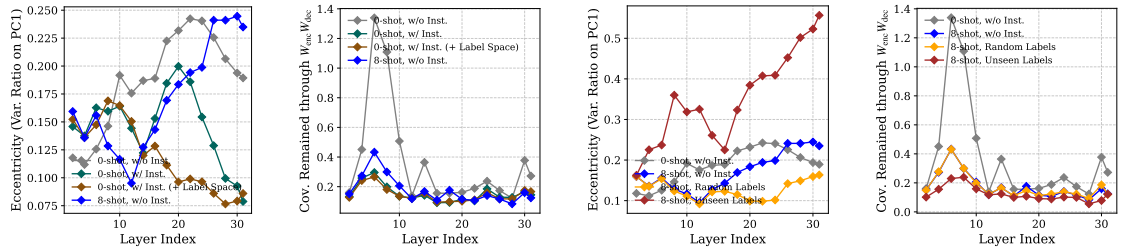
FP



SST-5

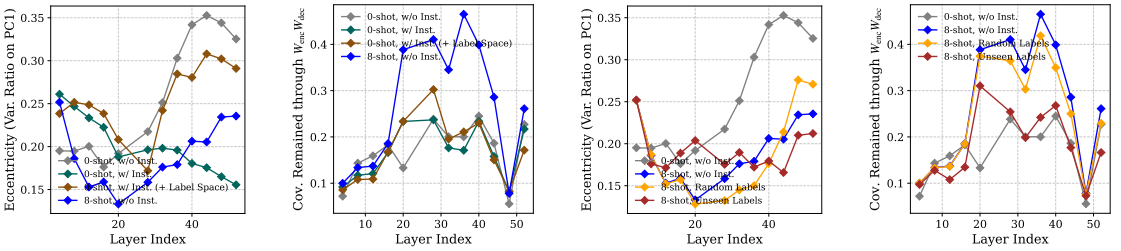


AGNews

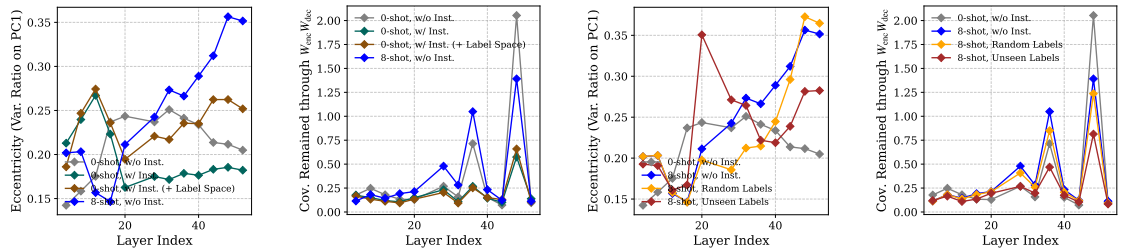


Subjective

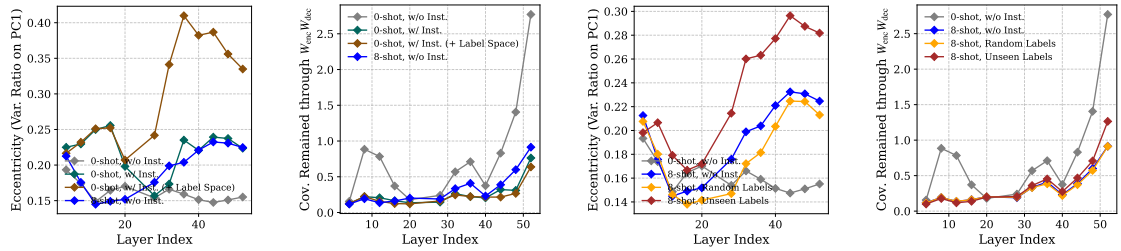
Figure B.41: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3-8B.



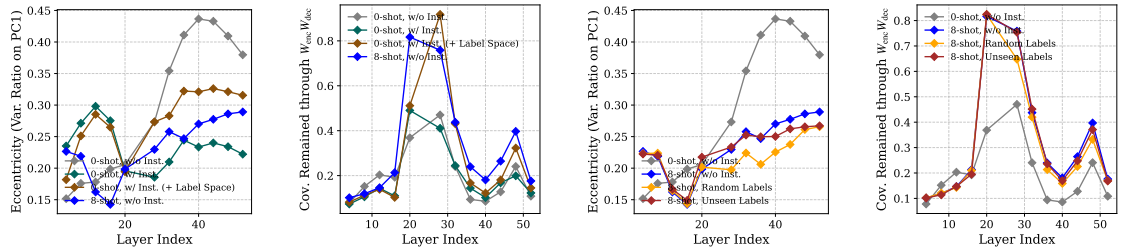
SST-2



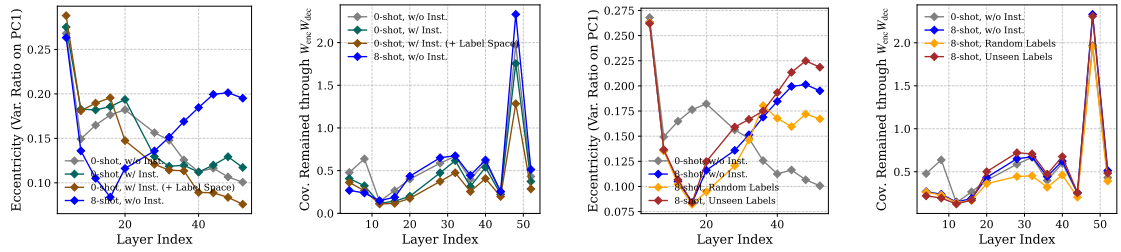
MR



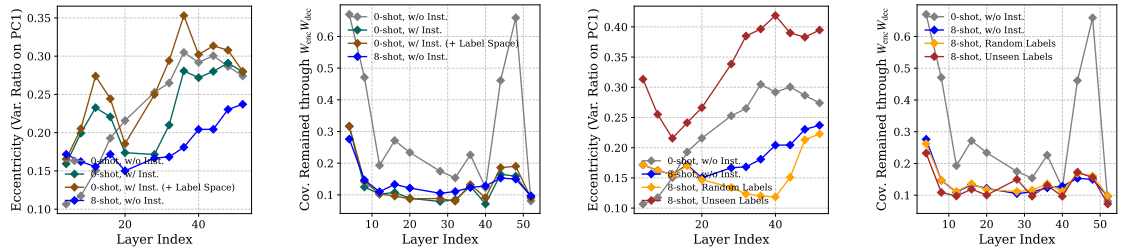
FP



SST-5

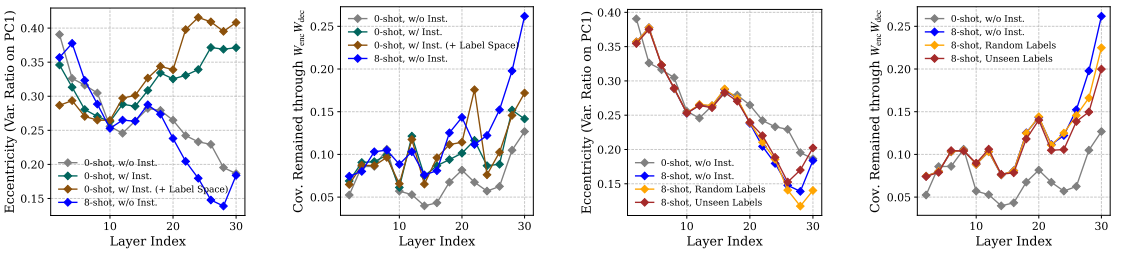


AGNews

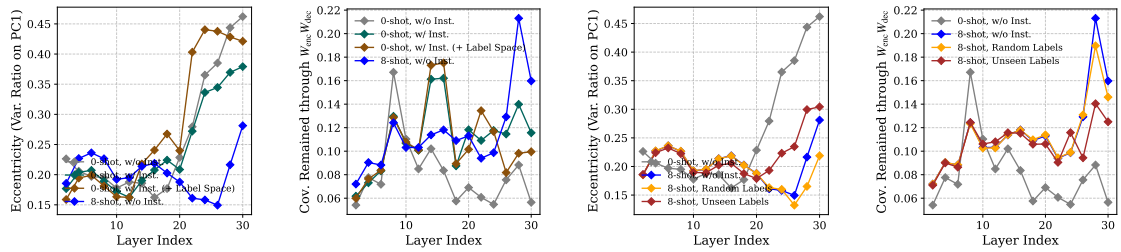


Subjective

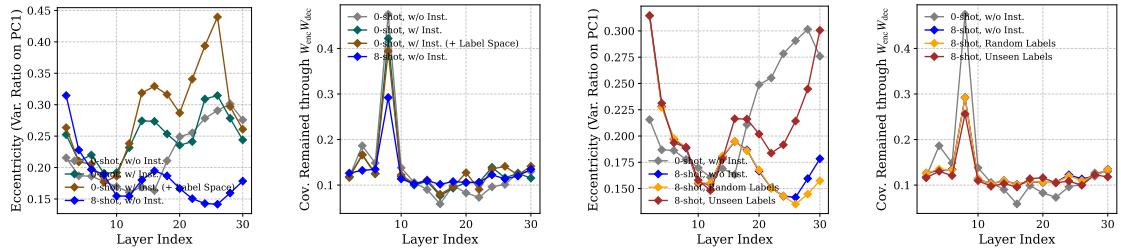
Figure B.42: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Llama 3-13B.



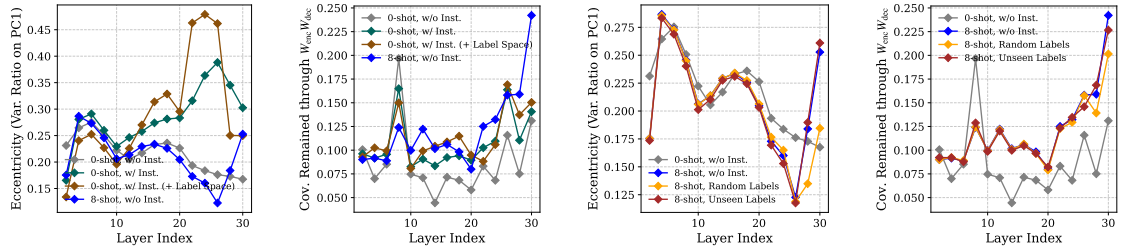
SST-2



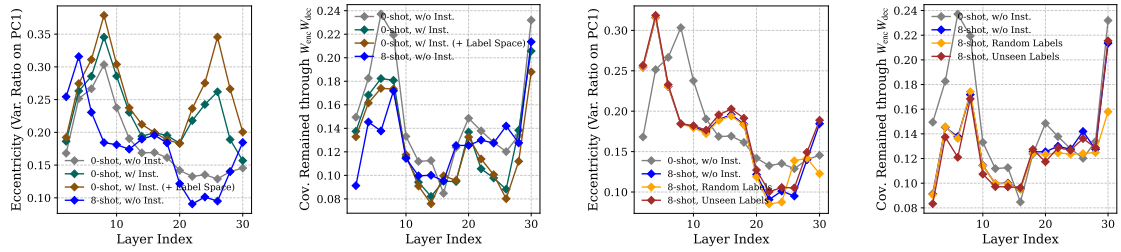
MR



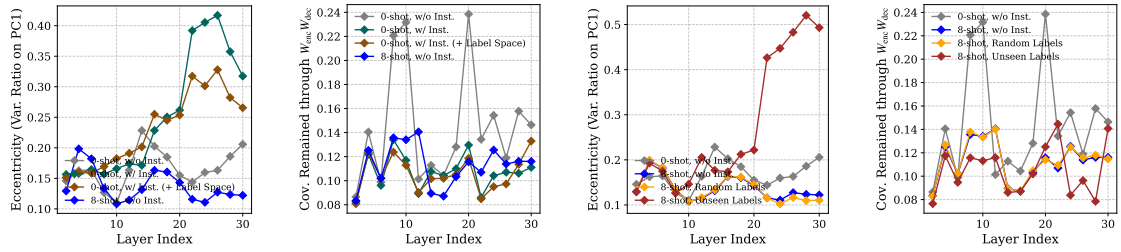
FP



SST-5

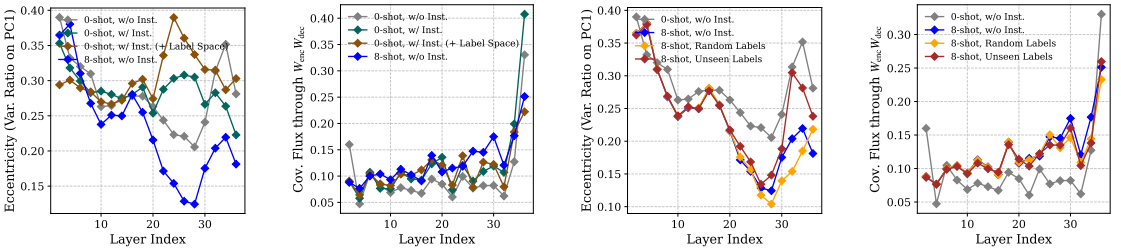


AGNews

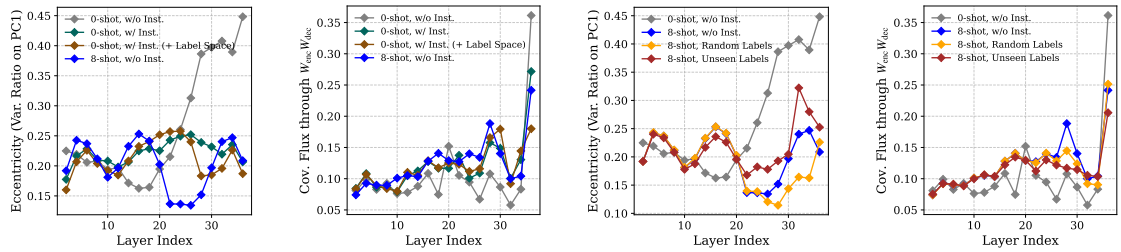


Subjective

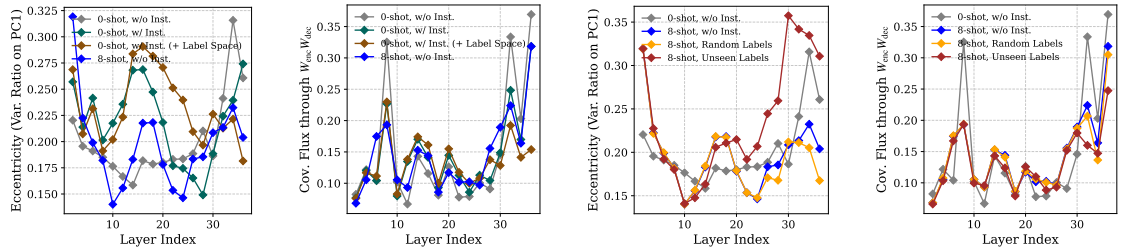
Figure B.43: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-3B.



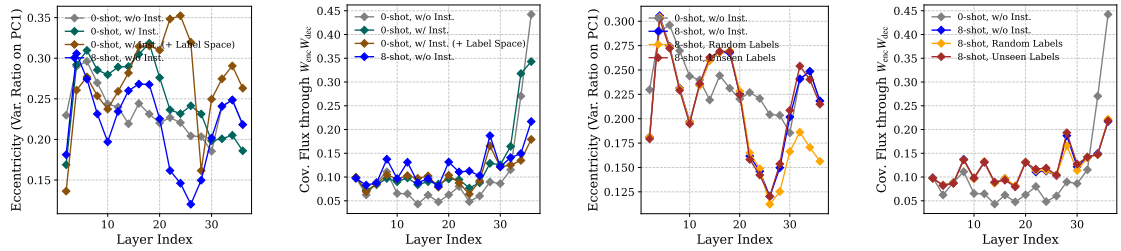
SST-2



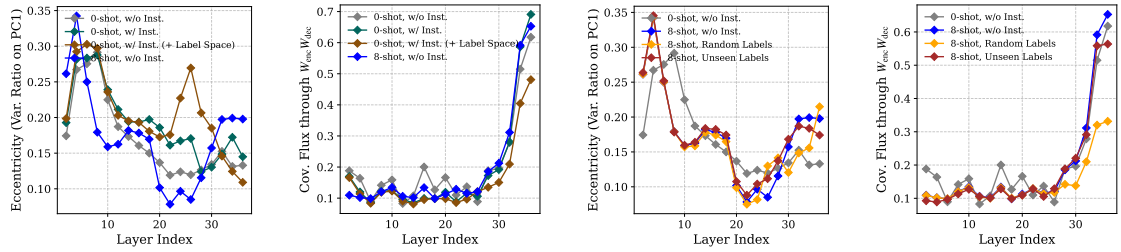
MR



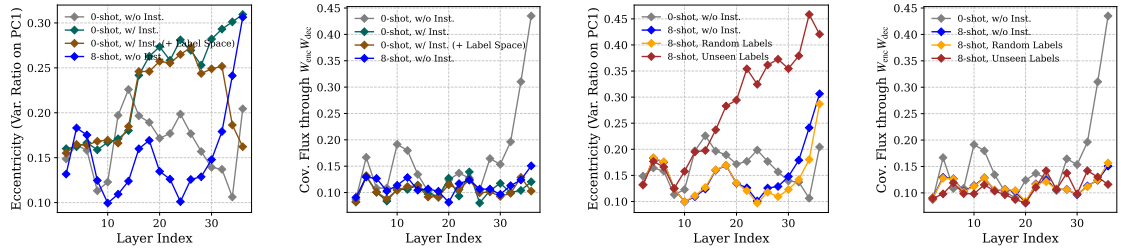
FP



SST-5

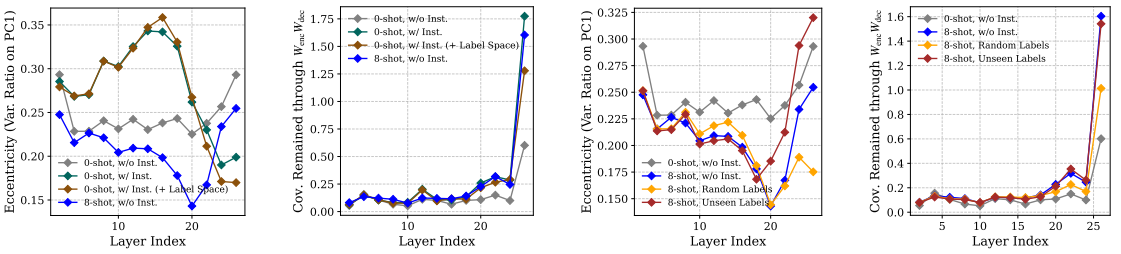


AGNews

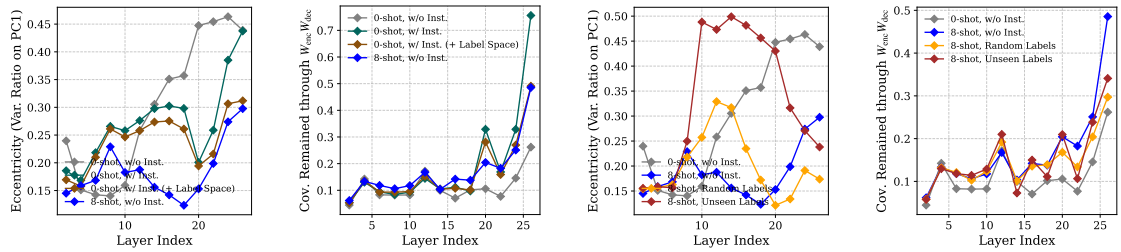


Subjective

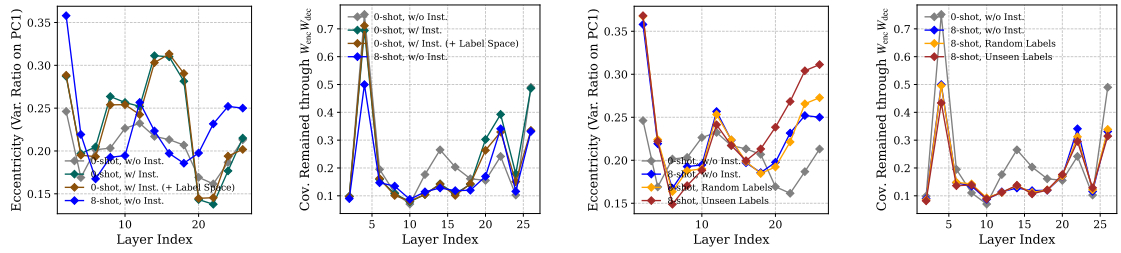
Figure B.44: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-3B Instruct.



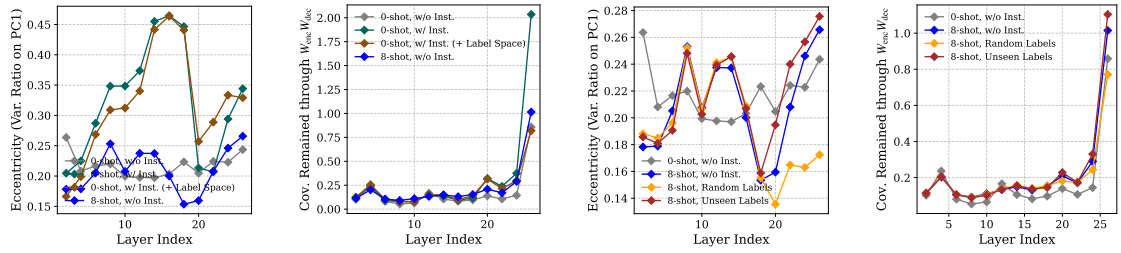
SST-2



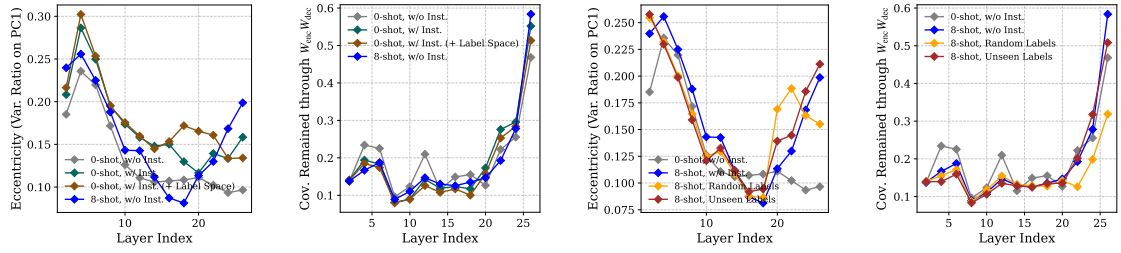
MR



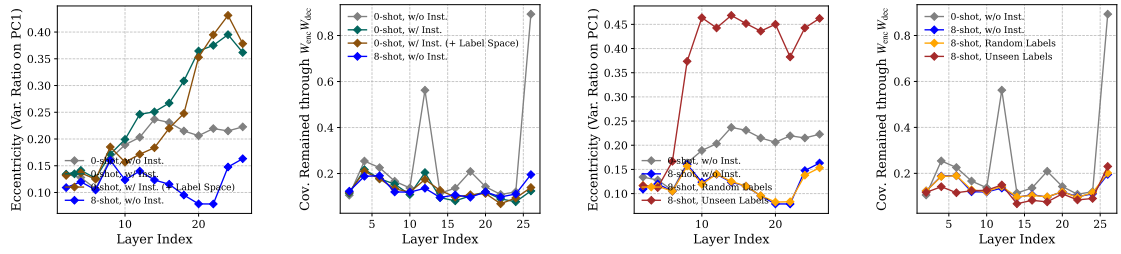
FP



SST-5



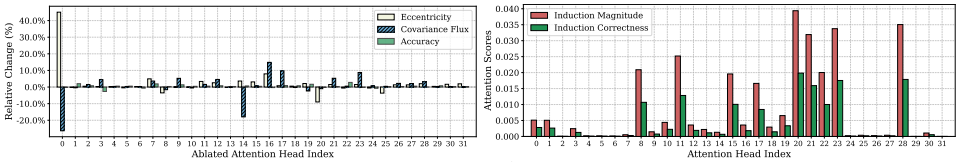
AGNews



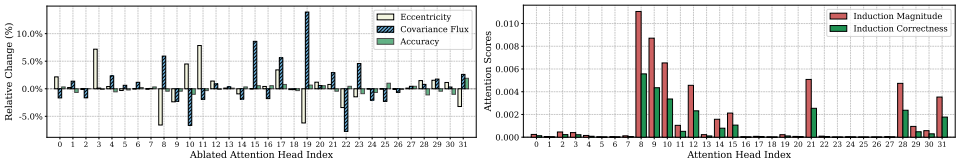
Subjective

Figure B.45: (Left 2) augmentation results for Fig. 4.5, (right 2) for Fig. 4.6 on Qwen 2.5-7B.

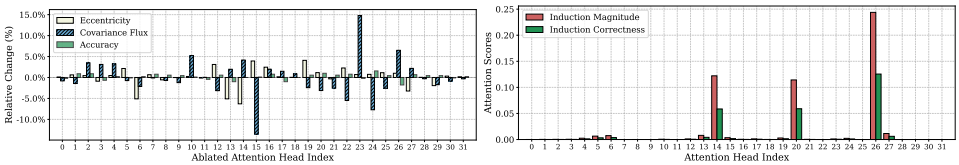
Layer 0



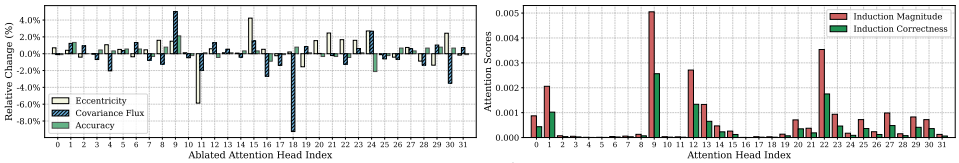
Layer 1



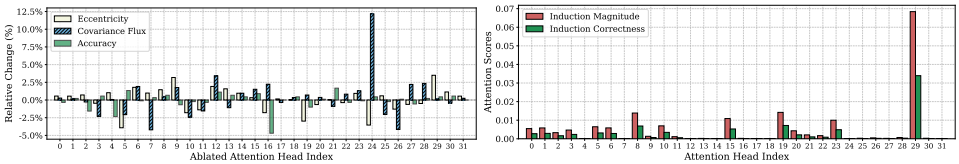
Layer 2



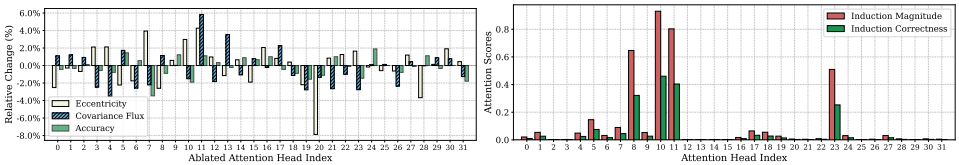
Layer 3



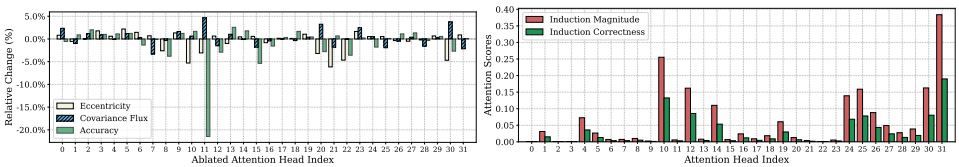
Layer 4



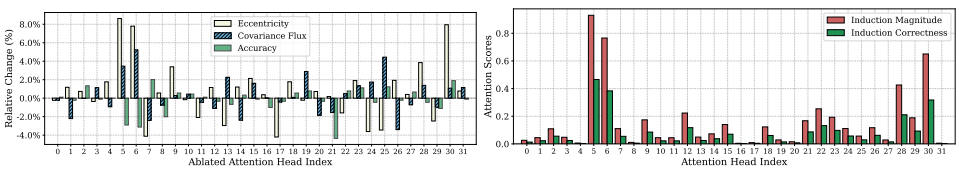
Layer 5



Layer 6



Layer 7



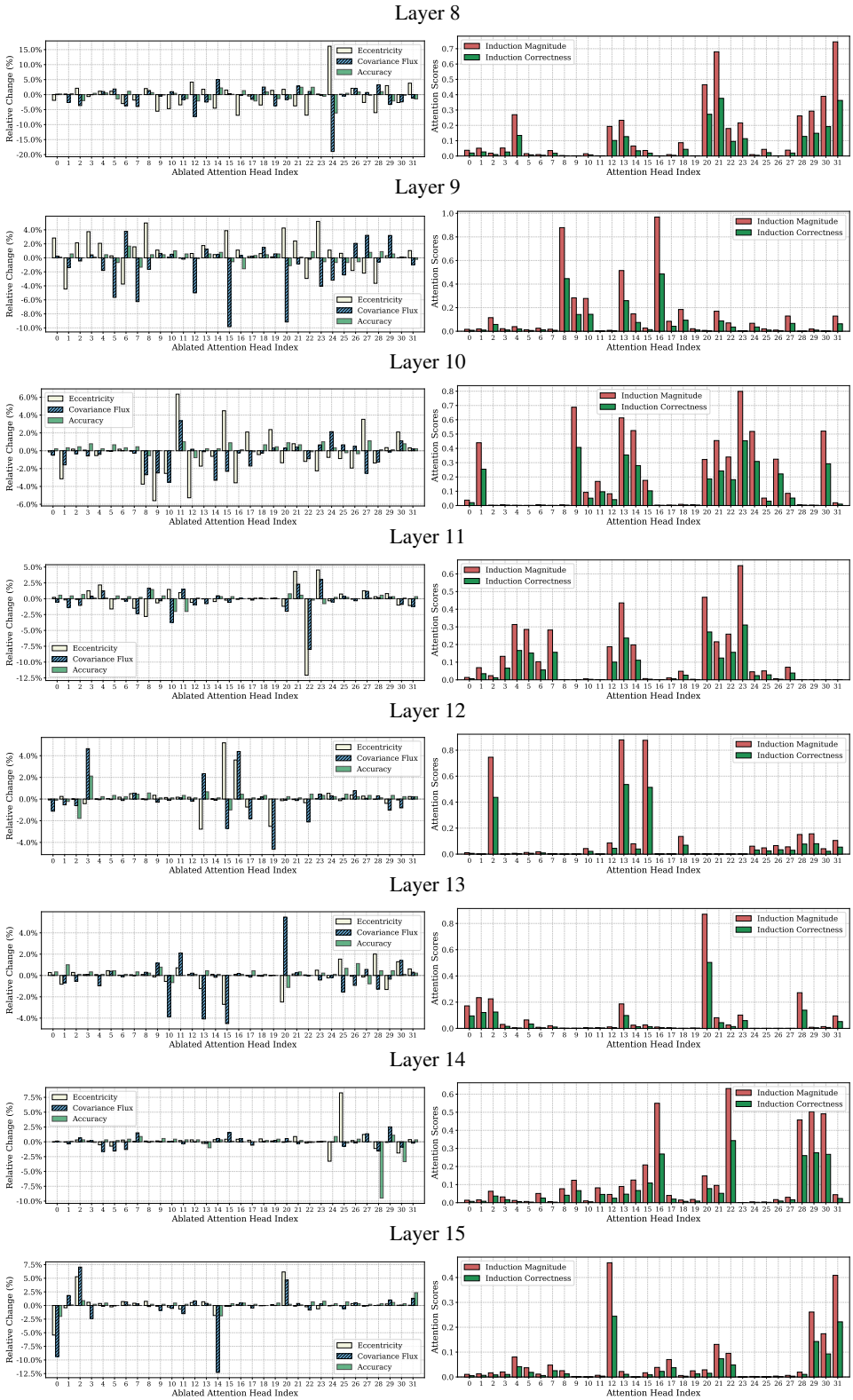


Figure B.46: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, SST-2.

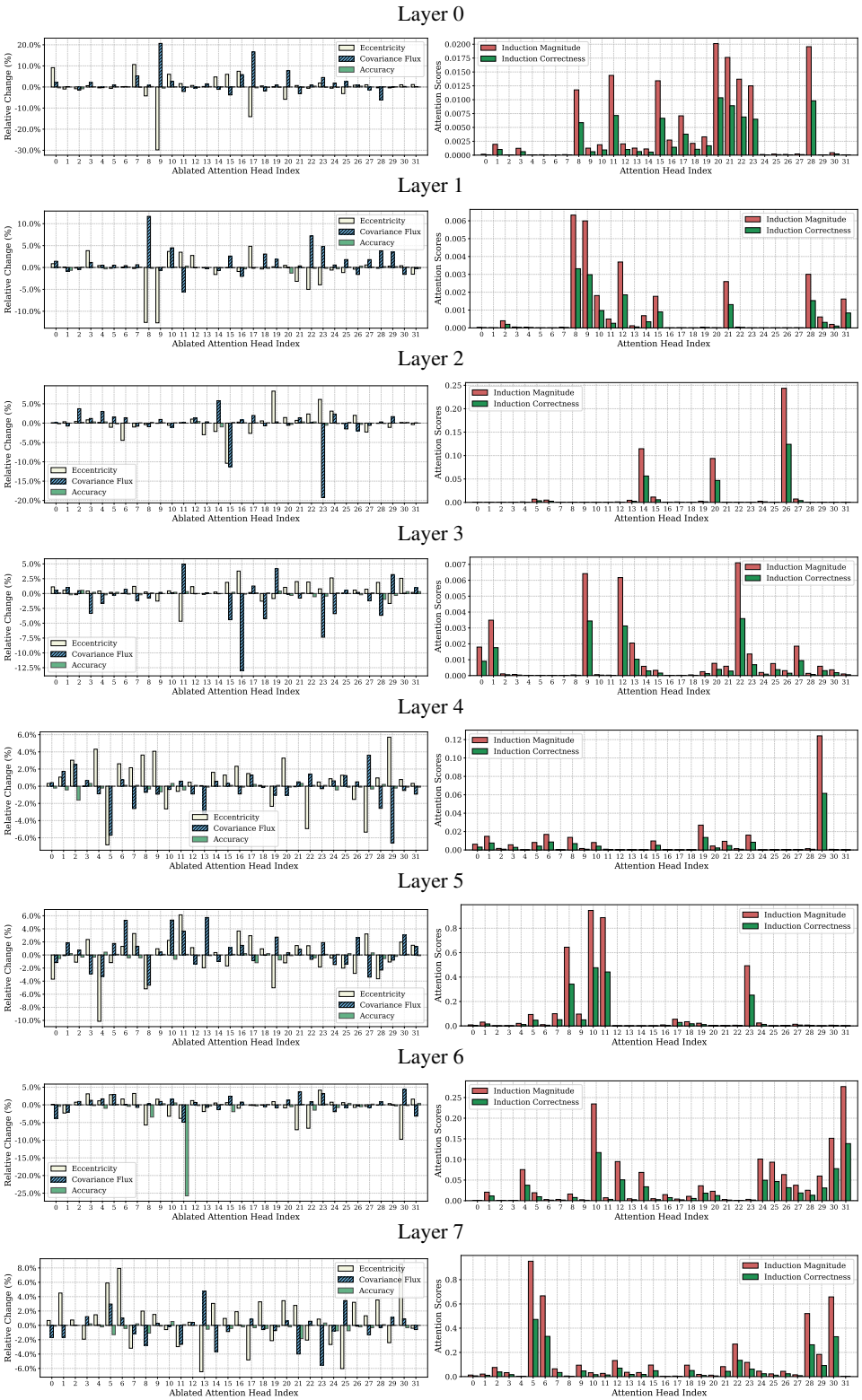
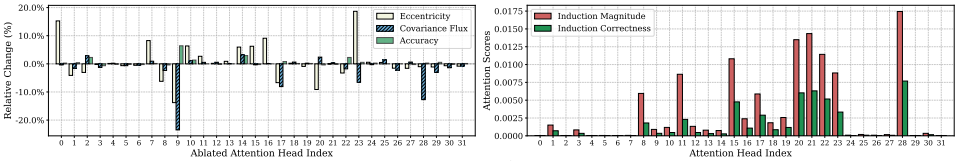

Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Relative Change (%)
Ablated Attention Head Index
Eccentricity
Covariance Flux
Accuracy
Attention Scores
Attention Head Index
Induction Magnitude
Induction Correctness

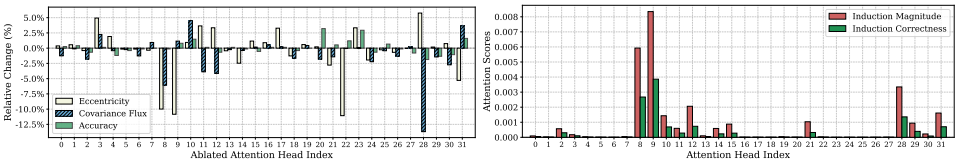


Figure B.47: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, MR.

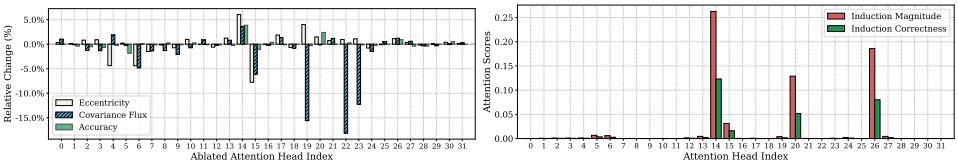
Layer 0



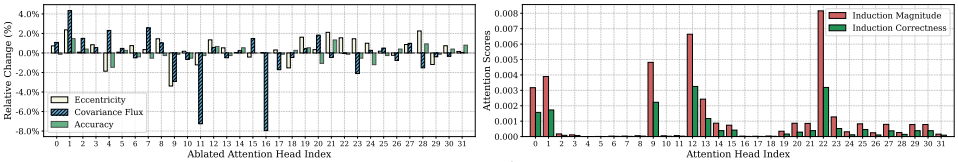
Layer 1



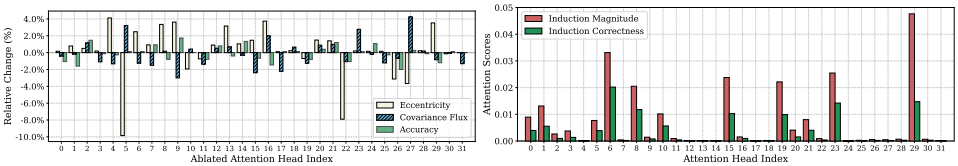
Layer 2



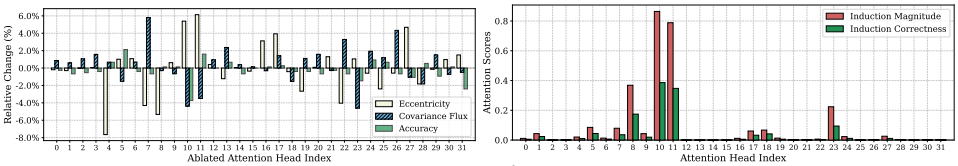
Layer 3



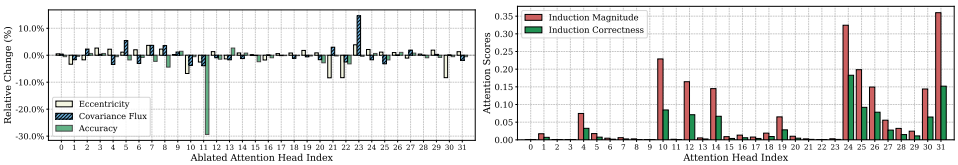
Layer 4



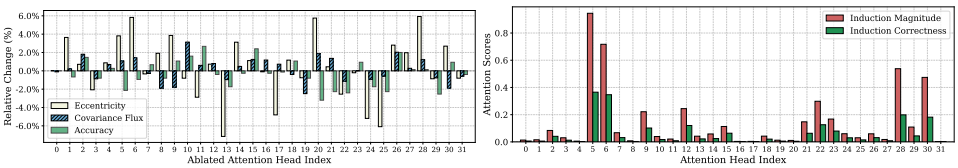
Layer 5



Layer 6



Layer 7



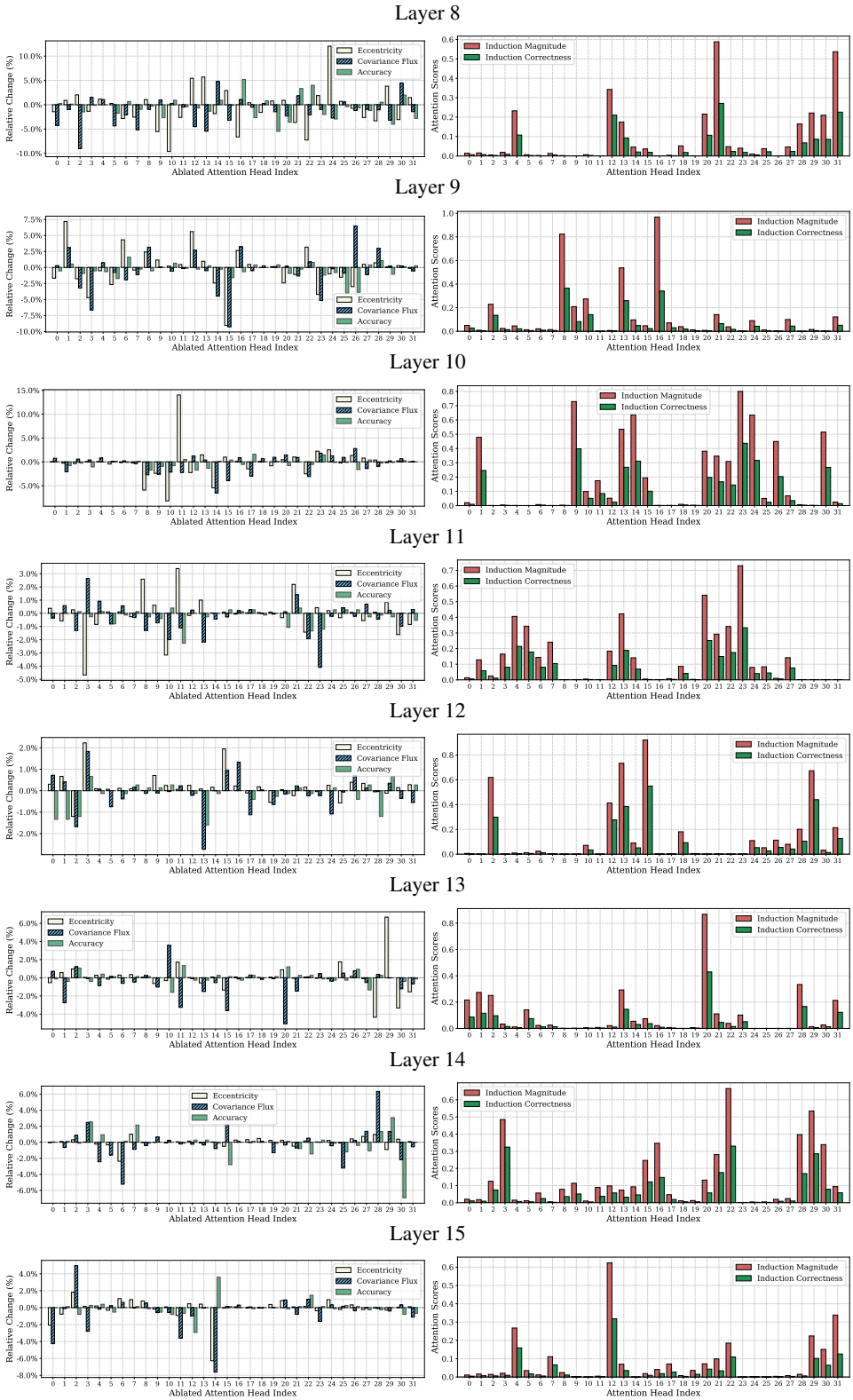
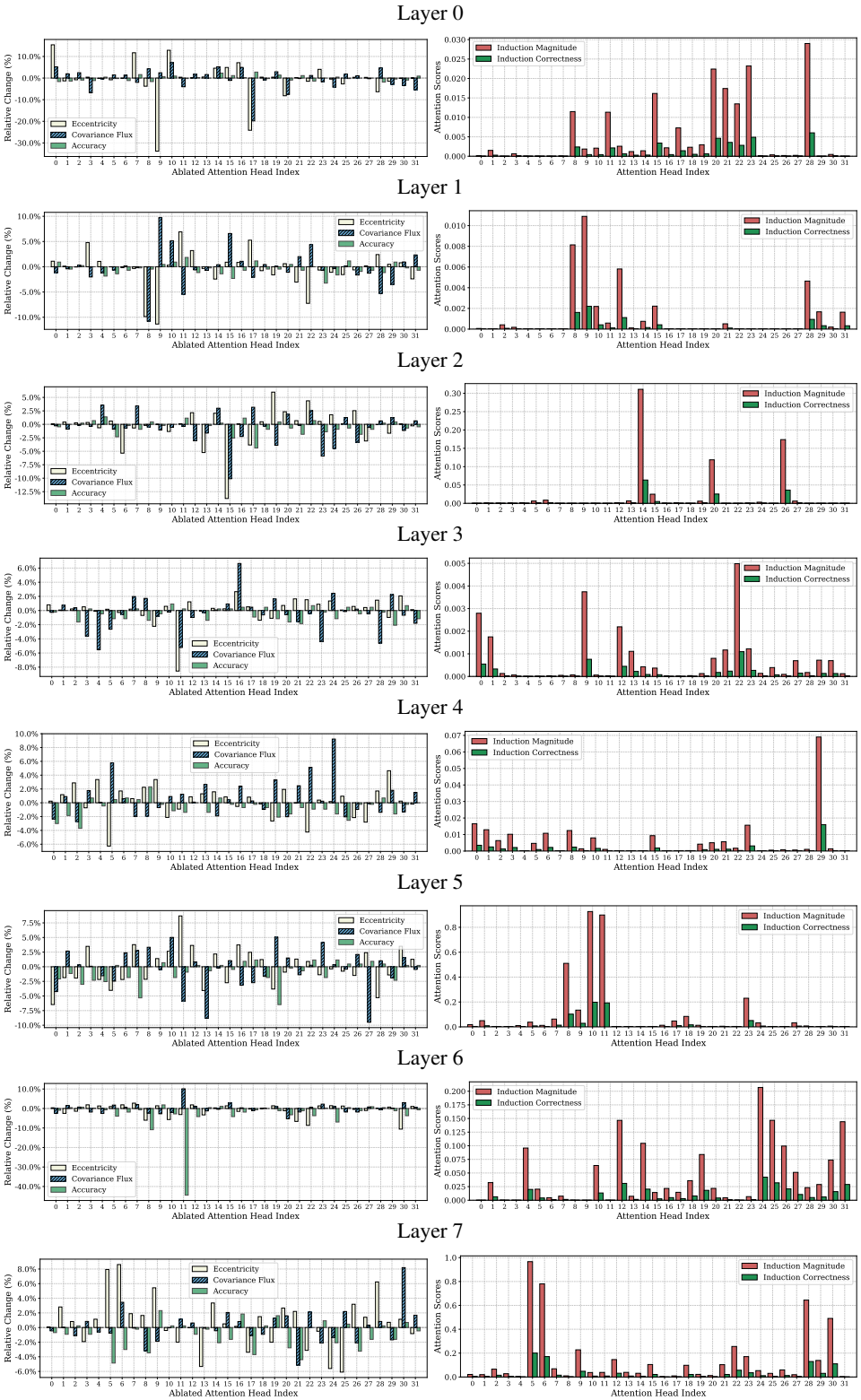


Figure B.48: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, FP.


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Relative Change (%)
Ablated Attention Head Index
Eccentricity
Covariance Flux
Accuracy
Attention Scores
Attention Head Index
Induction Magnitude
Induction Correctness

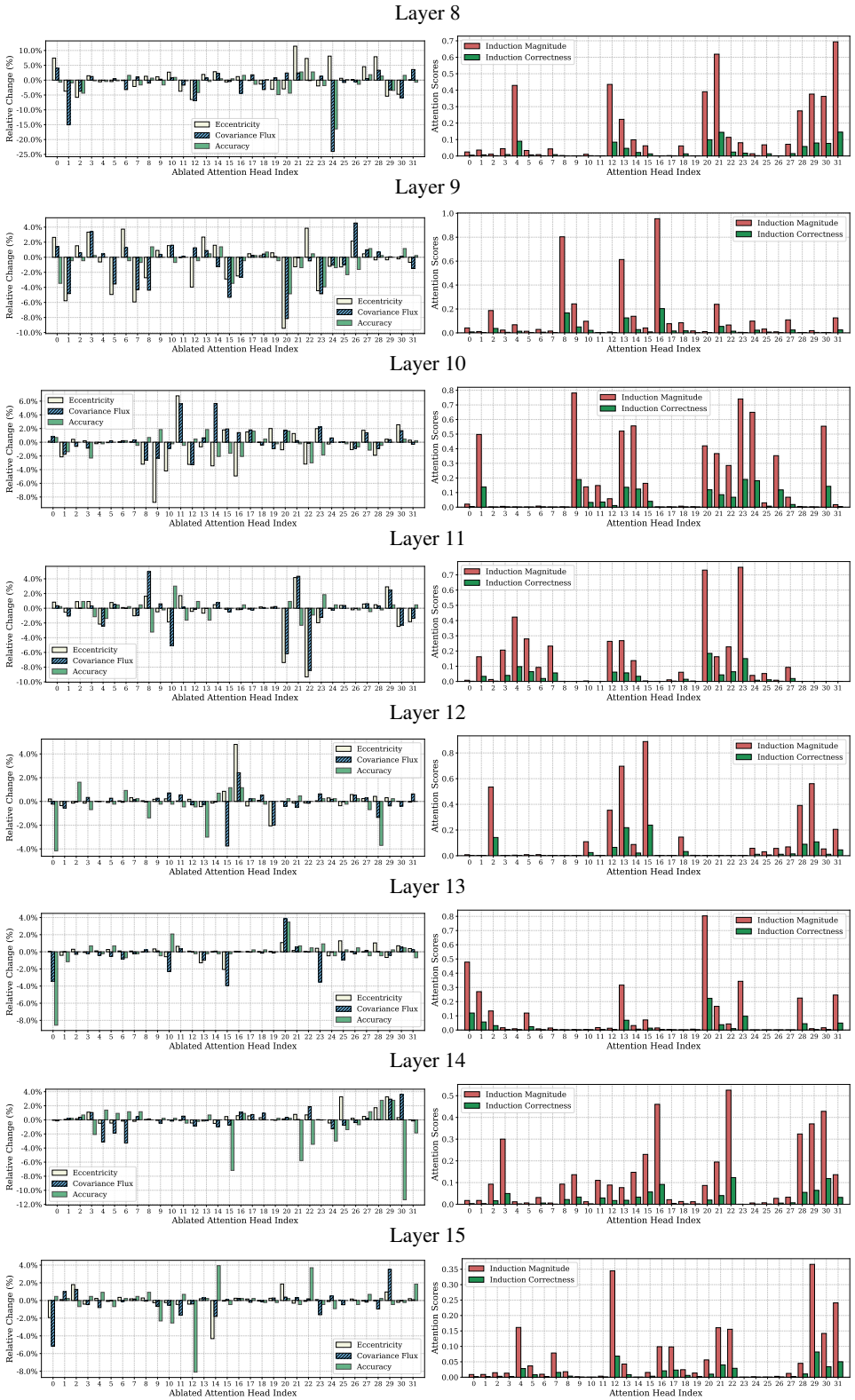
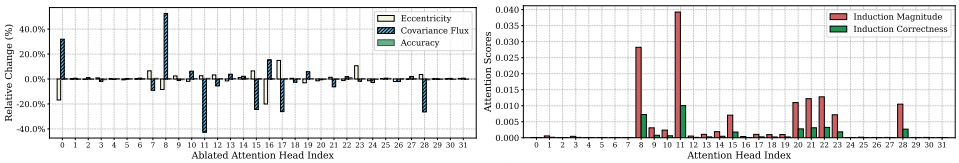
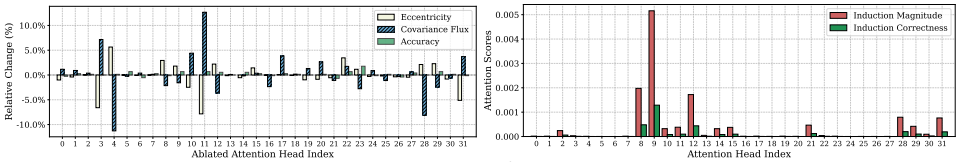


Figure B.49: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, SST-5.

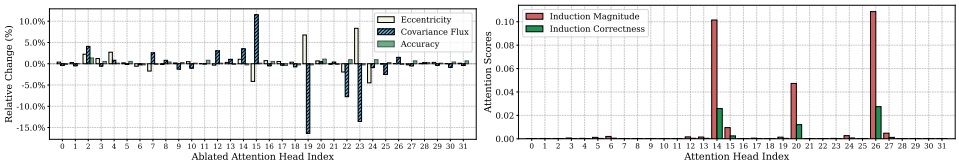
Layer 0



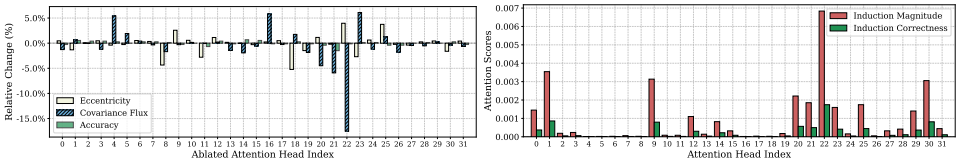
Layer 1



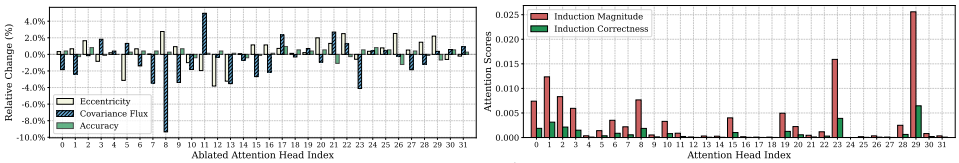
Layer 2



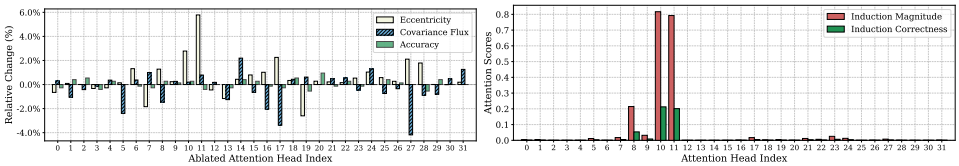
Layer 3



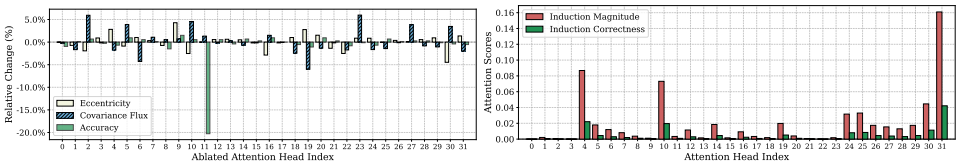
Layer 4



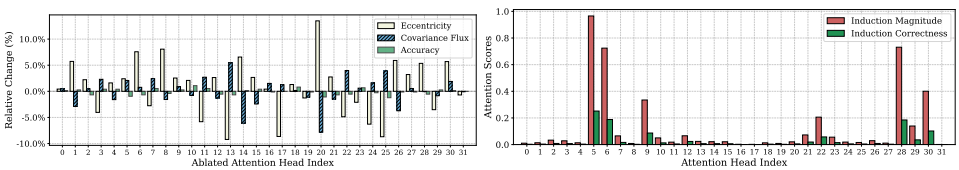
Layer 5



Layer 6



Layer 7



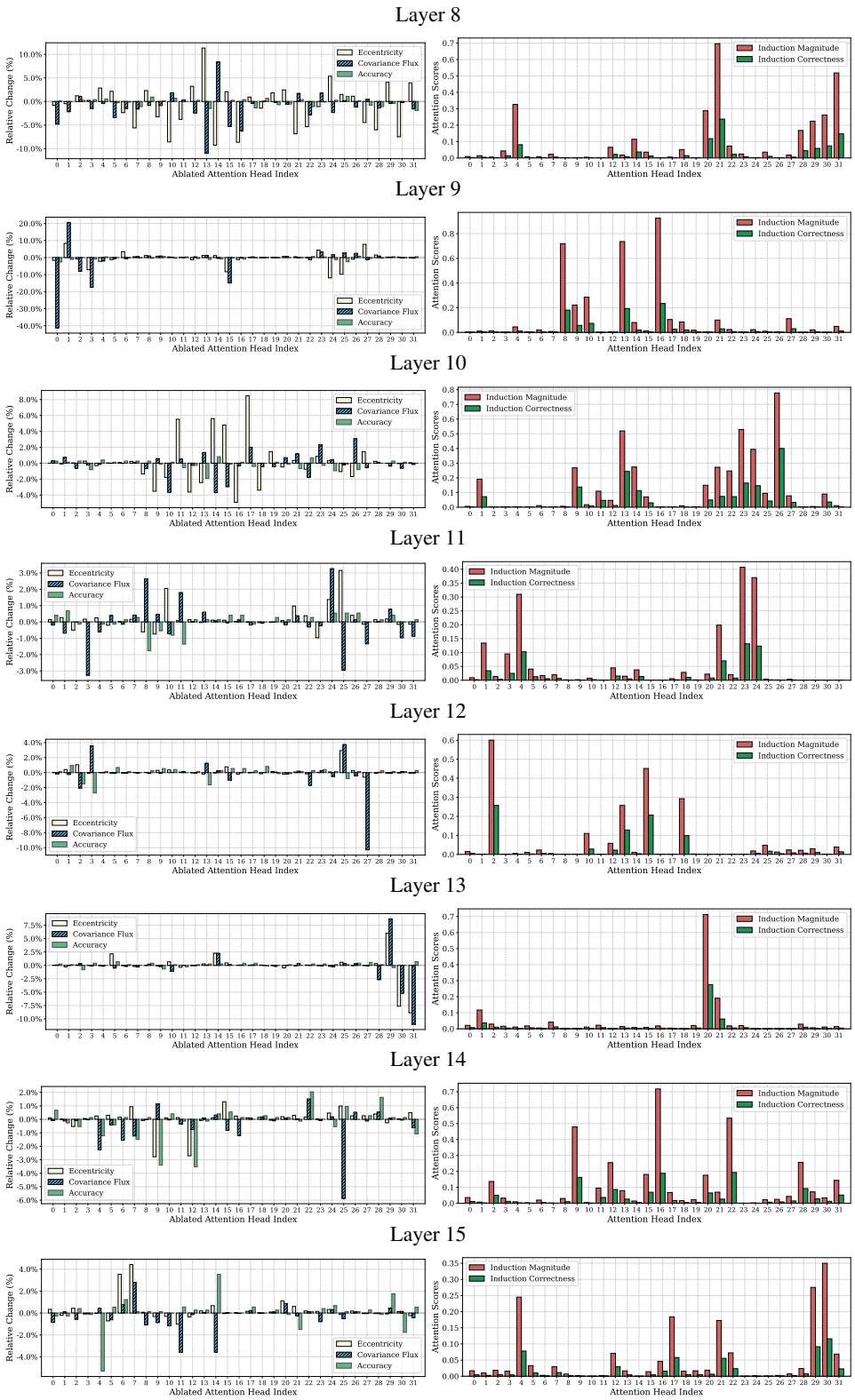
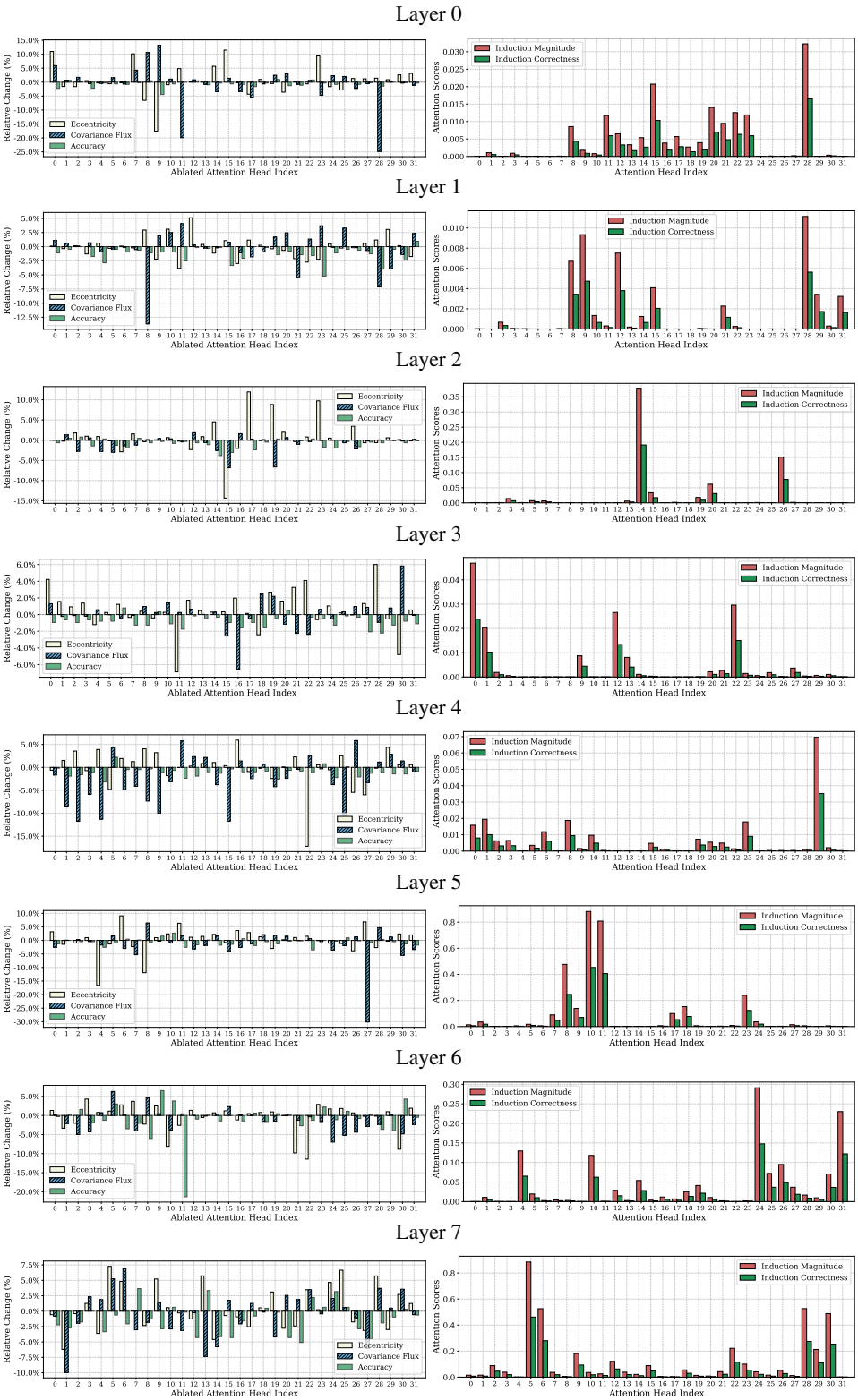


Figure B.50: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, AGNews.


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Relative Change (%)
Ablated Attention Head Index
Eccentricity
Covariance Flux
Accuracy
Attention Scores
Attention Head Index
Induction Magnitude
Induction Correctness

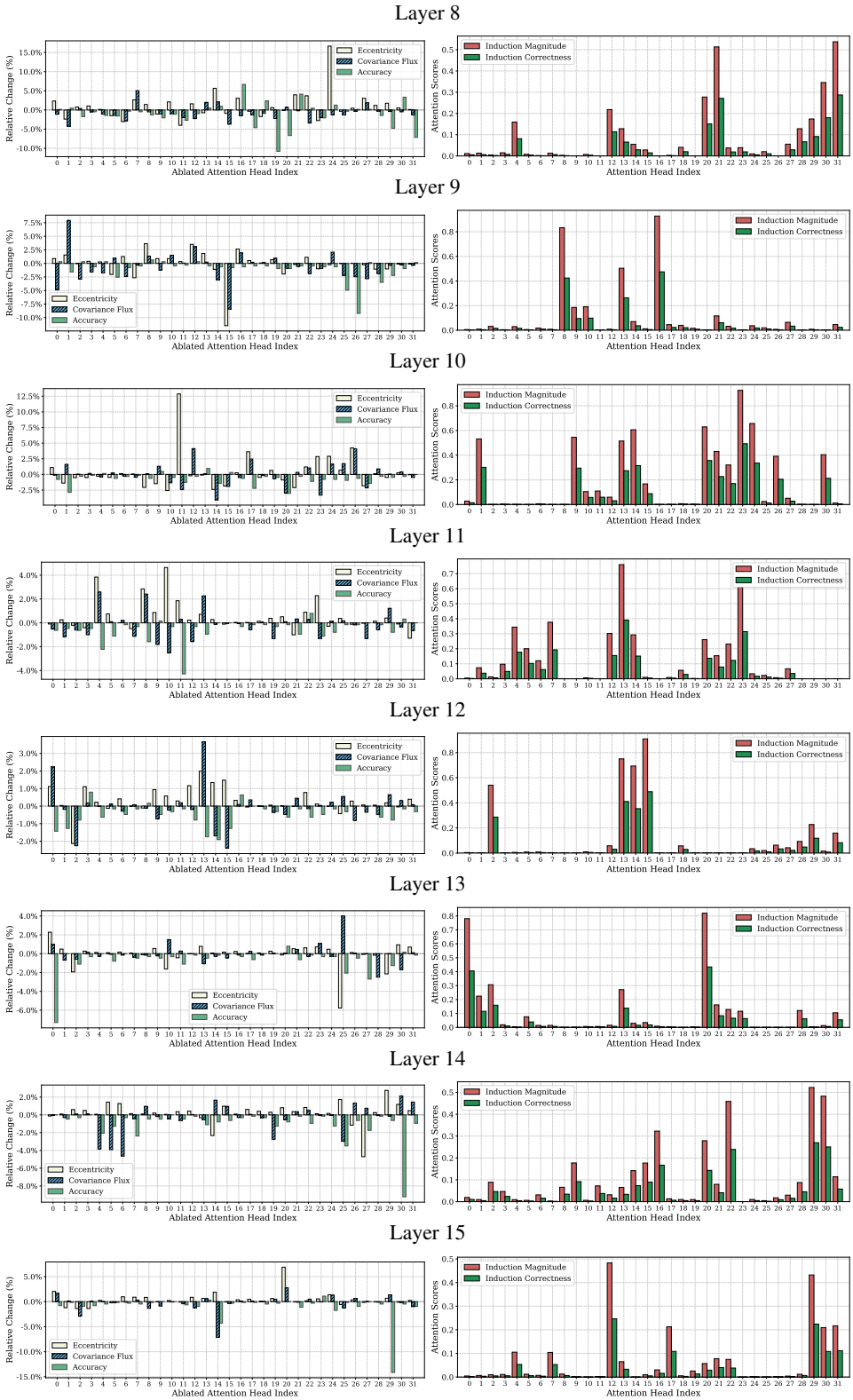


Figure B.51: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3.2-1B, Subjective.

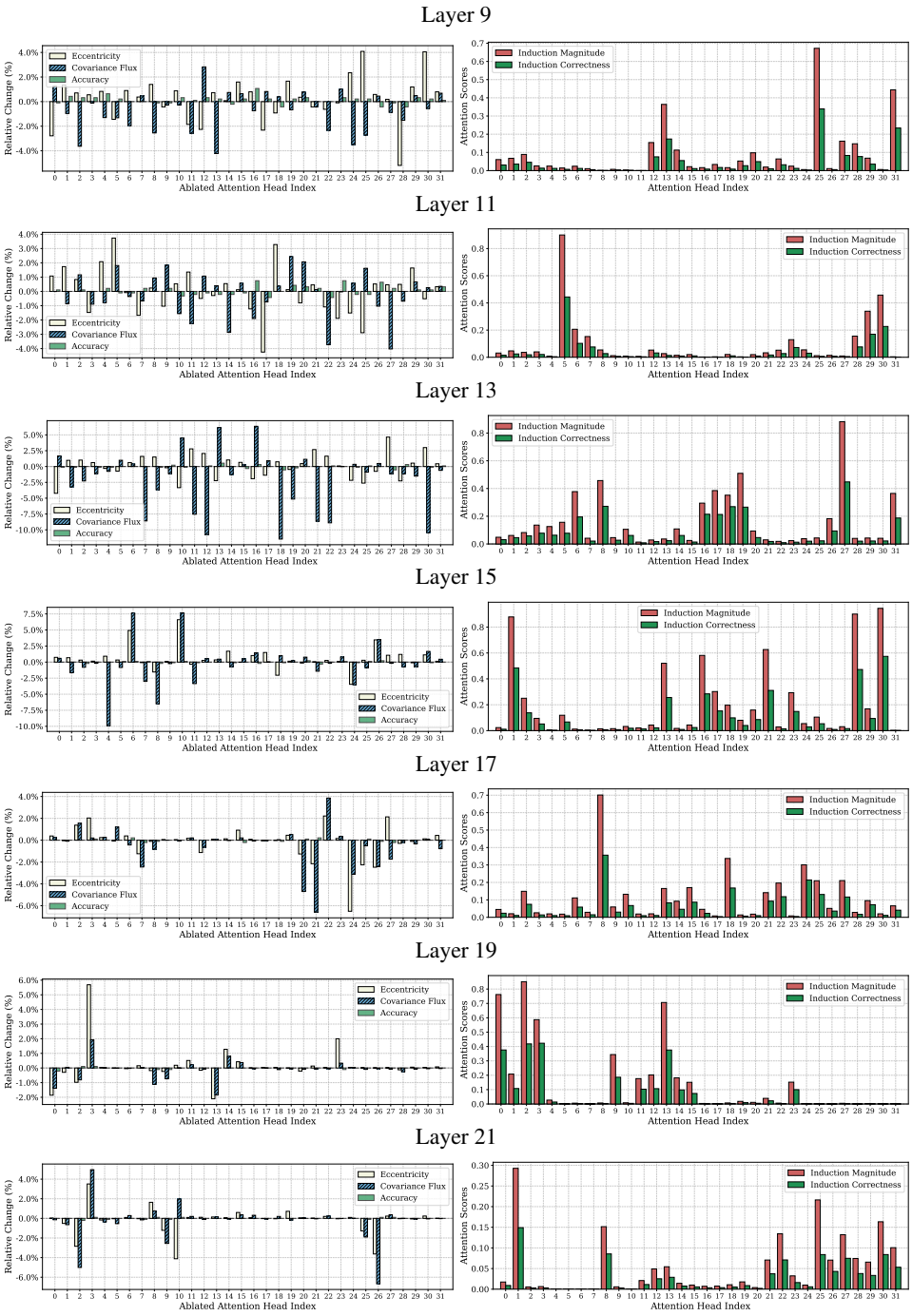


Figure B.52: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, SST-2.

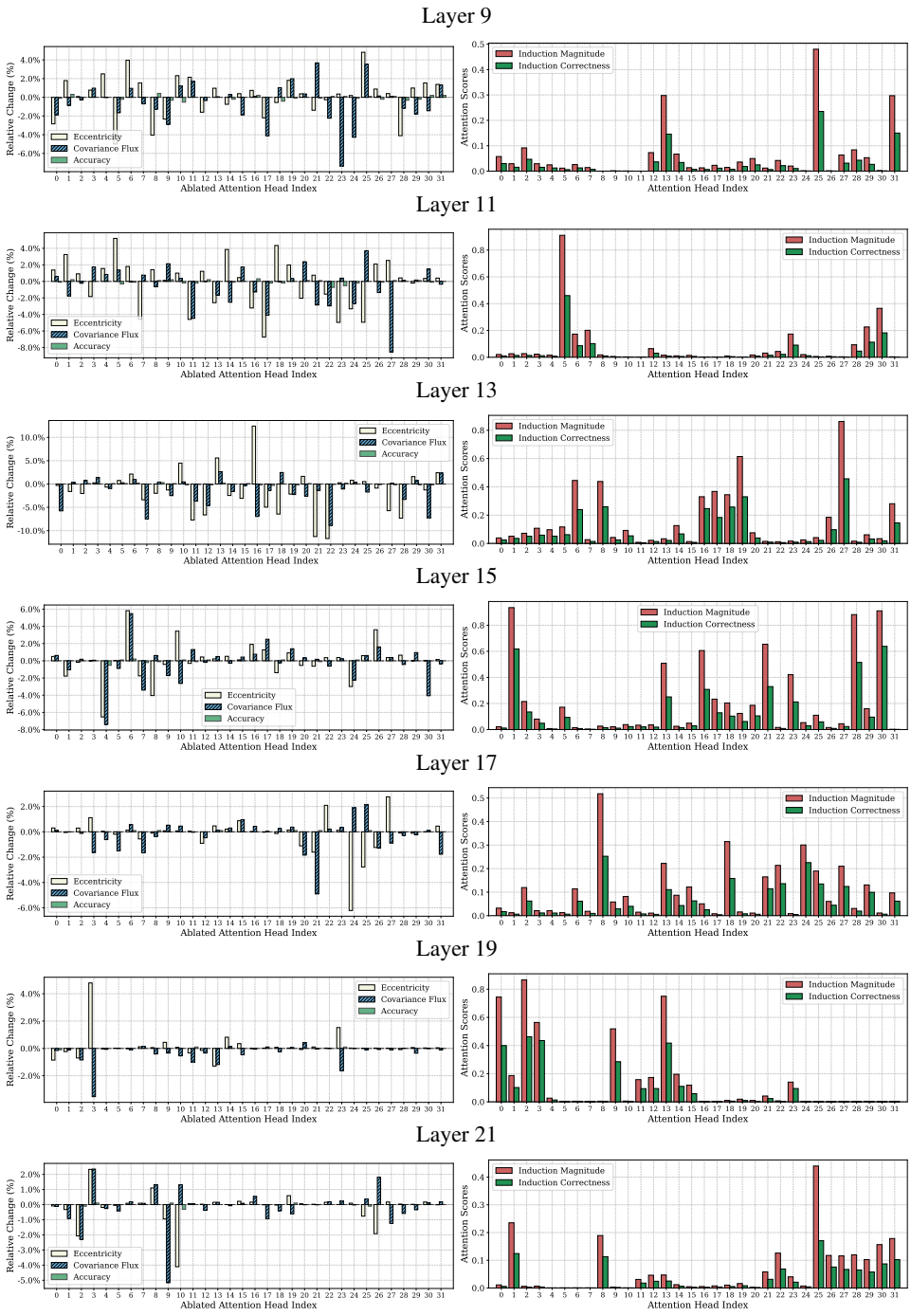


Figure B.53: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, MR.

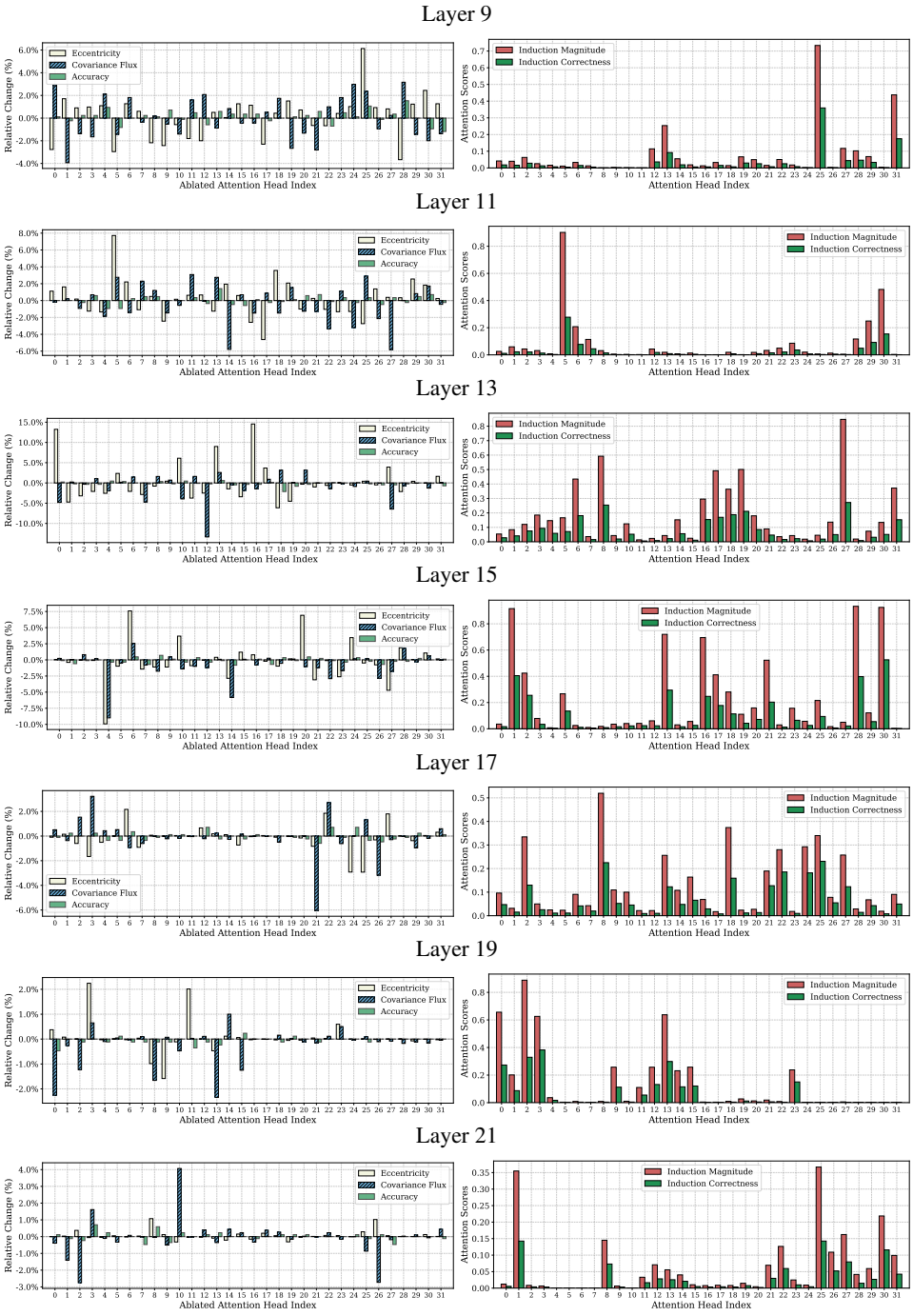


Figure B.54: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, FP.

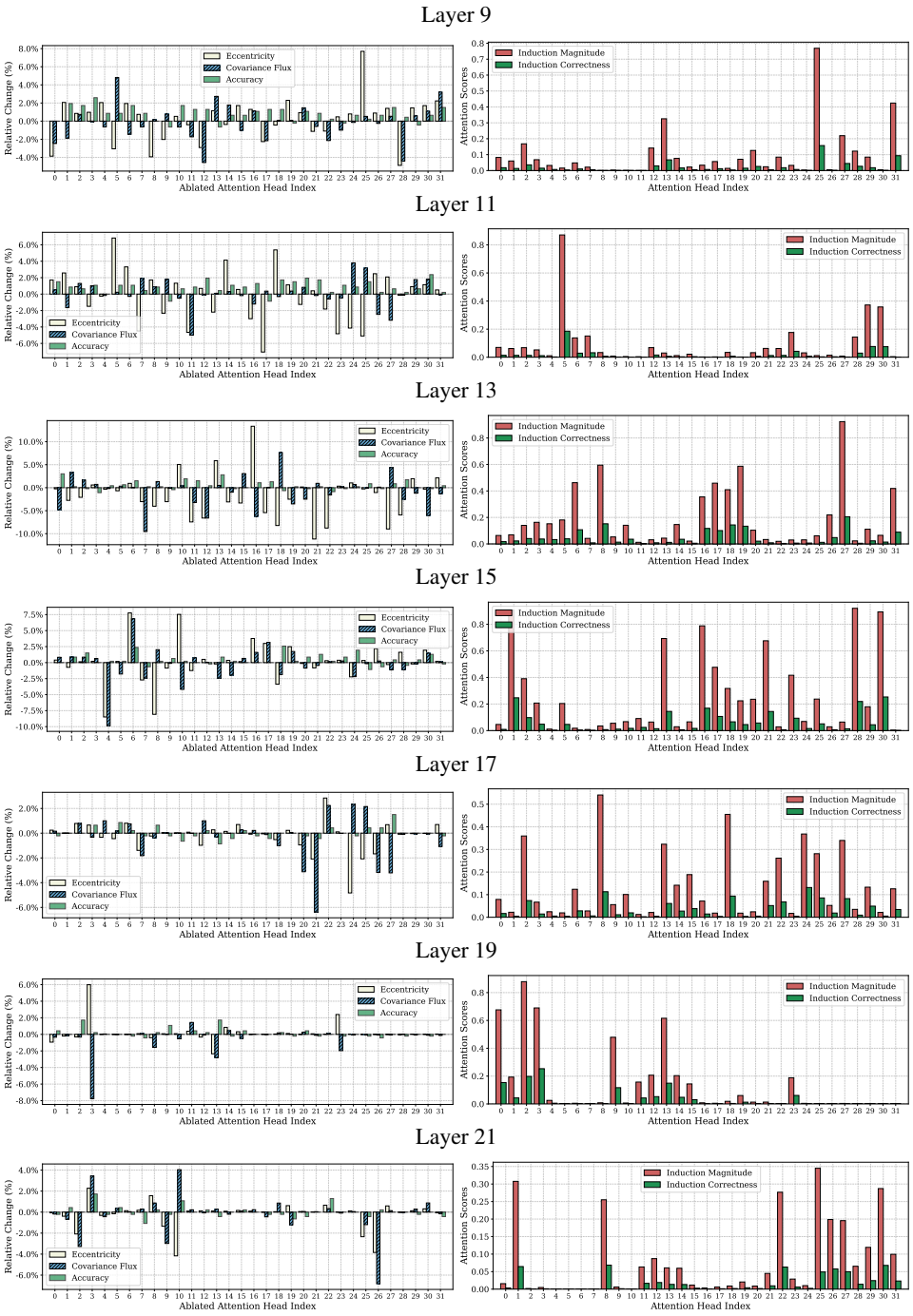


Figure B.55: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, SST-5.

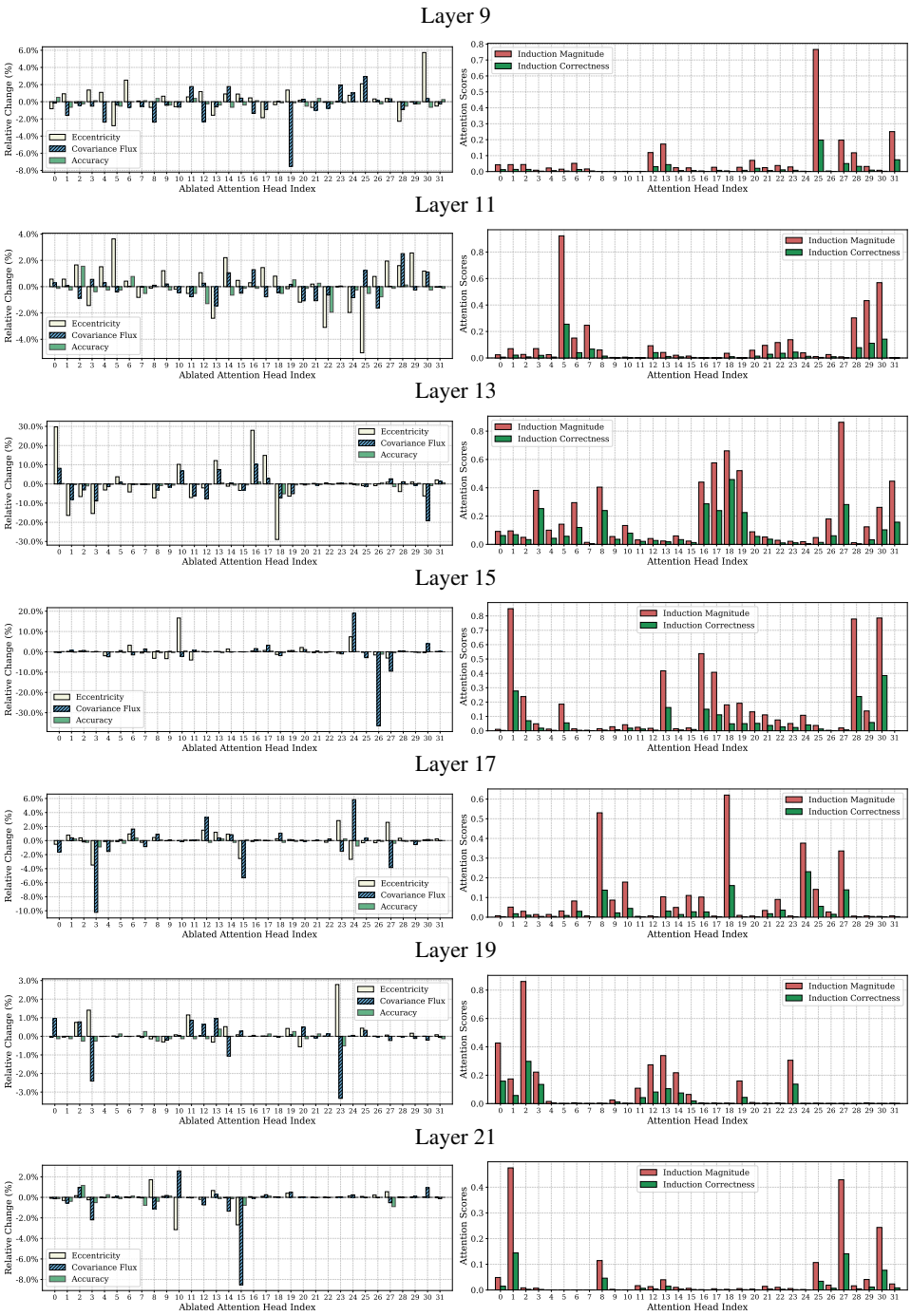


Figure B.56: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, AGNews.

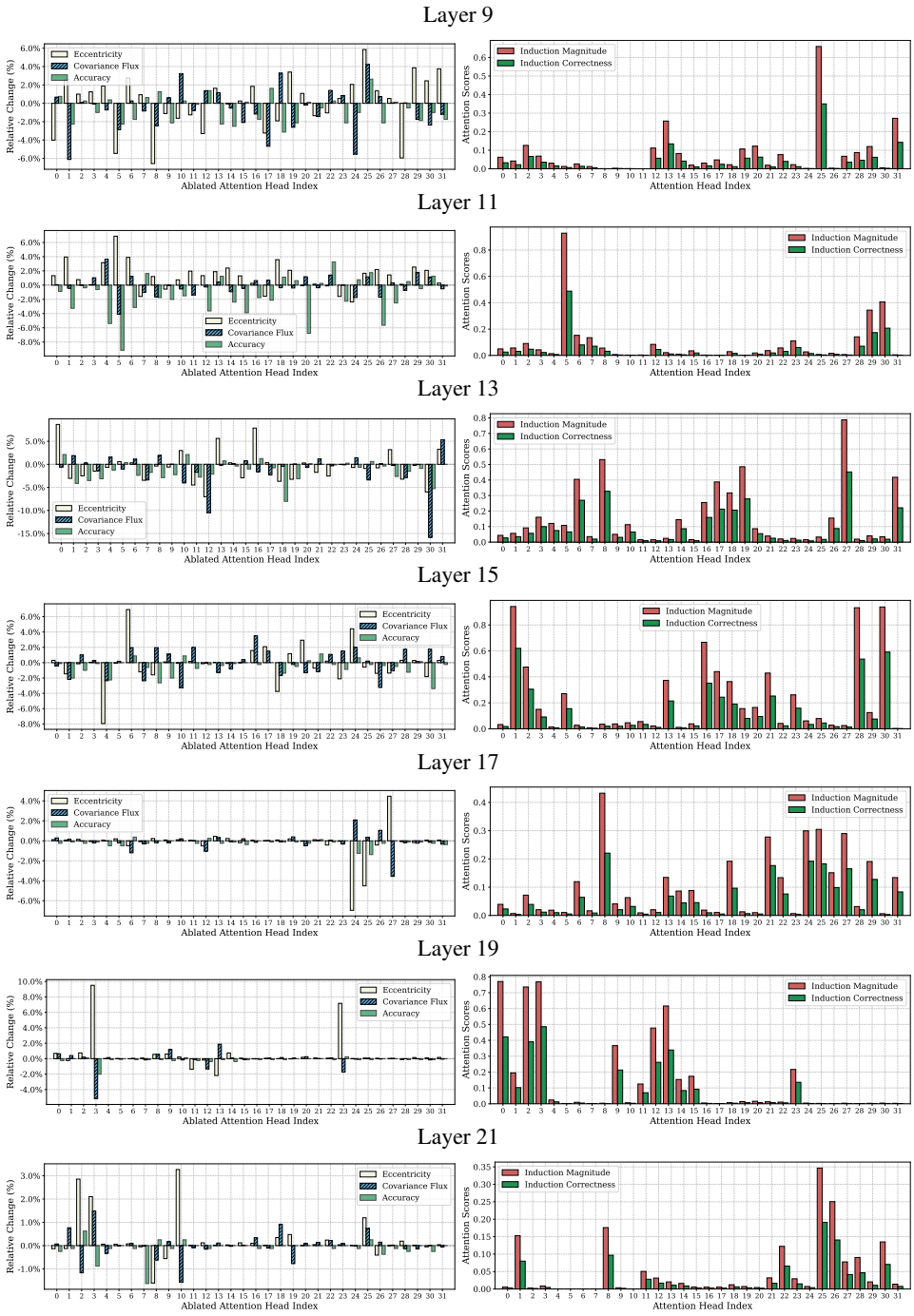
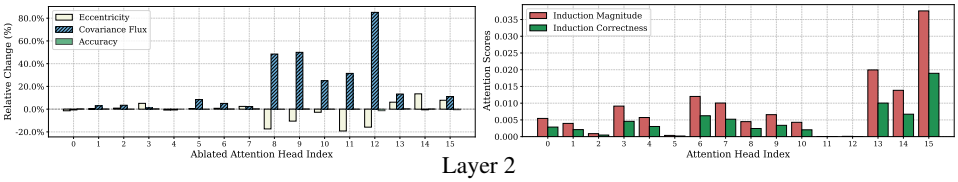
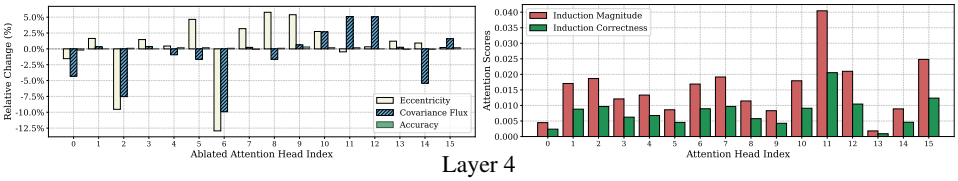


Figure B.57: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Llama 3-8B, Subjective.

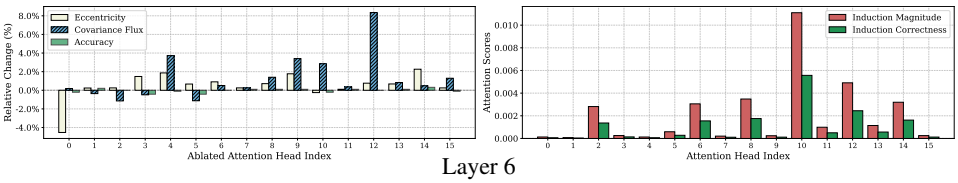
Layer 0



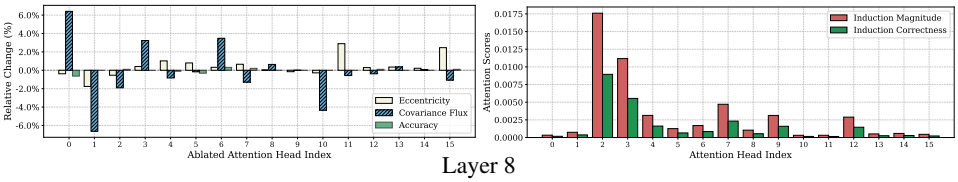
Layer 2



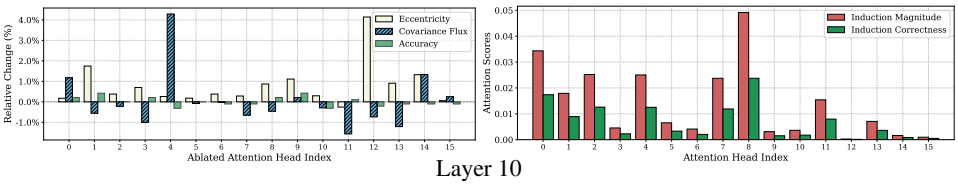
Layer 4



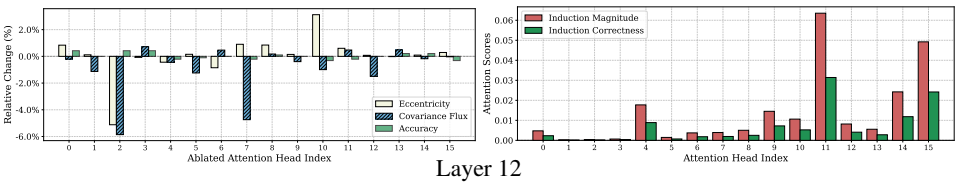
Layer 6



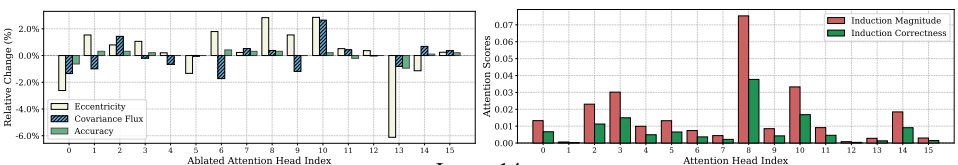
Layer 8



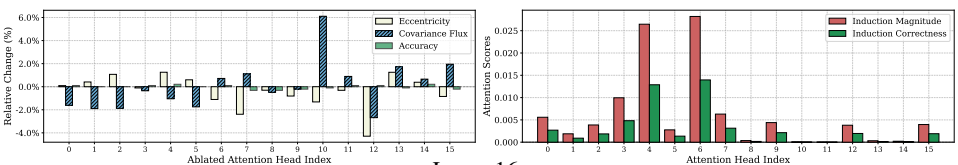
Layer 10



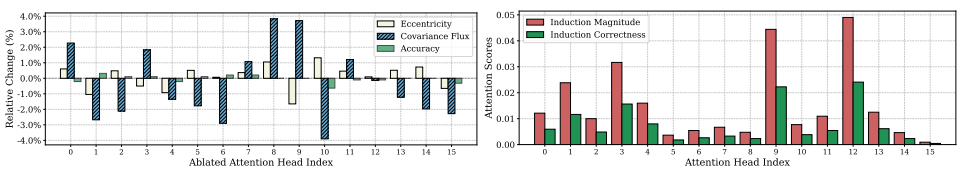
Layer 12



Layer 14



Layer 16



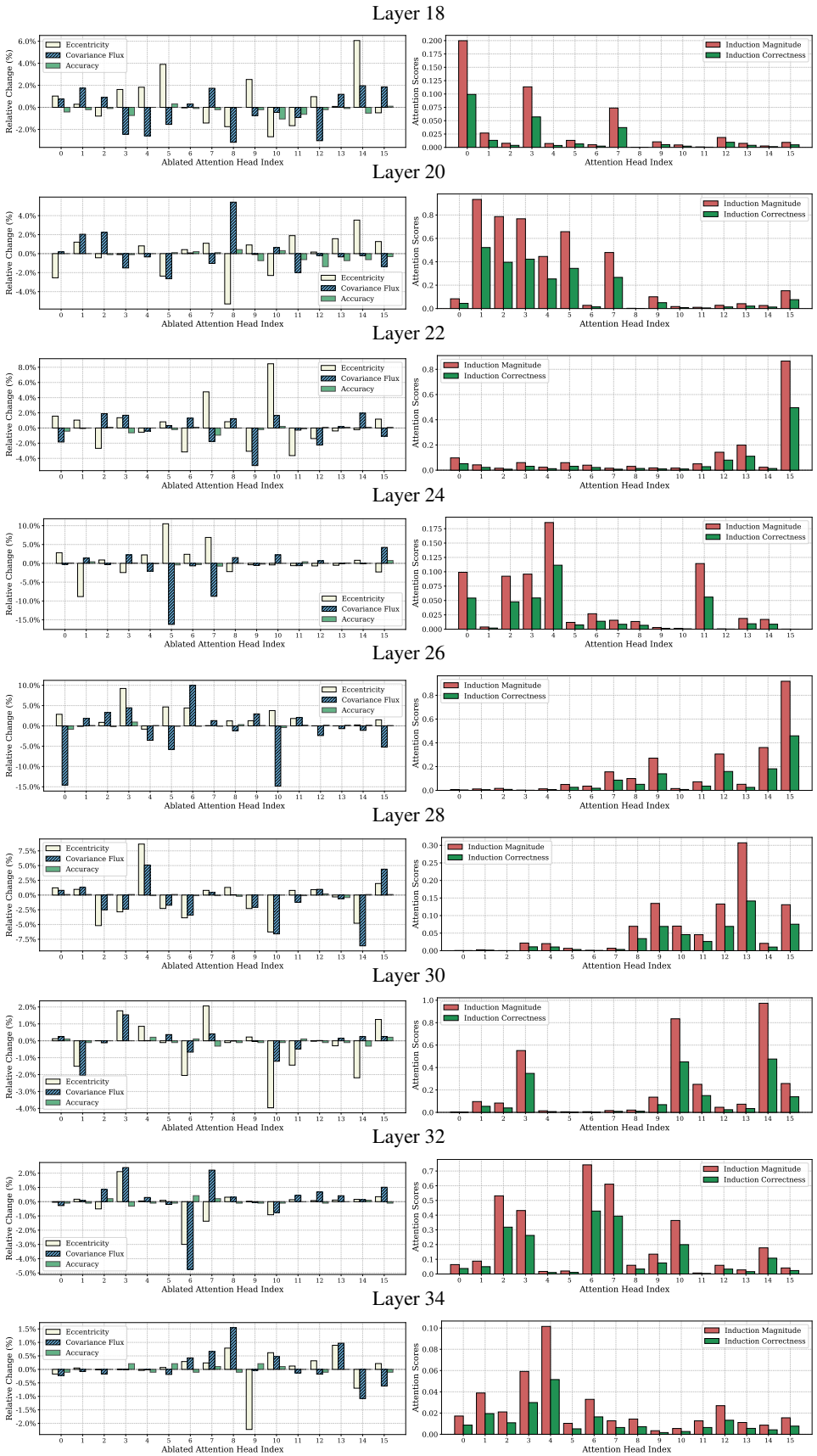
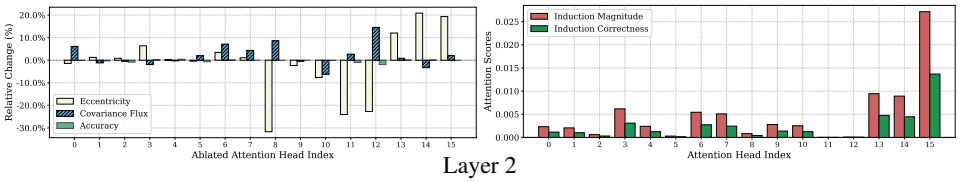
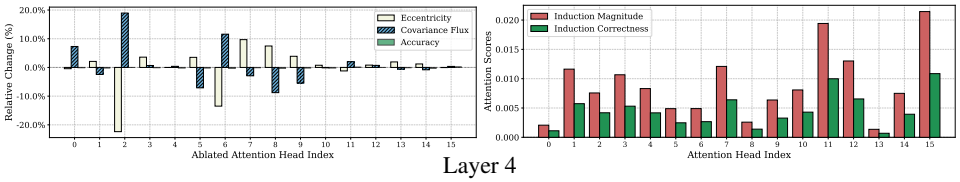


Figure B.58: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, SST-2.

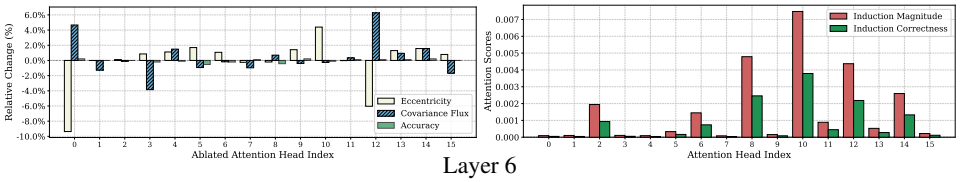
Layer 0



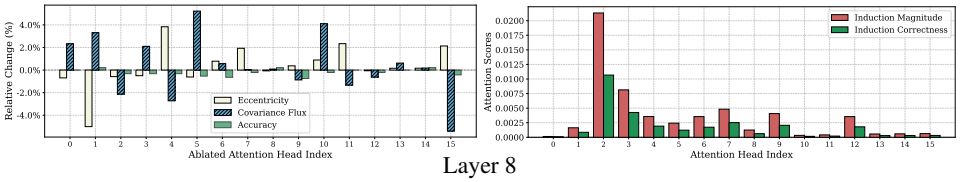
Layer 2



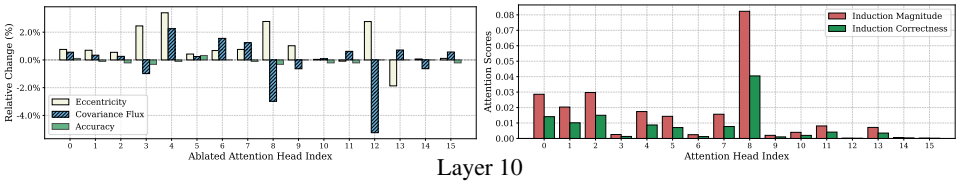
Layer 4



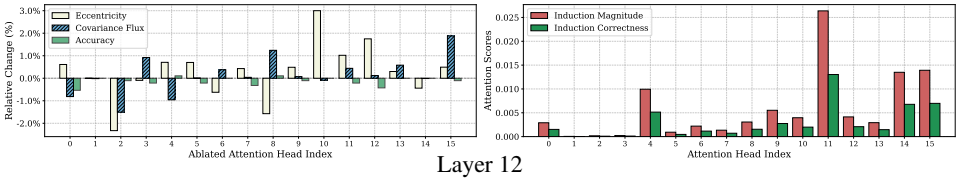
Layer 6



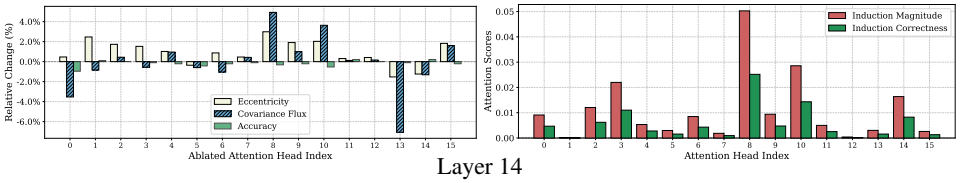
Layer 8



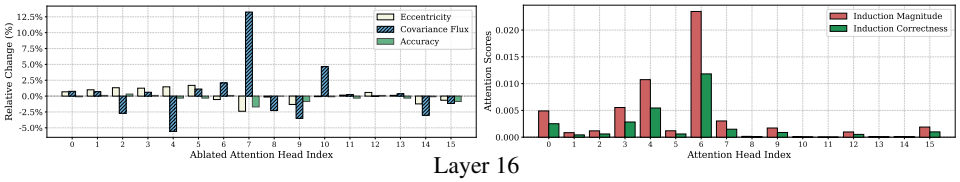
Layer 10



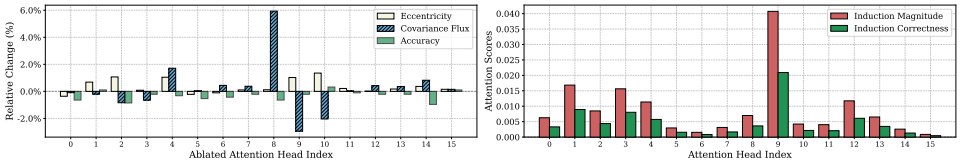
Layer 12



Layer 14



Layer 16



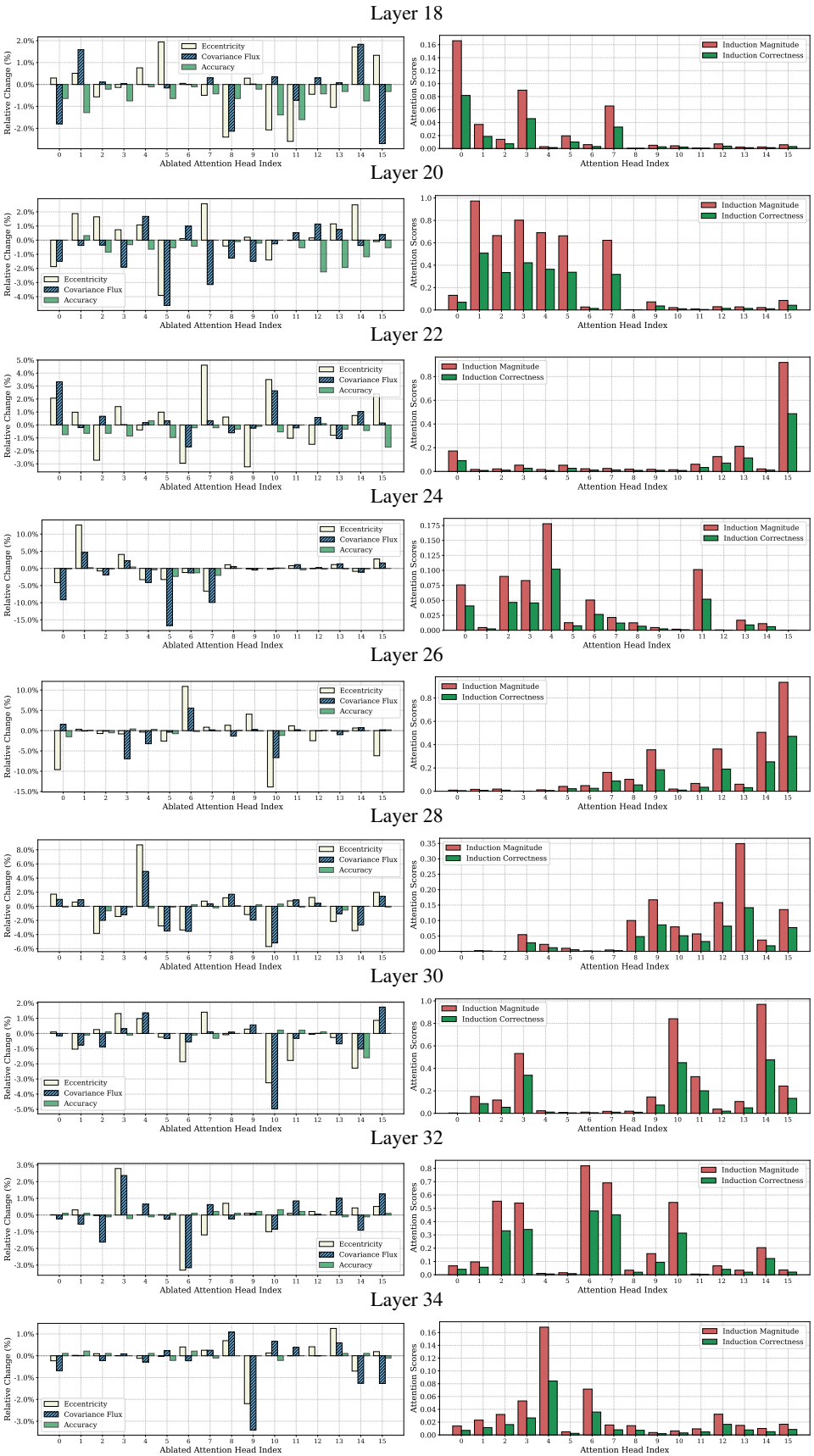
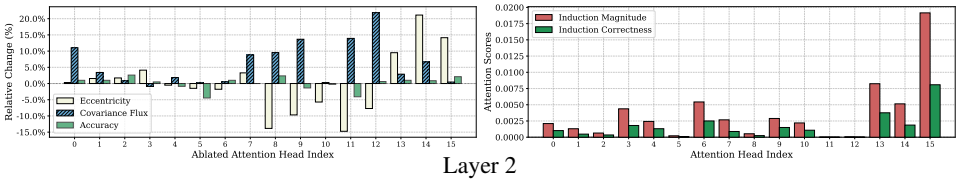
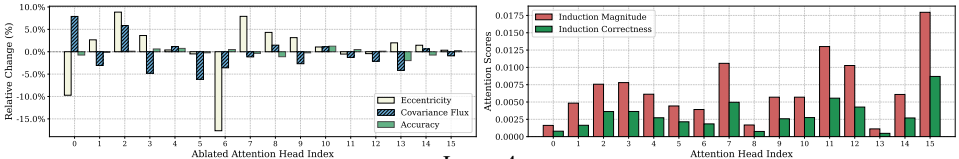


Figure B.59: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, MR.

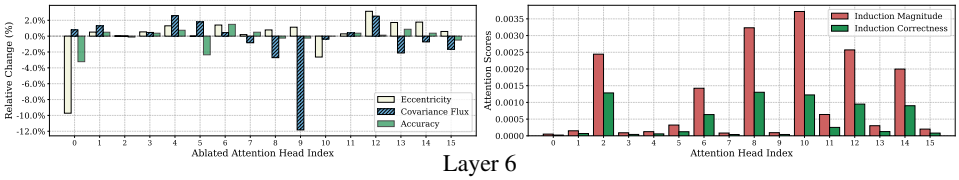
Layer 0



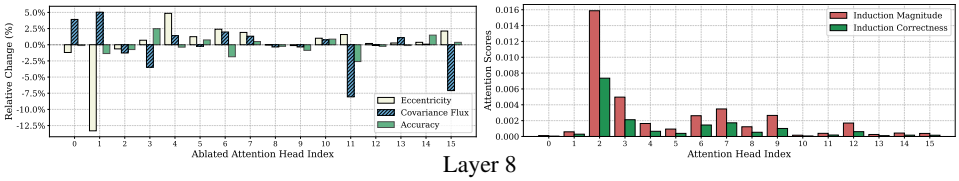
Layer 2



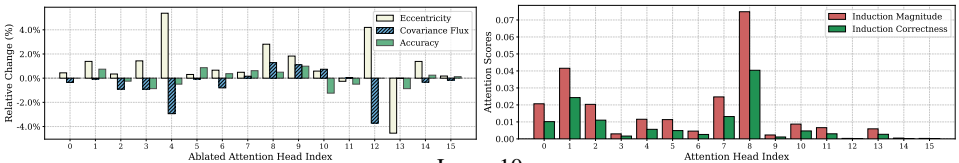
Layer 4



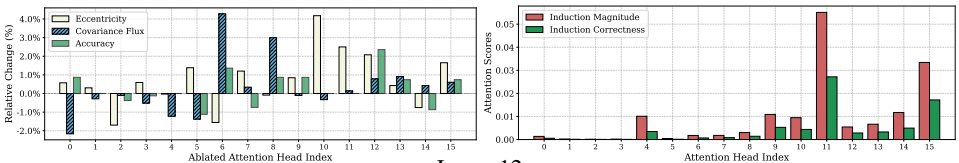
Layer 6



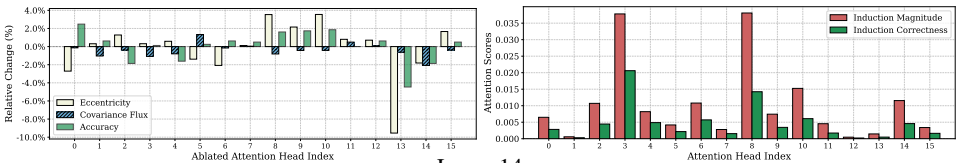
Layer 8



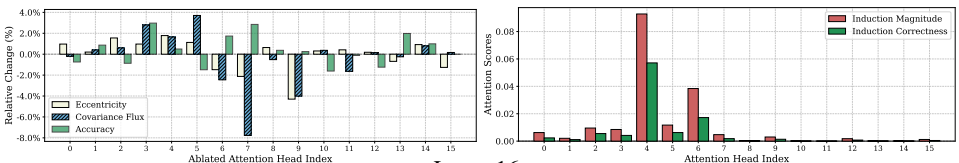
Layer 10



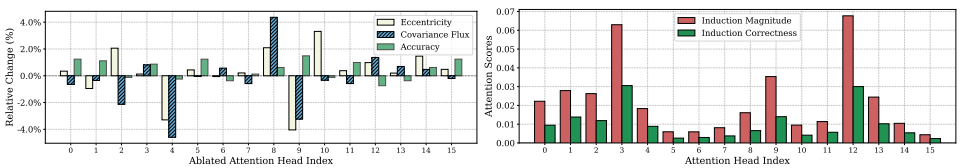
Layer 12



Layer 14



Layer 16



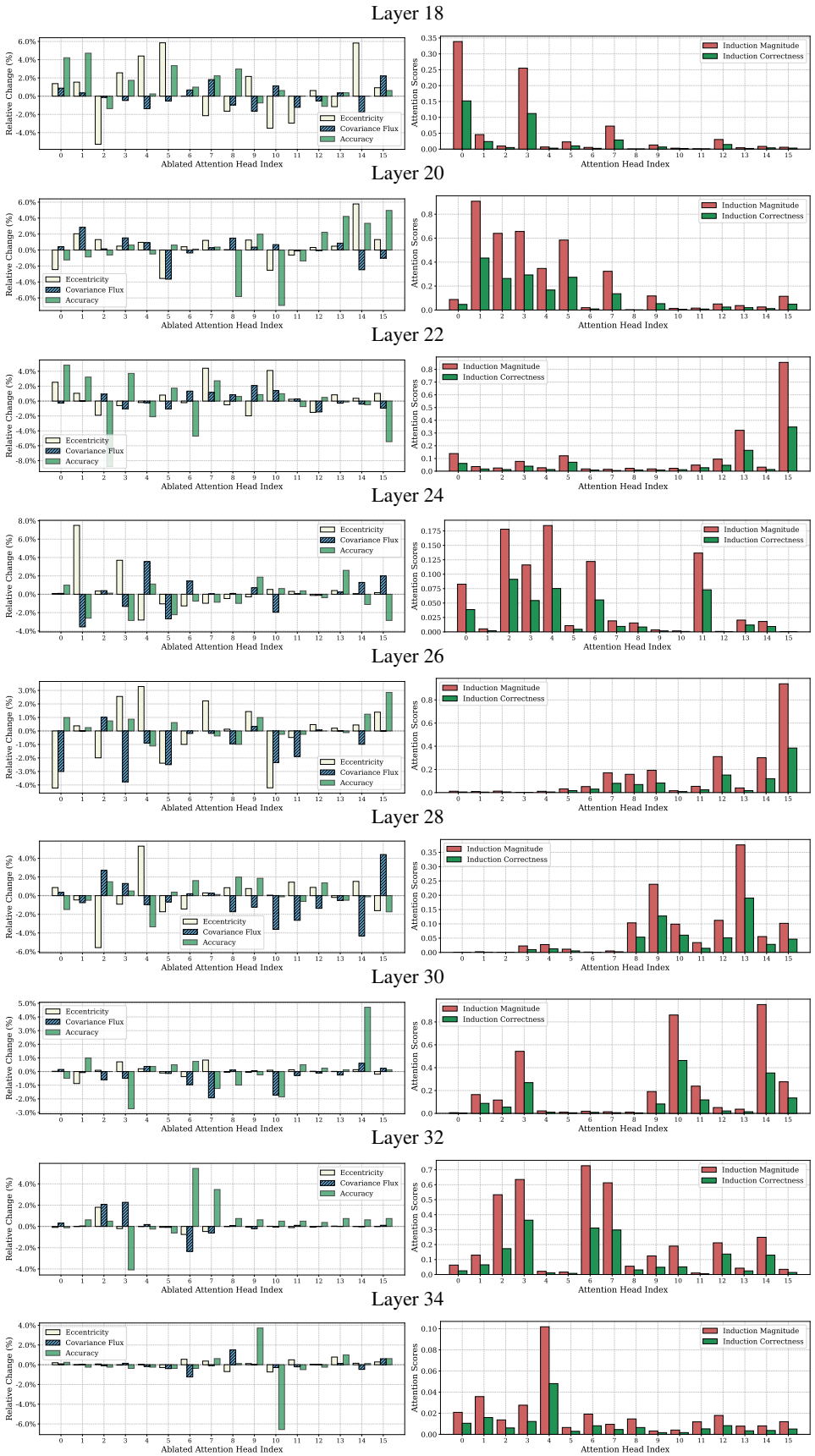
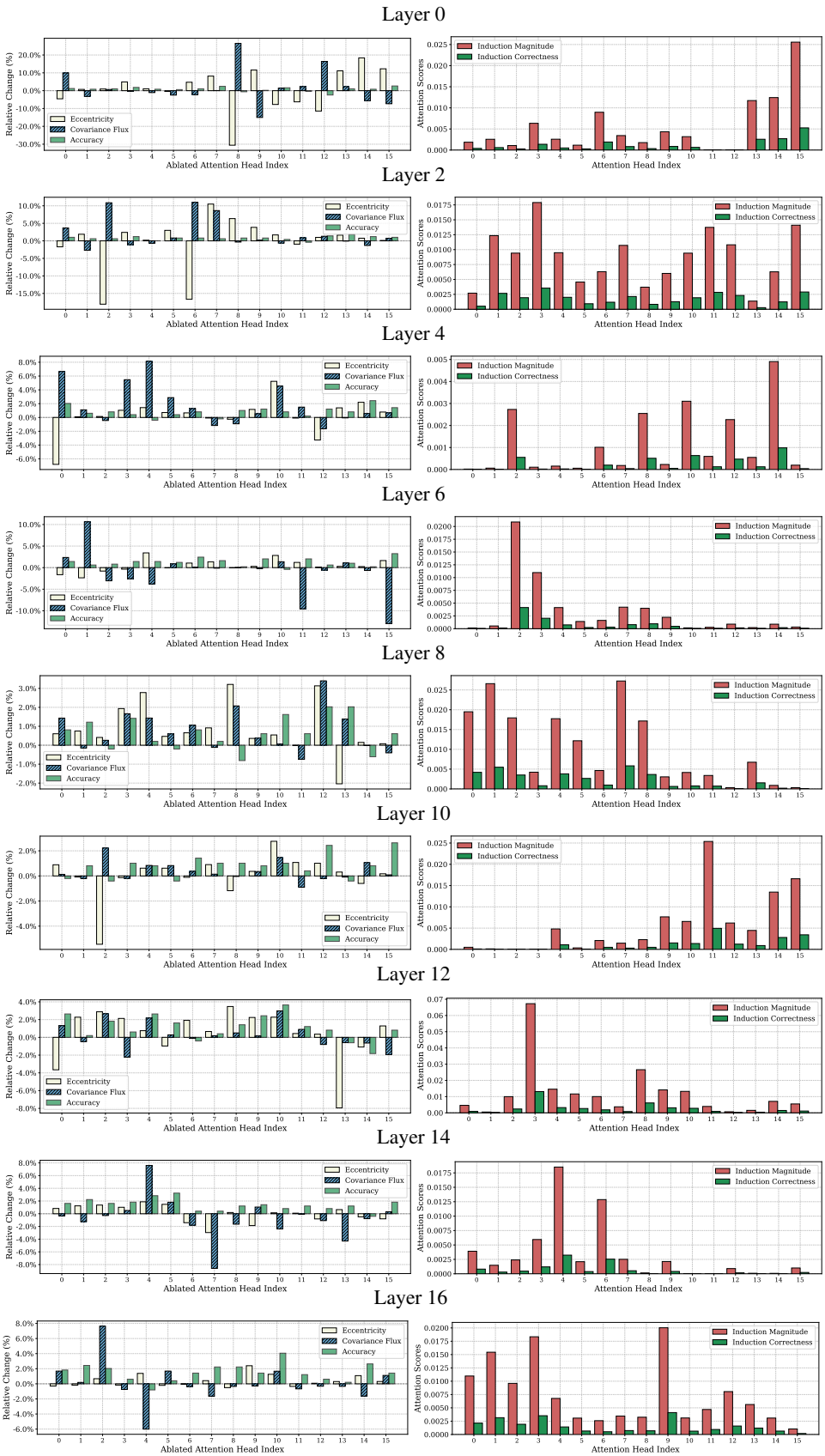


Figure B.60: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, FP.


Layer 0
Layer 2
Layer 4
Layer 6
Layer 8
Layer 10
Layer 12
Layer 14
Layer 16
Relative Change (%)
Ablated Attention Head Index
Eccentricity
Covariance Flux
Accuracy
Attention Scores
Attention Head Index
Induction Magnitude
Induction Correctness

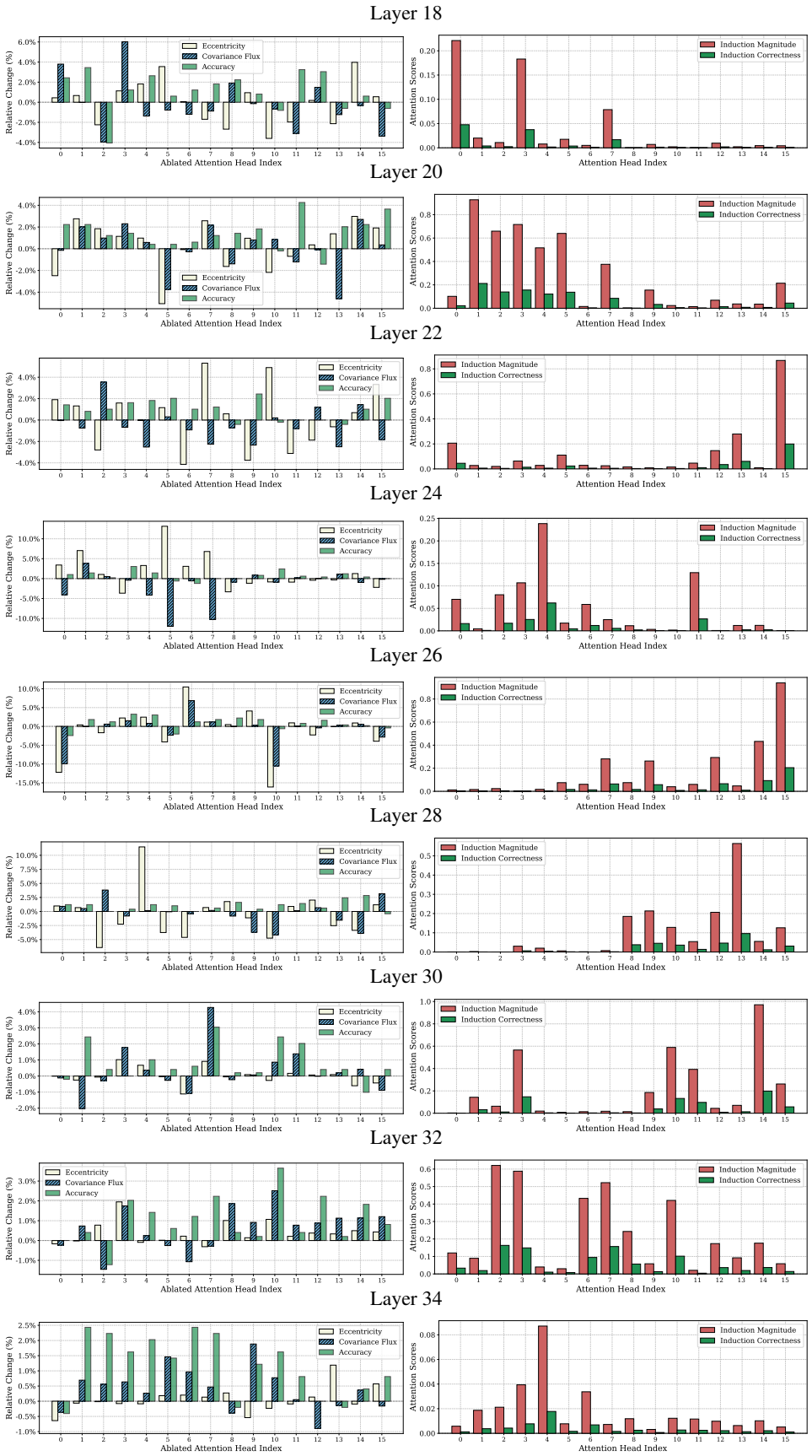
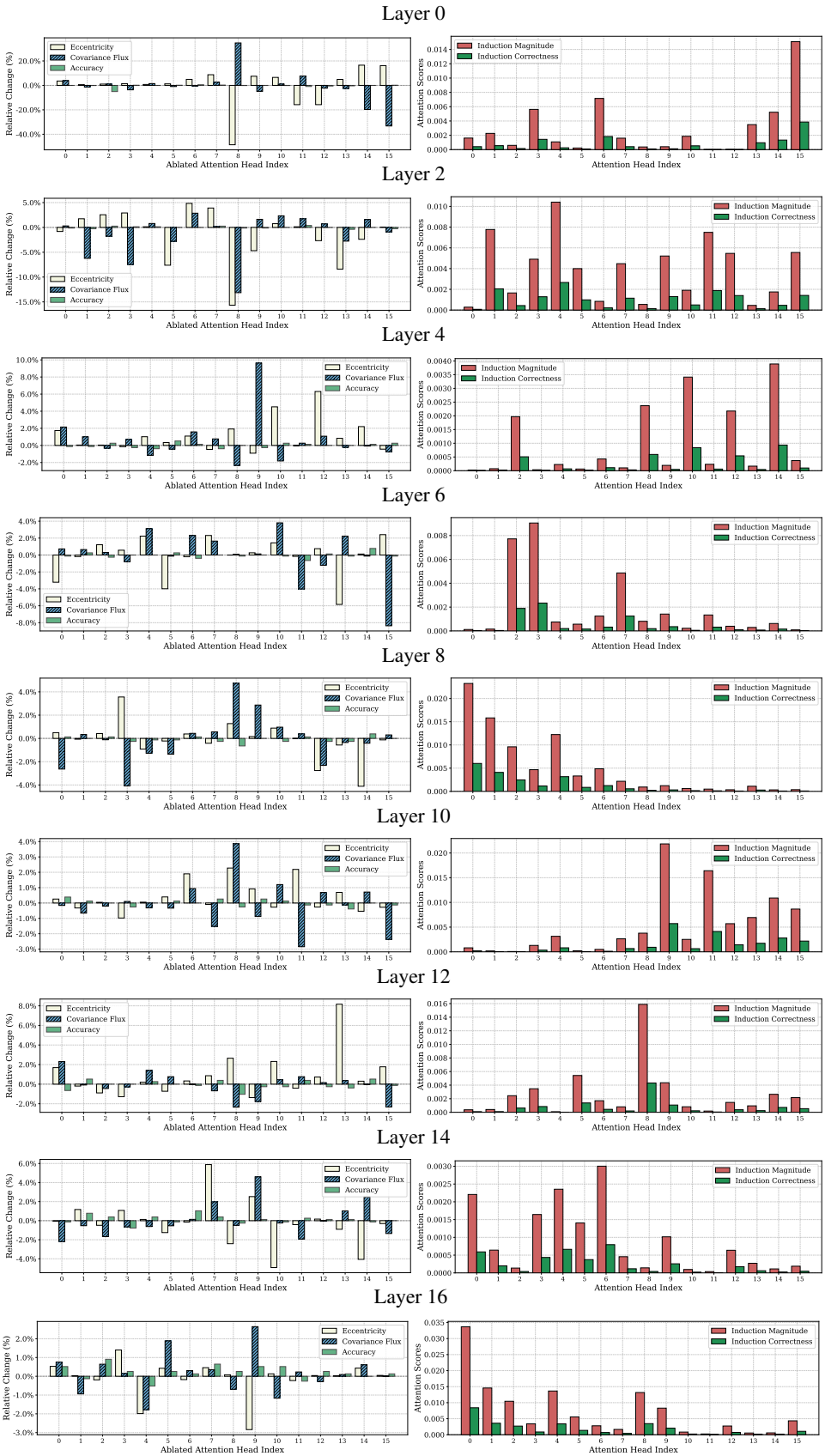


Figure B.61: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, SST-5.


Layer 0
Layer 2
Layer 4
Layer 6
Layer 8
Layer 10
Layer 12
Layer 14
Layer 16
Eccentricity
Covariance Flux
Accuracy
Relative Change (%)
Ablated Attention Head Index
Induction Magnitude
Induction Correctness
Attention Scores
Attention Head Index

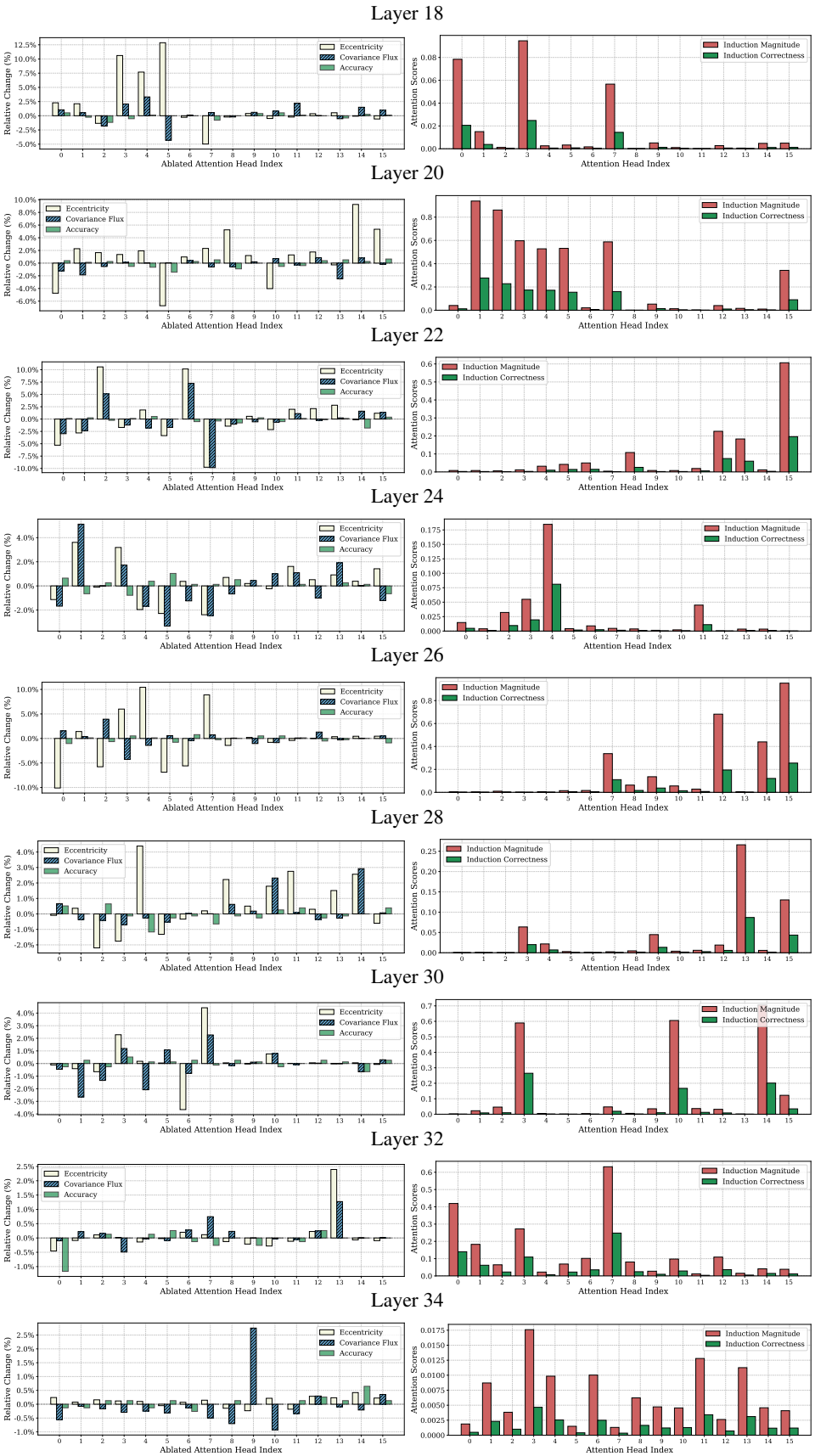
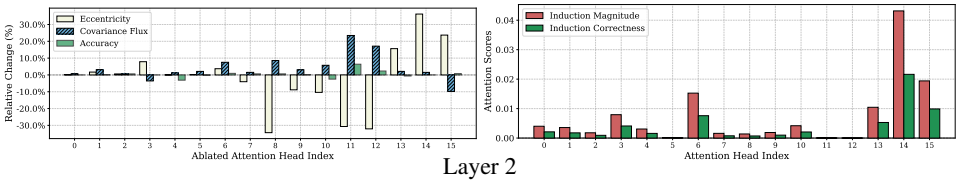
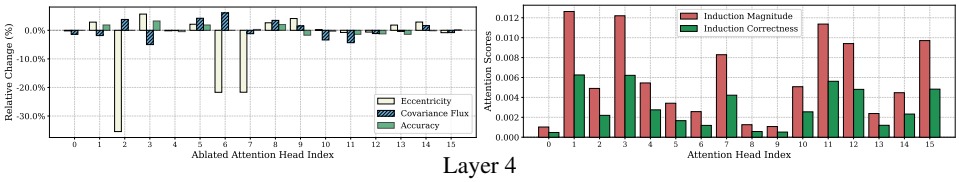


Figure B.62: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, AGNews.

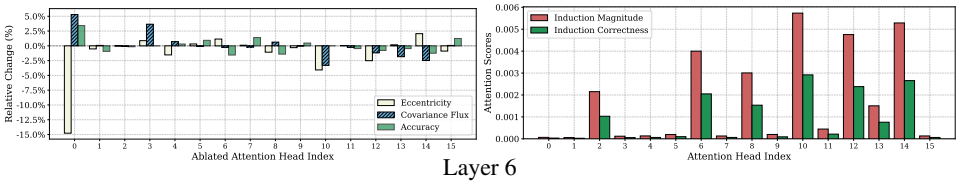
Layer 0



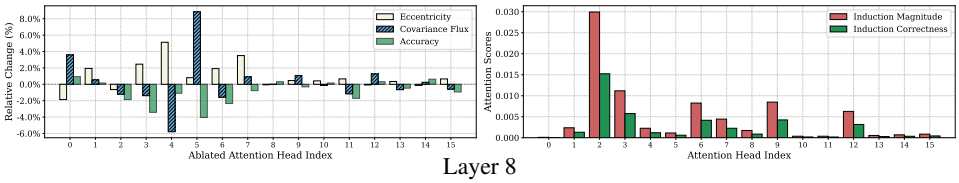
Layer 2



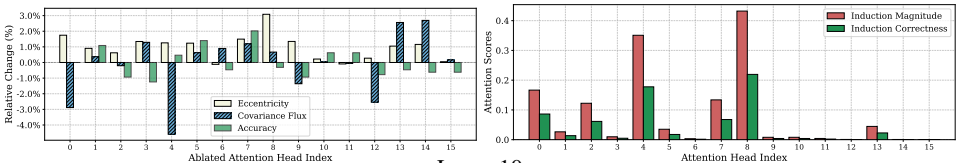
Layer 4



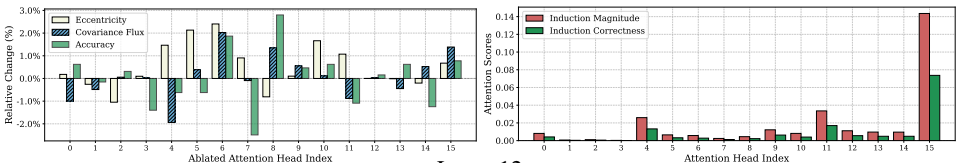
Layer 6



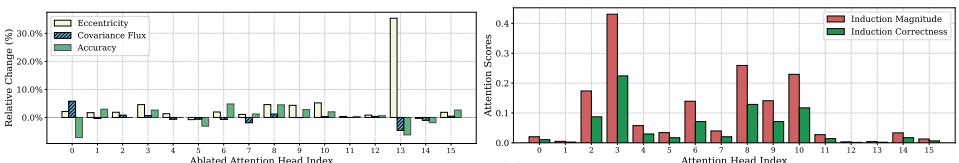
Layer 8



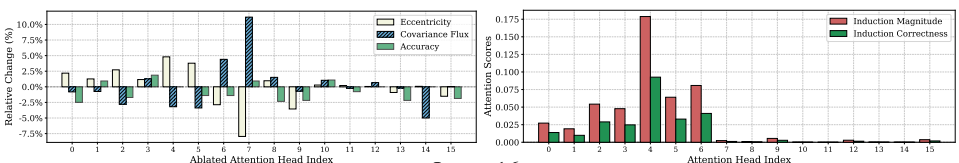
Layer 10



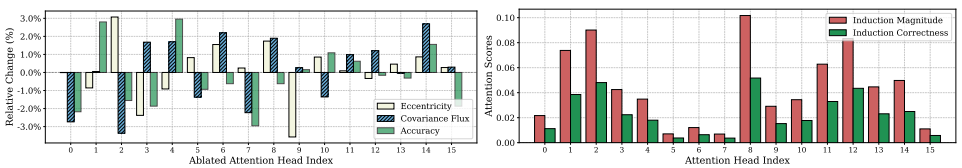
Layer 12



Layer 14



Layer 16



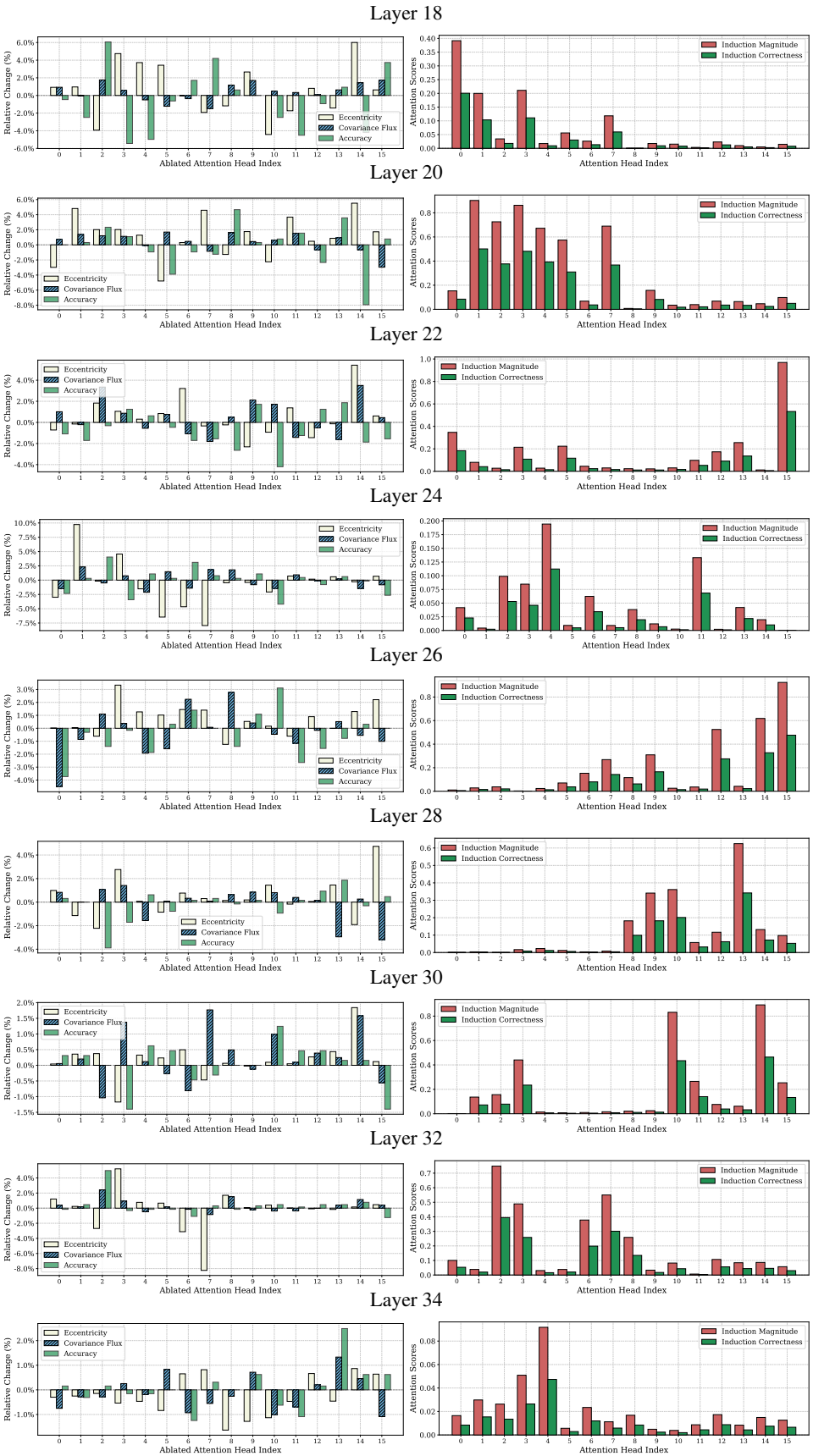
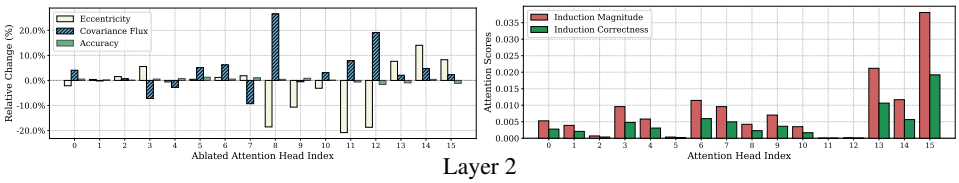
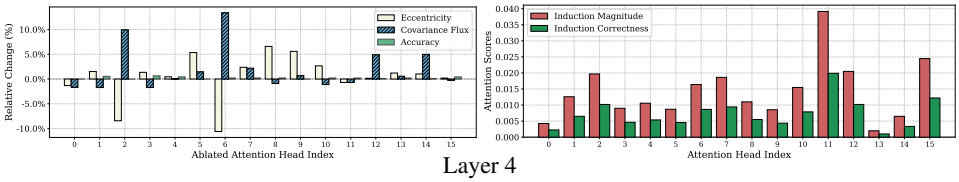


Figure B.63: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B, Subjective.

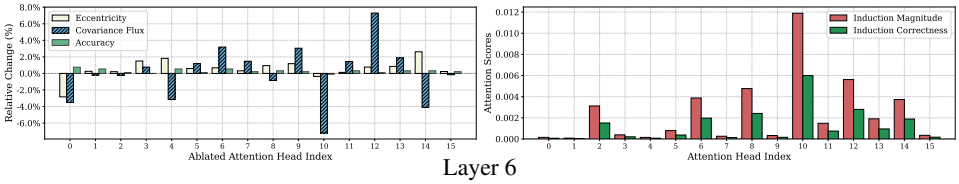
Layer 0



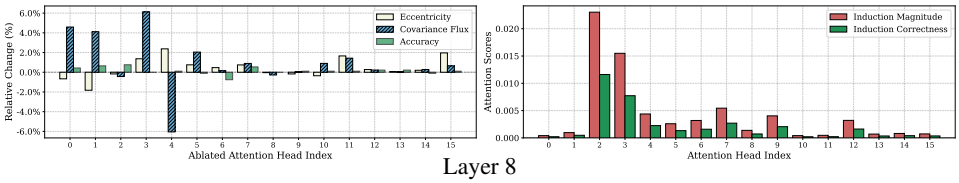
Layer 2



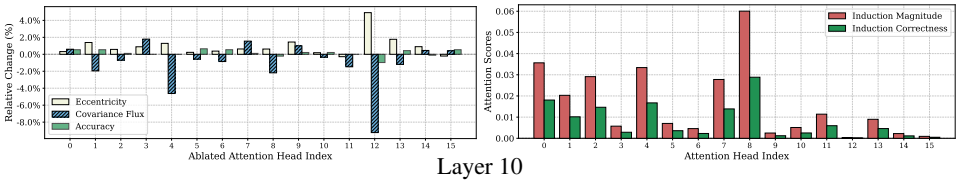
Layer 4



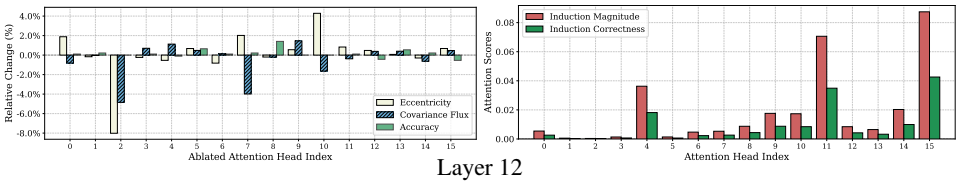
Layer 6



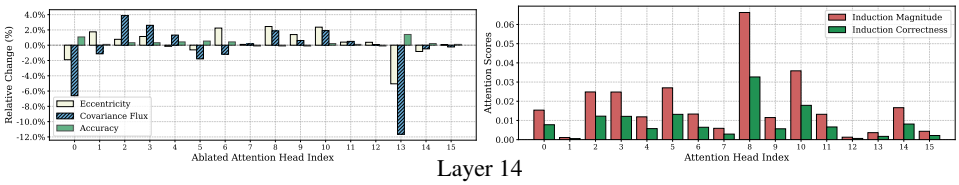
Layer 8



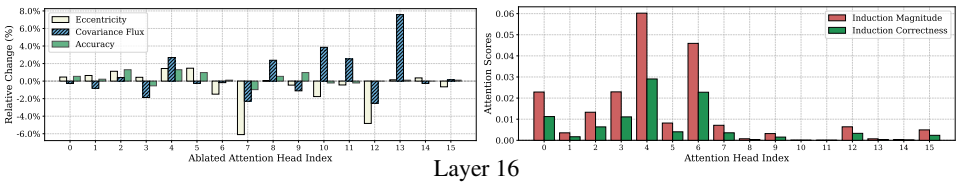
Layer 10



Layer 12



Layer 14



Layer 16

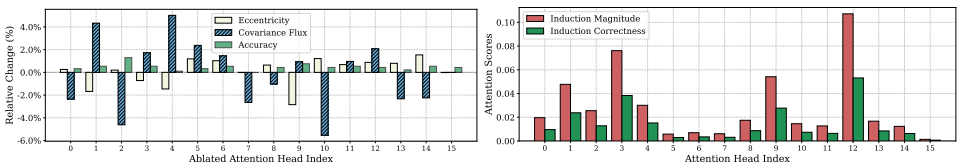
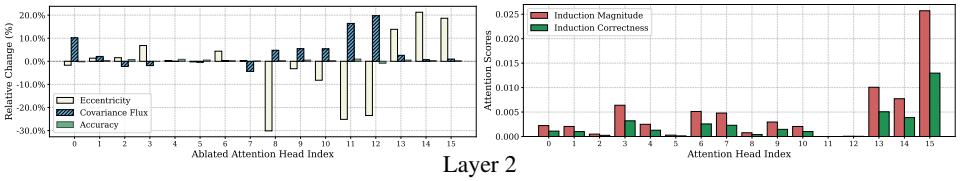


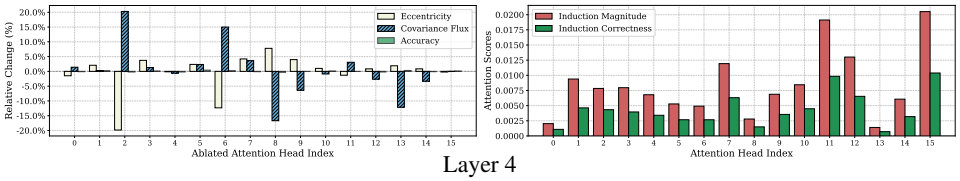


Figure B.64: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, SST-2.

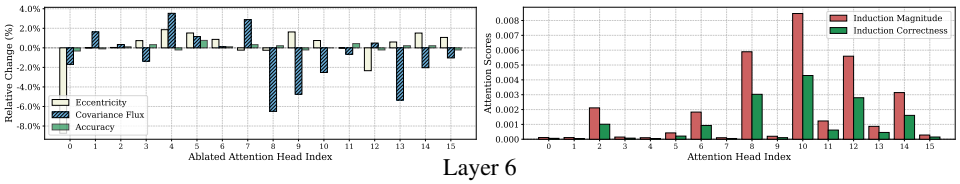
Layer 0



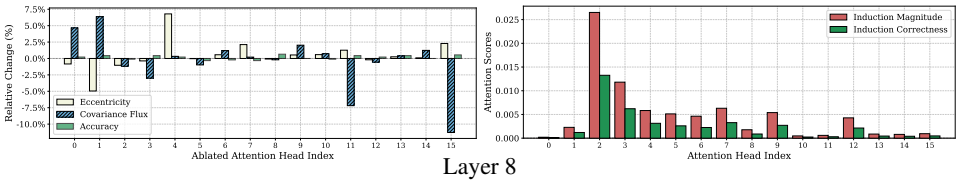
Layer 2



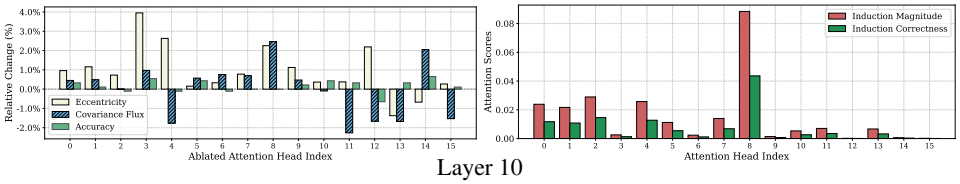
Layer 4



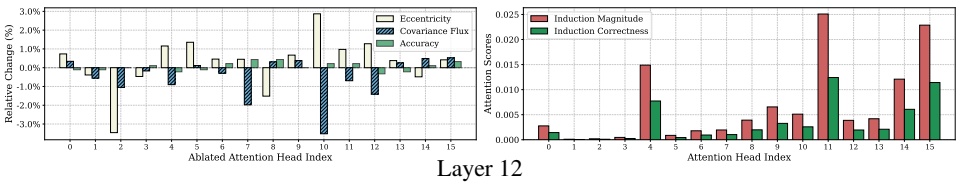
Layer 6



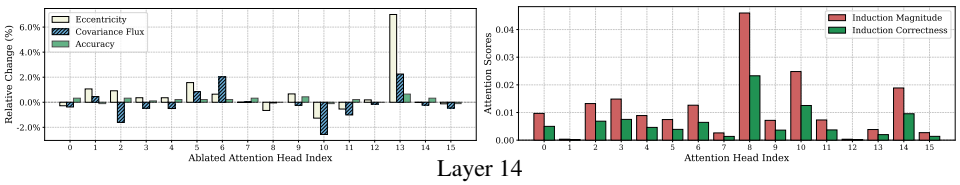
Layer 8



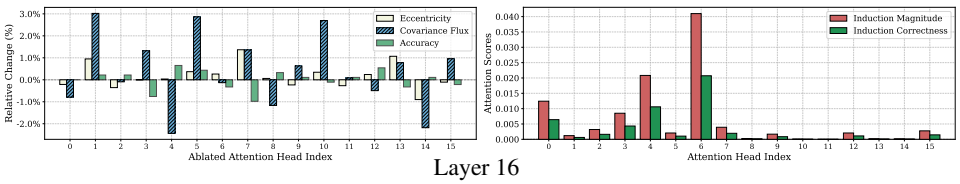
Layer 10



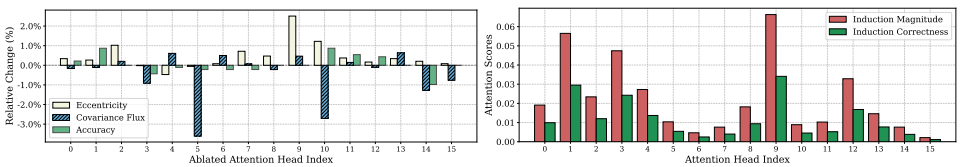
Layer 12



Layer 14



Layer 16



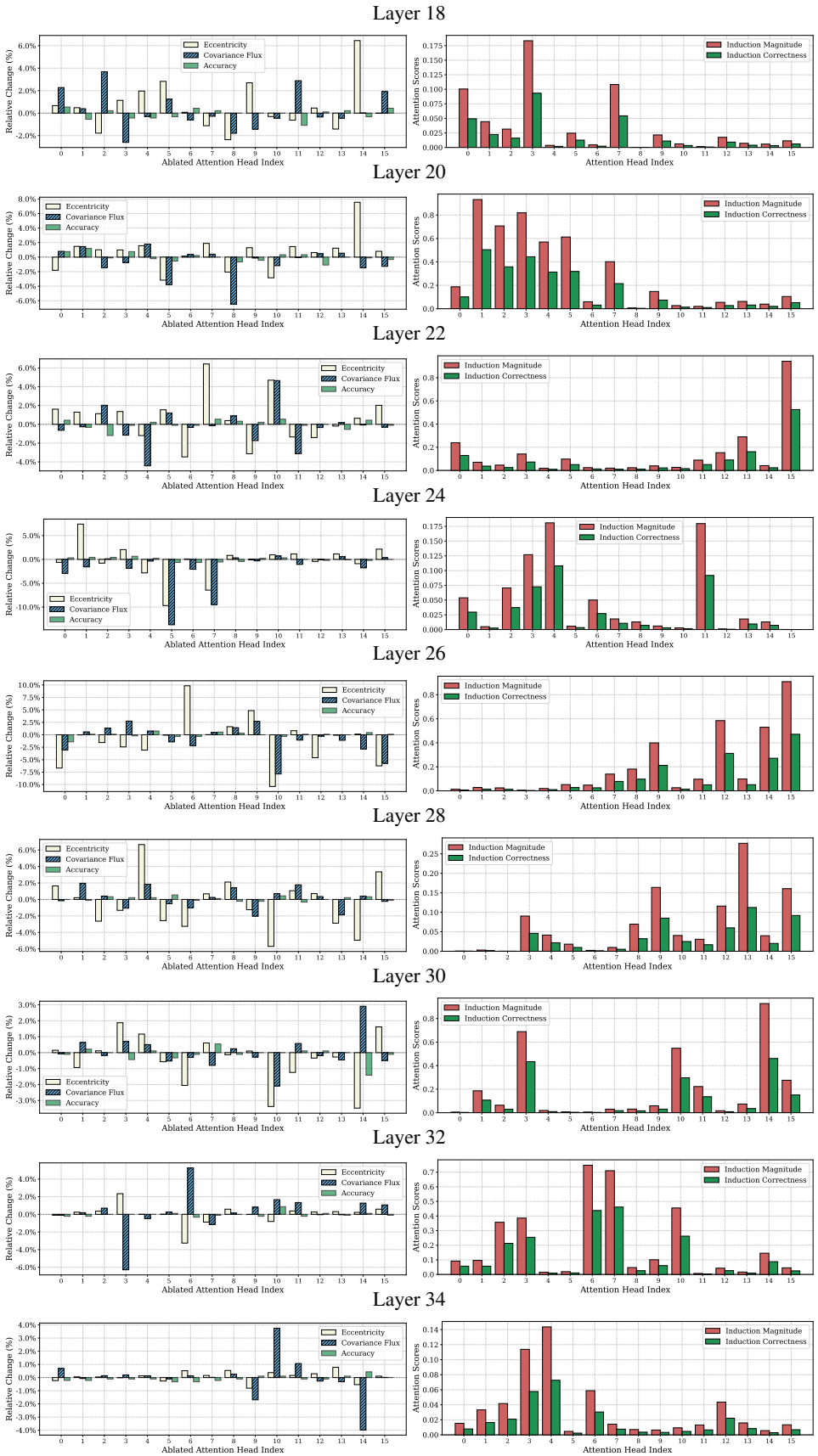


Figure B.65: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, MR.

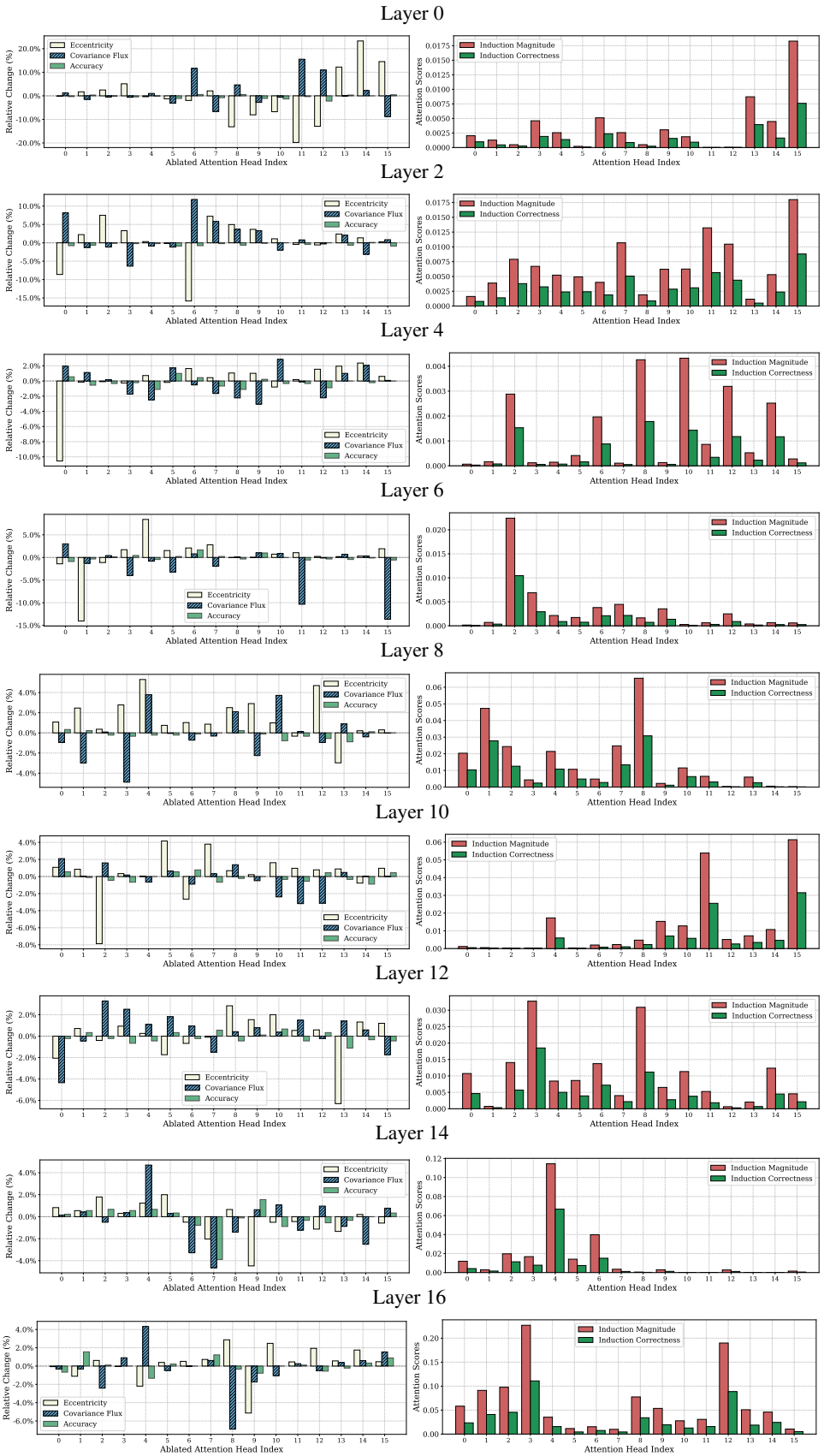
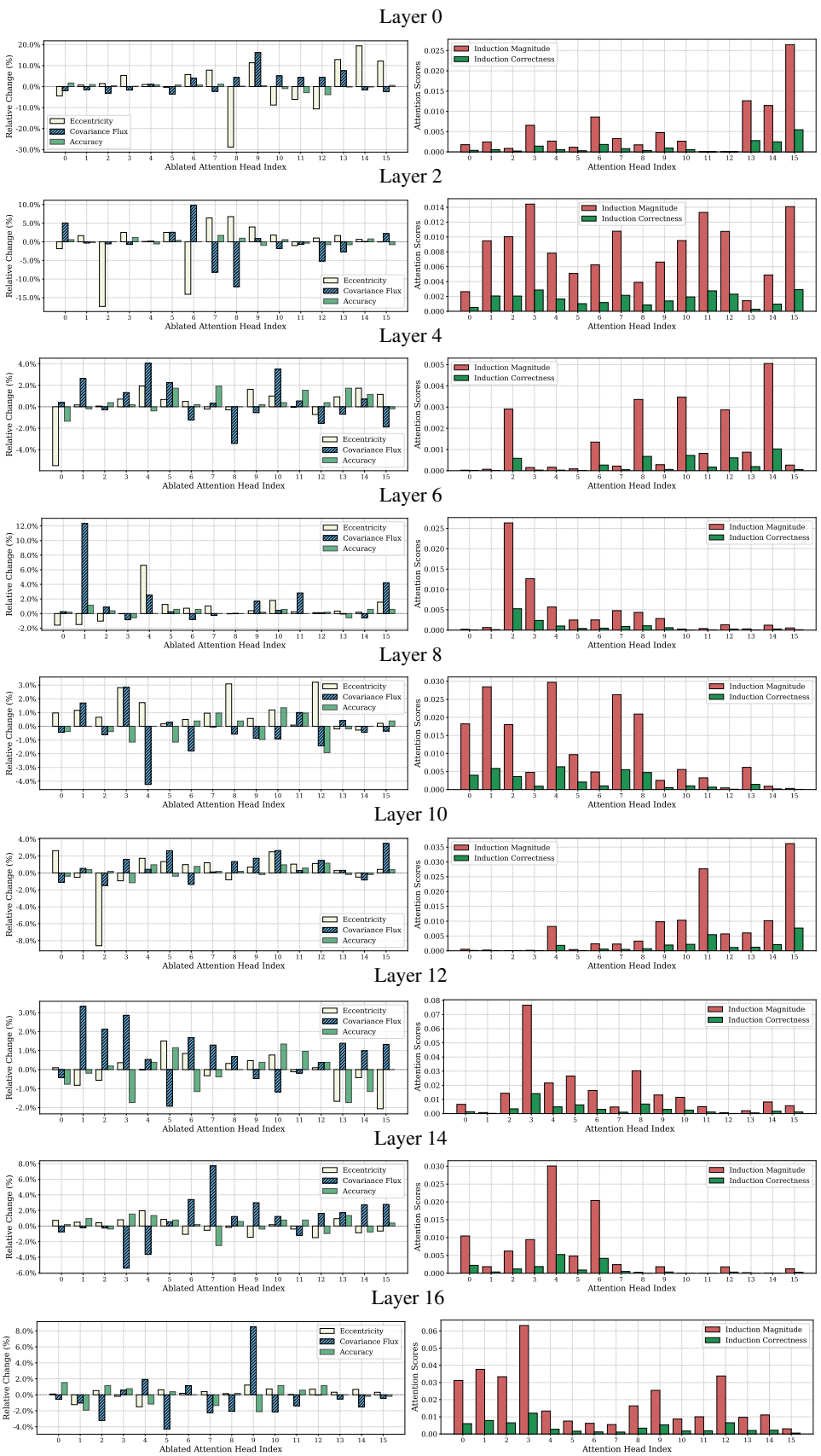

Layer 0
Layer 2
Layer 4
Layer 6
Layer 8
Layer 10
Layer 12
Layer 14
Layer 16
Eccentricity
Covariance Flux
Accuracy
Relative Change (%)
Ablated Attention Head Index
Induction Magnitude
Induction Correctness
Attention Scores
Attention Head Index



Figure B.66: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, FP.


Layer 0
Layer 2
Layer 4
Layer 6
Layer 8
Layer 10
Layer 12
Layer 14
Layer 16
Relative Change (%)
Ablated Attention Head Index
Eccentricity
Covariance Flux
Accuracy
Attention Scores
Attention Head Index
Induction Magnitude
Induction Correctness

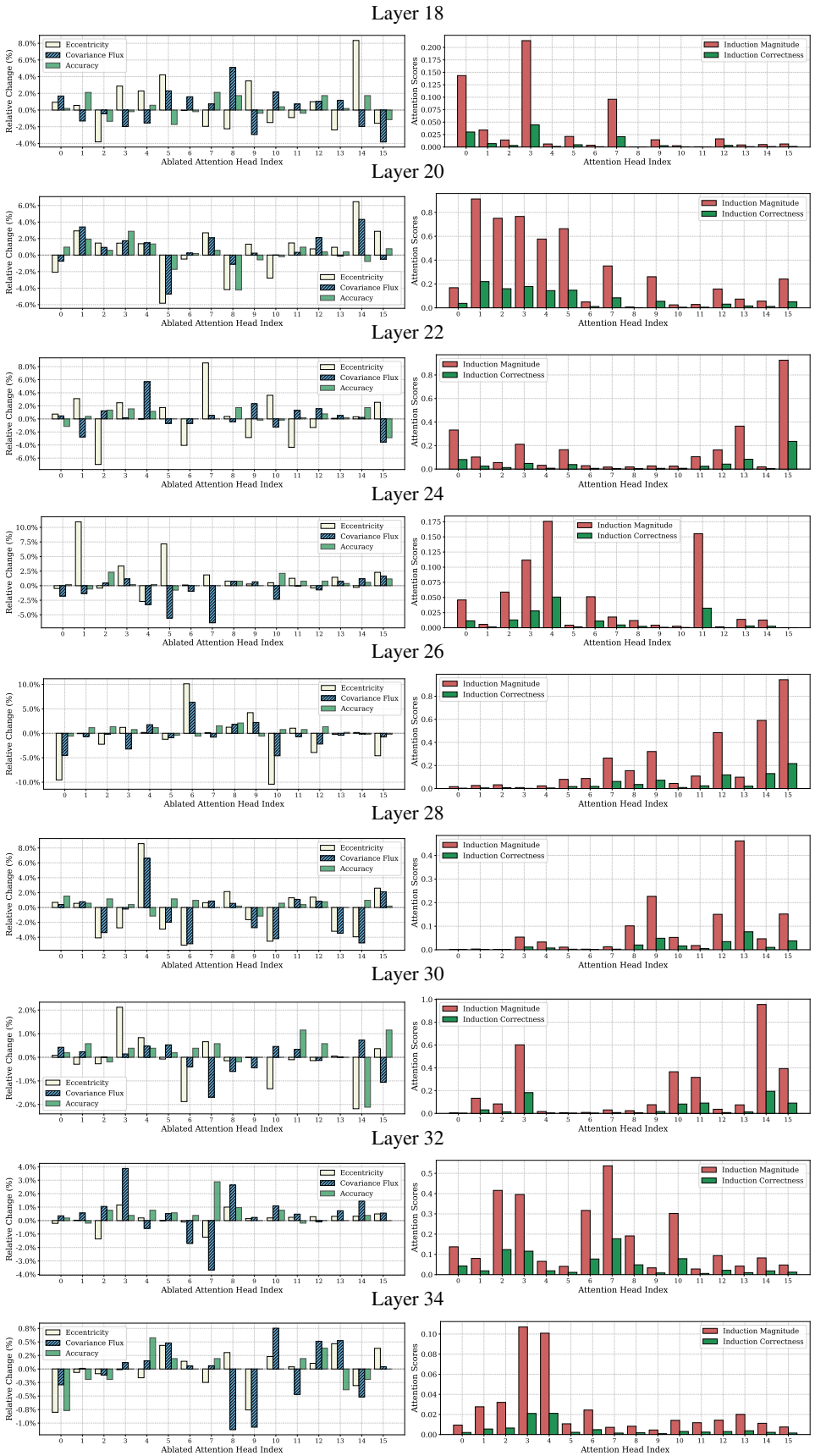
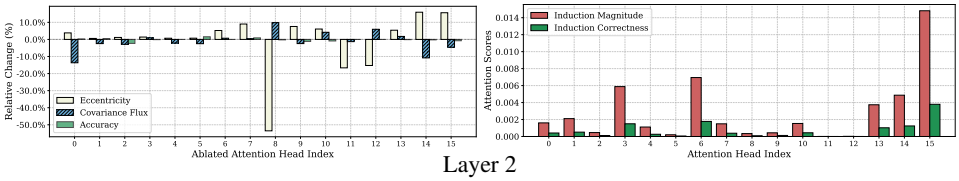
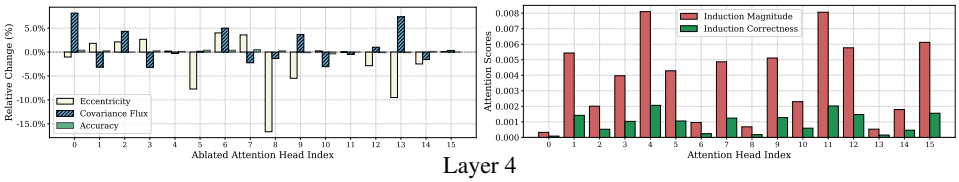


Figure B.67: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, SST-5.

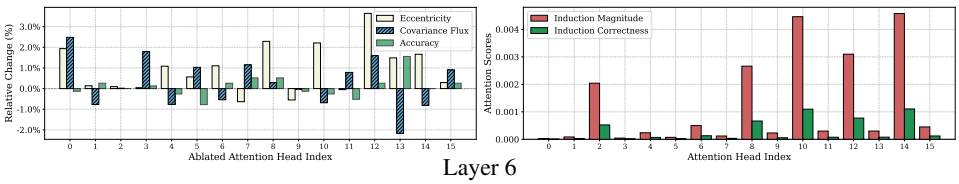
Layer 0



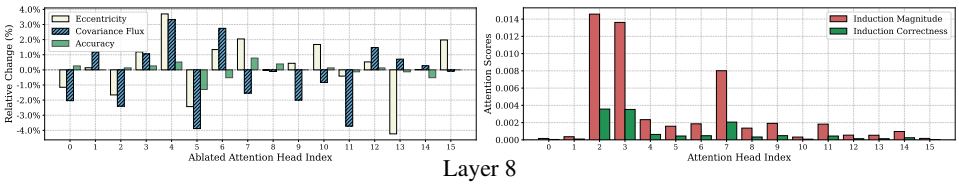
Layer 2



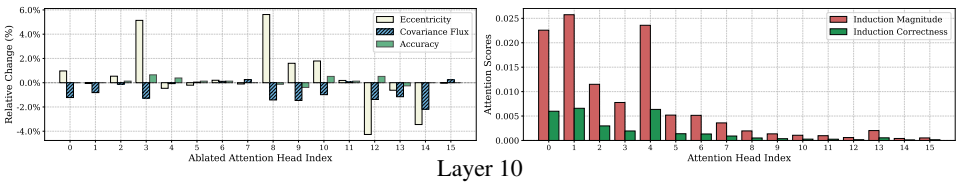
Layer 4



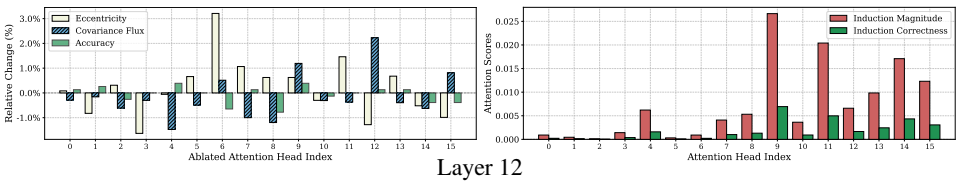
Layer 6



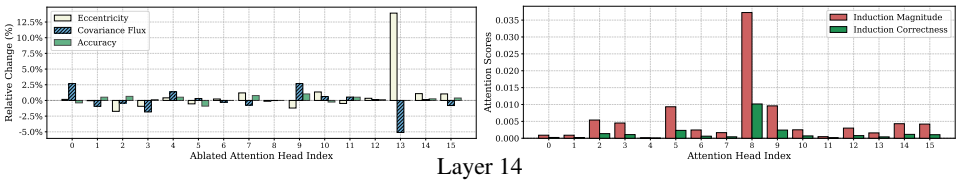
Layer 8



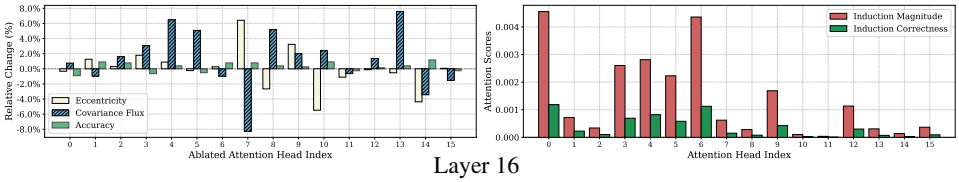
Layer 10



Layer 12



Layer 14



Layer 16

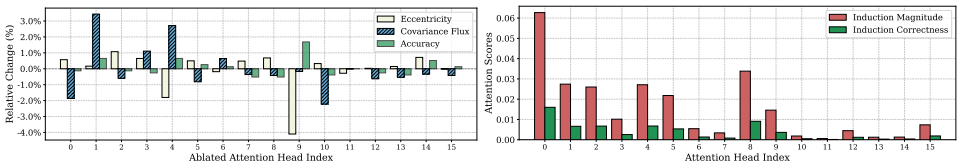
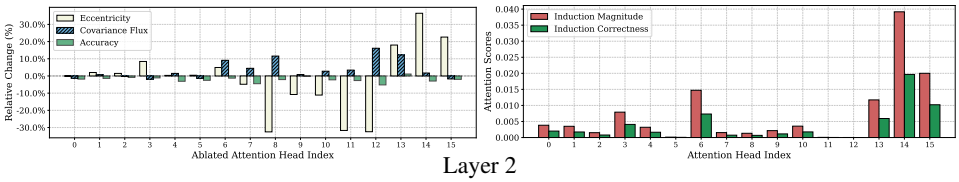


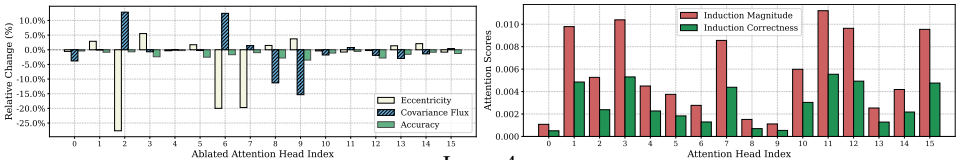


Figure B.68: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, AGNews.

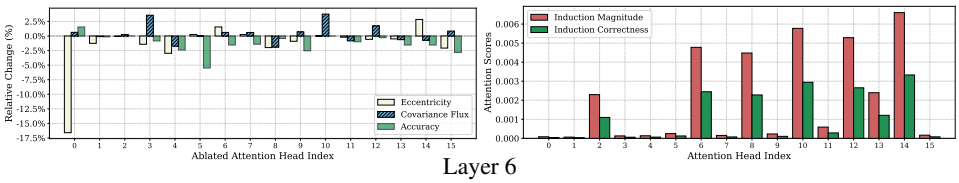
Layer 0



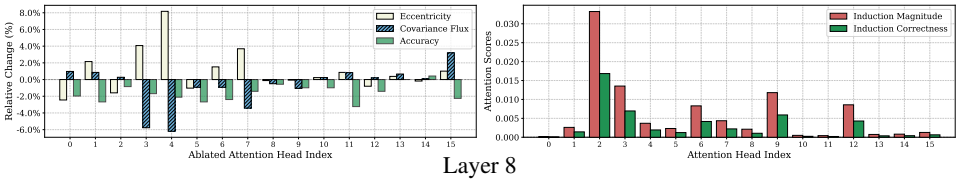
Layer 2



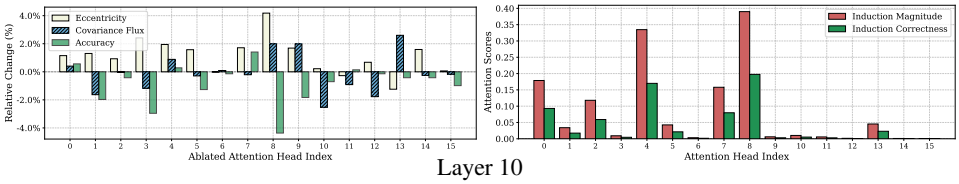
Layer 4



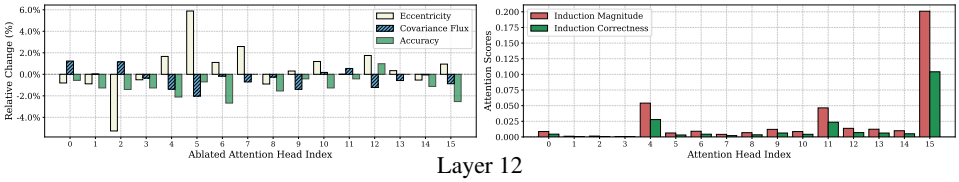
Layer 6



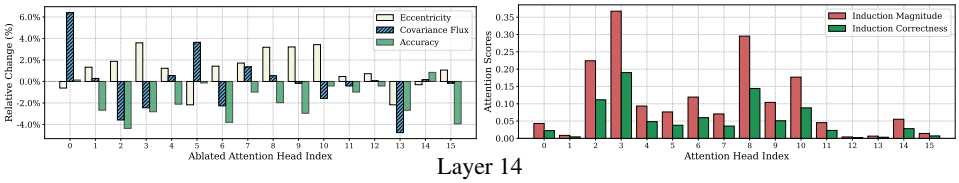
Layer 8



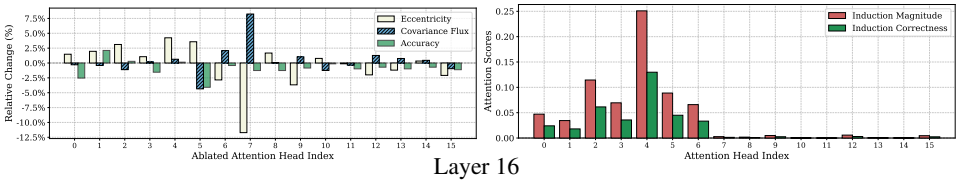
Layer 10



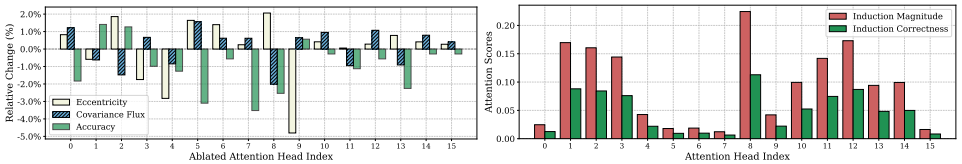
Layer 12



Layer 14



Layer 16



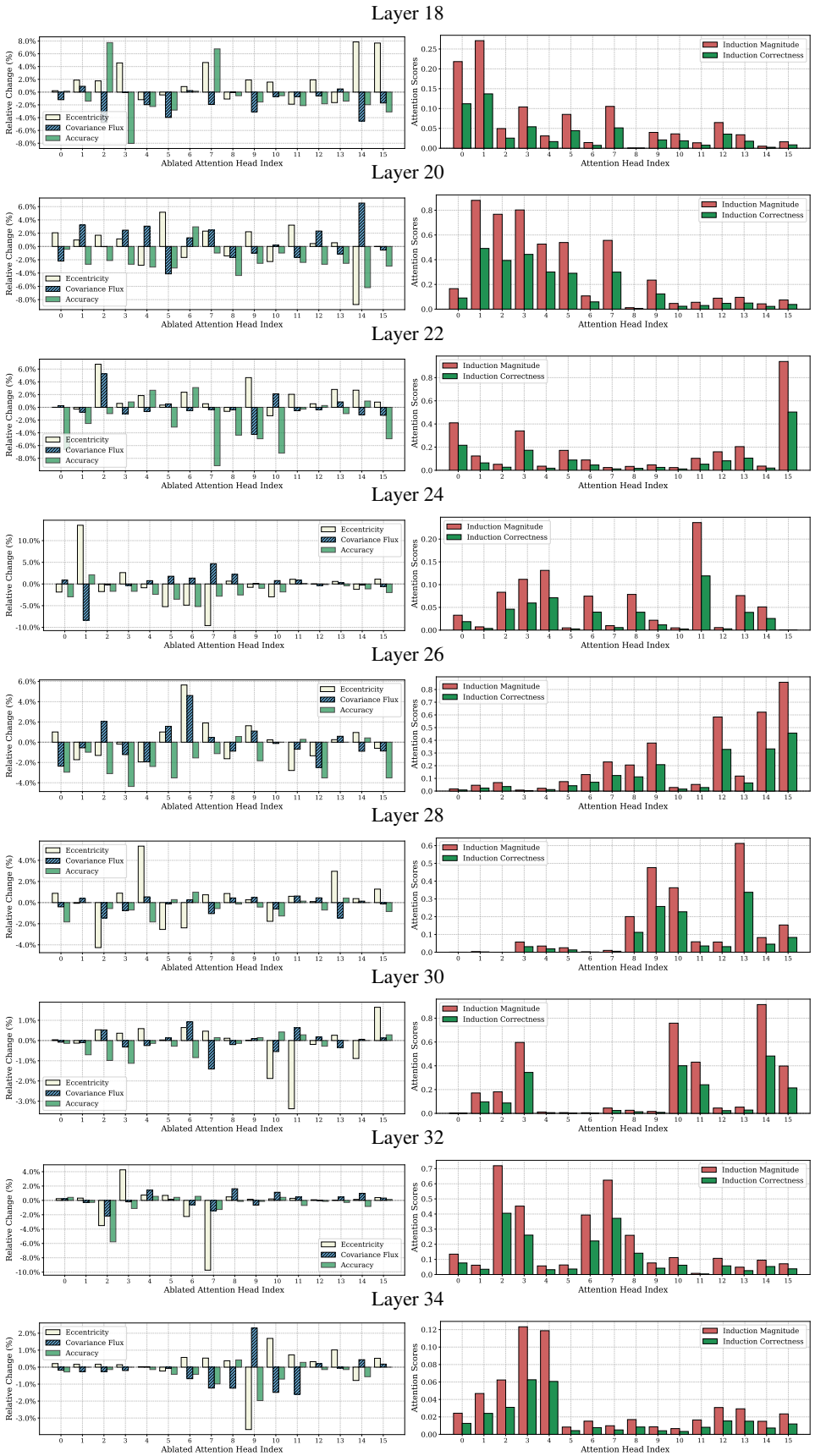
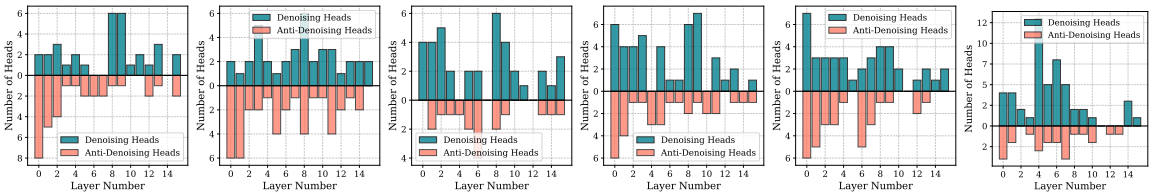
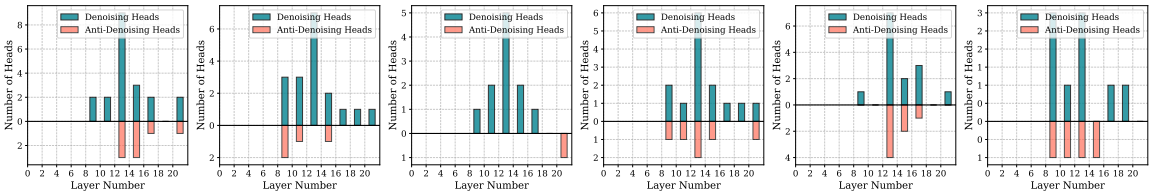


Figure B.69: (Left) augmentation results for Fig. 4.7, (right) induction score of each attention head on Qwen 2.5-3B Instruct, Subjective.



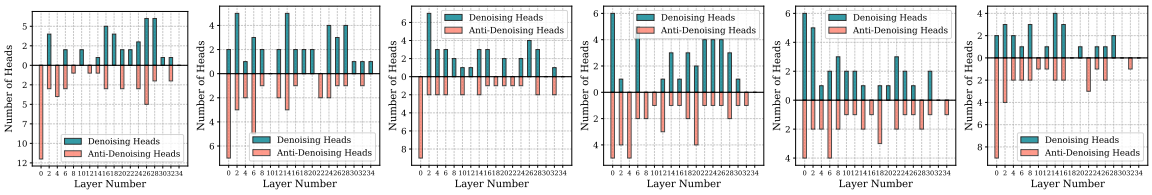
SST-2 MR FP SST-5 AGNews Subjective

Figure B.70: Augmentation results for Fig. 4.8 on Llama 3.2-1B.



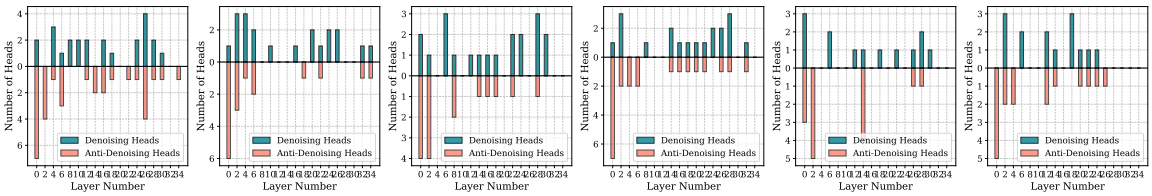
SST-2 MR FP SST-5 AGNews Subjective

Figure B.71: Augmentation results for Fig. 4.8 on Llama 3-8B.



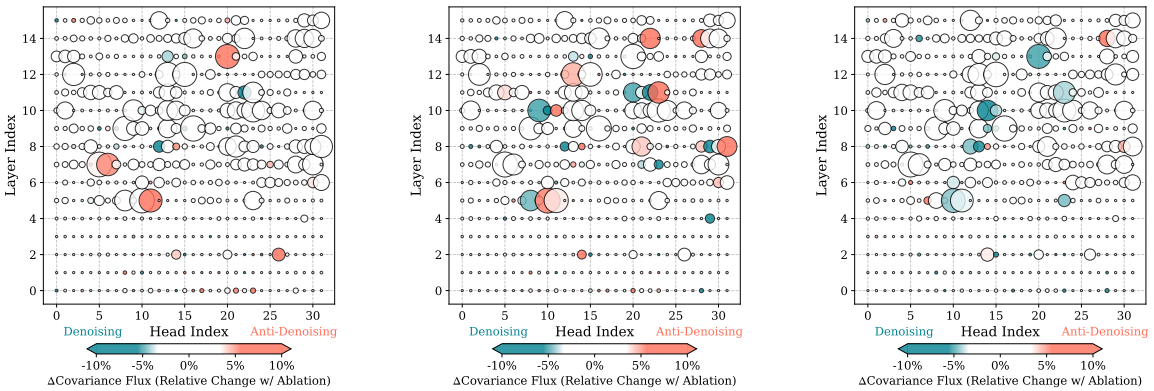
SST-2 MR FP SST-5 AGNews Subjective

Figure B.72: Augmentation results for Fig. 4.8 on Qwen 2.5-3B (the threshold is set to $\pm 2.5\%$).

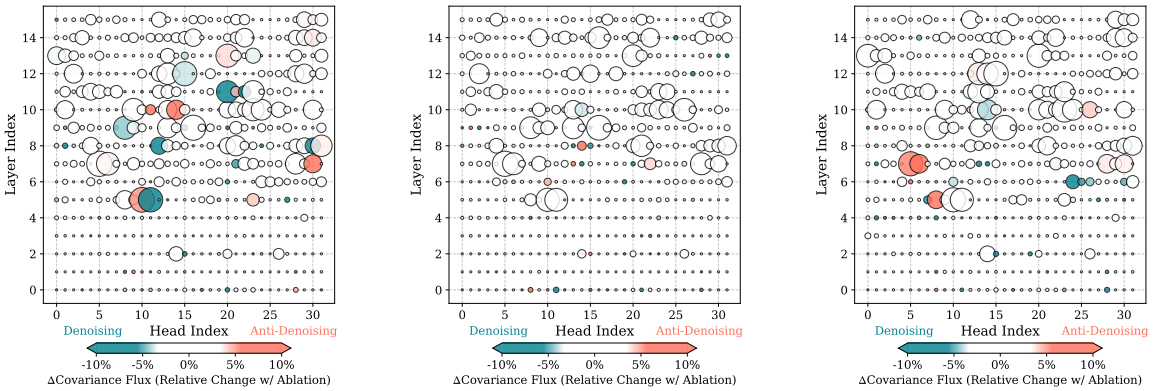


SST-2 MR FP SST-5 AGNews Subjective

Figure B.73: Augmentation results for Fig. 4.8 on Qwen 2.5-3B Instruct (the threshold is set to $\pm 2.5\%$).

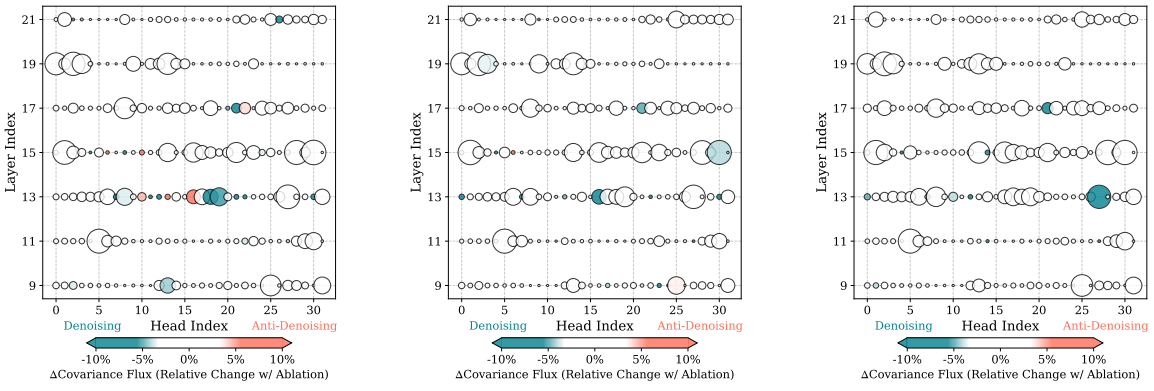


SST-2 MR FP

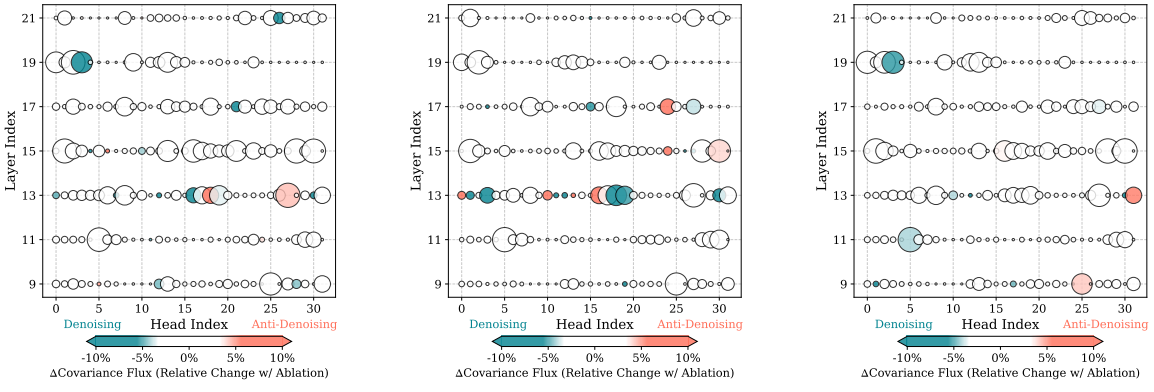


SST-5 AGNews Subjective

Figure B.74: Augmentation results for Fig. 4.9 on Llama 3.2-1B.



SST-2 MR FP



SST-5 AGNews Subjective

Figure B.75: Augmentation results for Fig. 4.9 on Llama 3-8B.

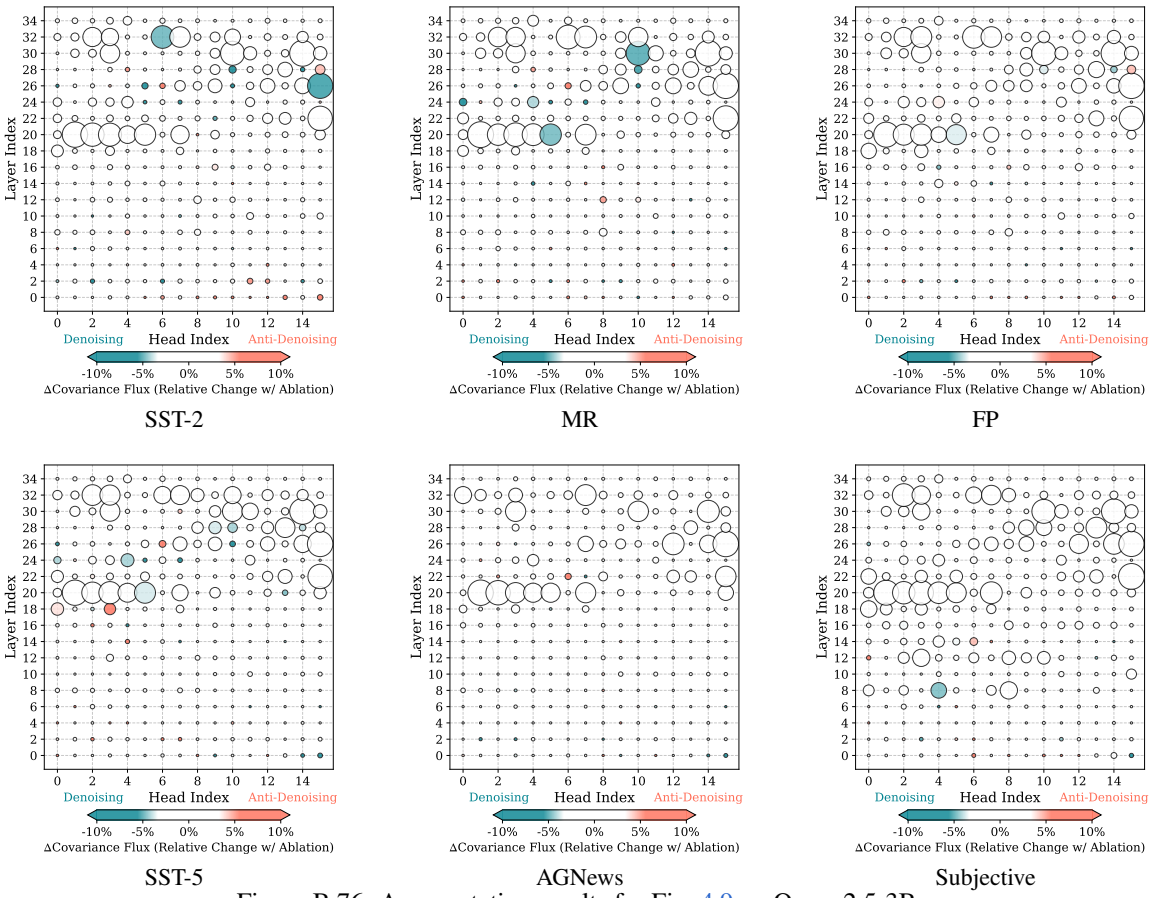


Figure B.76: Augmentation results for Fig. 4.9 on Qwen 2.5-3B.

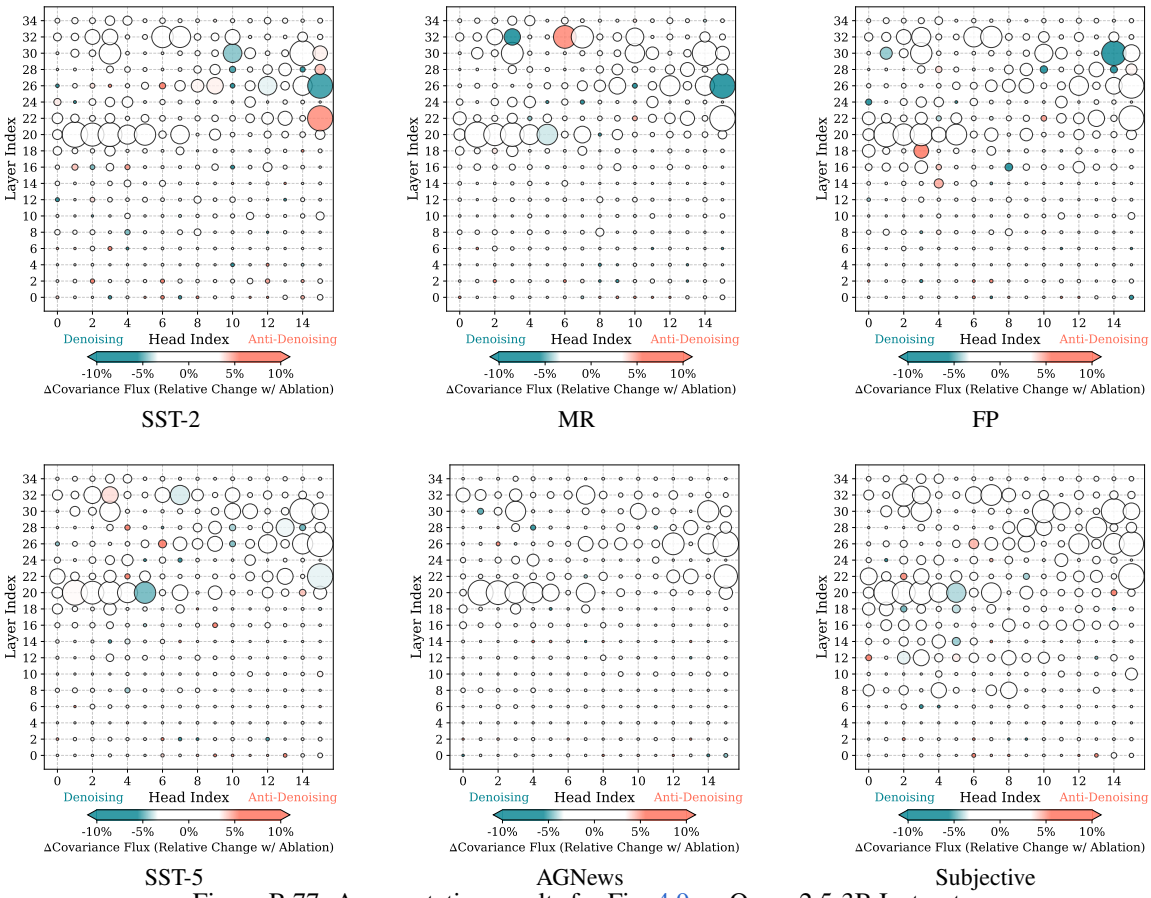


Figure B.77: Augmentation results for Fig. 4.9 on Qwen 2.5-3B Instruct.

Appendix C

Supplementary Materials for Chapter 5

C.1 Experimental Details

C.1.1 Datasets

In this chapter, 10 datasets are used as shown in Table C.2. Some datasets do not provide valid splitting, so we randomly split all of them into calibration sets and test sets: For each dataset, we first shuffle it with the random seed `42`. Then, we choose the 512 data at the tail as the testing data, and the 512 data at the head (all the datasets have more than 1024 examples) as the calibration data. Each data point in a test set is used once for each experiment trial to build a prompt example and test for performance.

`AGNews` and `GLUE-RTE` have over-length examples. So, in the main experiments, we filter out those examples: for Llama 2-6.9B, when $k = 8$, we filter out all the examples with a string length greater than 512 in `AGNews` and 128 in `GLUE-RTE`. Also, for Llama 3-8B, when $k = 8$, we filter out all the examples with a string length greater than 128 in GLUE-RTE and omit the experiments on `AGNews`. In the experiments in §5.5.2, for all the models, we filter out all the examples with a string length greater than 256 for all the k.

C.1.2 Baselines

6 baselines (1 vanilla and 5 improved) are used in this chapter. Here we introduce the 5 improved baselines.

Contextual Calibration (Con.C). Proposed by Zhao et al. (2021), Con.C uses empty queries with normal demonstrations as calibration samples to estimate the calibration term A. In detail, Con.C inputs m samples with empty queries into the model and gets the averaged normalized label probabilities $\bar{p}'$ among m samples. We take the reciprocal of the probabilities as calibration term $A = \bar{p}'^{-1}$, while the $B = \mathbf{0}$.

Table C.1: Prompt templates used in this work.

Dataset	Prompt Template	Verbalizer
AGNews	Input: [x], Label: [y]	world, sport, business, science
SemE.R	Input: [x], Aspect: [a], Label: [y]	positive, neutral, negative
SemE.L	Input: [x], Aspect: [a], Label: [y]	positive, neutral, negative
PoemS	Input: [x], Label: [y]	positive, neutral, negative, mix
RTE	Input: [x], Text 2: [a], Label: [y]	include, neutral
TEE	Input: [x], Label: [y]	anger, joy, positive, sad
TEH	Input: [x], Label: [y]	normal, hate
TES	Input: [x], Label: [y]	positive, neutral, negative
FP	Input: [x], Label: [y]	positive, neutral, negative
Rott.T	Input: [x], Label: [y]	positive, negative

Batch Calibration (Bat.C). Proposed by Zhou et al. (2024a), Bat.C is an inference-time calibration, using the negative averaged normalized label probabilities $-\bar{p}$ of m samples in inference time as the calibration term $B = -\bar{p}$, while the $A = \mathbf{1}$, where $\mathbf{1}$ is the all-one vector.

Domain Calibration (Dom.C). Proposed by Fei et al. (2023), Dom.C acts similarly to the Con.C, with the difference that it uses a random sequence sampled on the random tokens from the calibration dataset as queries instead of empty ones. We fix the sampled length to 32.

KNN Prompt (KNN). Proposed by Xu et al. (2022), KNN uses the whole output vocabulary probability distribution as the classification feature, instead of the label tokens. In detail, first, features of calibration examples are calculated as k-NN anchors. Then, during the inference, a k-NN classifier is used to classify the feature from the test samples. We use m examples to calculate the anchors for k-NN, and the nearest neighbor number is set to 3.

Central Calibration (Cent.C). This is the control method proposed by us with a calculation process completely consistent with the Hidden Calibration, except that the usage of the hidden state is replaced by the whole output vocabulary probability distribution consistent with KNN. This method compares with Hidden Calibration to prove that the output probability distribution is not a good classification feature for ICL in a

controlled setting.

Notice that: these label-probability-based methods (Con.C, Bat.C, Dom.C) use A or B *along*, which may be another major drawback of these calibration methods: According to Fig. 5.2, if a calibration rotates the mapping direction suitably, and transfer the 0-point properly, a decision boundary close to the Hidden Calibration can be found. This also leads to a new research direction for calibration: the simultaneous usage of translation and rotation methods.

C.1.3 Prompts

In this chapter, we use a minimum prompt template shown in Table C.1. The separator between demonstrations is "\n".

To facilitate the replication of label probability-based methods, we limit all the labels in the label space to one token by synonymous conversion. Note that Hidden Calibration does not need to meet such a one-token requirement.

Especially, in §5.4.3, we use 6 more prompt templates to test the stability of each ICL method against the prompt templates. We list these extra templates in Table C.3.

Table C.2: Datasets and Abbreviations used in this work.

Dataset	Abbr.
`AGNews` (Zhang et al., 2015)	AGNews
`SemEval 2014-Task 4 Restaurant` (Pontiki et al., 2014)	SemE.R
`SemEval 2014-Task 4 Laptops` (Pontiki et al., 2014)	SemE.L
`Poem Sentiment` (Sheng & Uthus, 2020)	PoemS
`GLUE-RTE` (Wang et al., 2019)	RTE
`tweet_eval_emotion` (Mohammad et al., 2018)	TEE
`tweet_eval_hate` (Basile et al., 2019)	TEH
`tweet_eval_sentiment` (Rosenthal et al., 2017)	TES
`financial_phrasebank (all agree)` (Malo et al., 2014)	FP
`rotten_tomatoes` (Pang & Lee, 2005)	Rott.T

C.1.4 Details of Visualization in §5.3.1

Principle Component Analysis (PCA). Given a hidden state set $\mathcal{H} = \left\{h^{(i)}\right\}_{i=1}^{n}$, we span all the hidden state vector into a matrix $H \in \mathbb{R}^{n\times d}$. The covariance matrix is $\mathrm{cov}\left(H\right) = \frac{1}{n}\left(H-\bar{H}\right)^T\left(H-\bar{H}\right)$, where the $\bar{H}$ is the matrix spanned by the element-wise average vectors $\bar{h}$ of hidden state set $\mathcal{H}$. We conduct Eigenvalue Decomposition on $\mathrm{cov}\left(H\right)$ and adjust the dimensions to arrange the eigenvalues Λ in a descending order along the row:

$$\mathrm{cov}\left(H\right) = P\Lambda P^T, \tag{C.1}$$

where the $P \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. Taking the top-$\tilde{d}$ lines of P and span them into $\tilde{P} \in \mathbb{R}^{d\times\tilde{d}}$, we get the principle component mapping:

$$\mathrm{PCA}_{\mathcal{H}}(h) = \left(h-\bar{h}\right)\tilde{P} = h\tilde{P} - \bar{h}\tilde{P}. \tag{C.2}$$

Notice that $\tilde{P}\tilde{P}^T = I$, where I is the identity matrix.

Dot-product after PCA. Suppose we have dot-product with vector[29] h and E in the original space $\mathbb{R}^d$, producing the dot-product similarity classification criterion α:

$$\alpha = h\left(E^T - \mathbf{0}^T\right). \tag{C.3}$$

When we conduct a same PCA on both h and E^T to get dot-product similarity in a dimensionality-reduced space similar to Fig. 5.2:

$$\tilde{\alpha} = \mathrm{PCA}_{\mathcal{H}}\left(h\right)\underbrace{\left(\mathrm{PCA}_{\mathcal{H}}\left(E\right)^T - \mathrm{PCA}_{\mathcal{H}}\left(\mathbf{0}\right)^T\right)}_{\text{Mapping direction selected after PCA}} \tag{C.4}$$

$$= \left(h\tilde{P} - \bar{h}\tilde{P}\right)\left(E\tilde{P}\right)^T \tag{C.5}$$

$$= h\tilde{P}\tilde{P}^T E^T - \bar{h}\tilde{P}\tilde{P}^T E^T \tag{C.6}$$

$$= \alpha - \bar{h}E^T. \tag{C.7}$$

[29] Due to excessive superscripts, in this section, we omit the superscripts U in the notation of un-embedding E_l^U.

Notice that we use the mapping direction $\left(\mathrm{PCA}_{\mathcal{H}}\left(E\right)^T - \mathrm{PCA}_{\mathcal{H}}\left(\mathbf{0}\right)^T\right)$ after the PCA, instead of $\left(\mathrm{PCA}_{\mathcal{H}}\left(E\right)^T - \mathbf{0}^T\right)$, and this is the reason why the oblique axis in Fig. 5.2 does not necessarily pass through the coordinate origin. In such a scenario, the dot productions after PCA only differ by a fixed constant bias $-\bar{h}E^T$ from the ones before PCA. This is the reason why the normal line of oblique axis on the 0-point doesn't pass the coordinate origin of the 2D-plane in Fig. 5.2.

Table C.3: The 7 prompt templates used in the experiment (on `Rotten_Tomatoes`) of Fig. 5.5 (left).

#	Prompt Template
Original	Input: [x], Label: [y]
1	Sentence: [x], Label: [y]
2	sentence: [x], Label: [y]
3	sentence:\n [x], Label: [y]
4	Input: [x], Sentiment: [y]
5	Input: [x], sentiment: [y]
6	x: [x], y: [y]

Decision Boundary after PCA. Notice that the decision boundary of two classes l_1 and l_2 in an non-rotated ICL scenario is:

$$\mathcal{B} = \left\{h | hE_{l_1}^T - hE_{l_2}^T = C\right\}. \tag{C.8}$$

Where the C is the calibration term without rotation. Notice that it is a hyperplane in $\mathbb{R}^d$ with normal vector $\left(E_{l_1} - E_{l_2}\right)^T$. Also, the normal plane which pass the 0-point of direction $\left(E_{l_1} - E_{l_2}\right)^T$ in $\mathbb{R}^{\tilde{d}}$ after PCA is:

$$\begin{aligned}\tilde{\mathcal{B}} = \{&\mathrm{PCA}_{\mathcal{H}}(h) | \mathrm{PCA}_{\mathcal{H}}(h) \\ &\left(\mathrm{PCA}_{\mathcal{H}}(E_{l_1} - E_{l_2}) - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0})\right)^T = 0\}.\end{aligned} \tag{C.9}$$

By the aforementioned transformation, we have:

$$\tilde{\mathcal{B}} = \left\{\mathrm{PCA}_{\mathcal{H}}(h) | hE_{l_1}^T - hE_{l_2}^T = \bar{h}\left(E_{l_1}^T - E_{l_2}^T\right)\right\}. \tag{C.10}$$

That is, the dimensionality-reduced decision boundary $\tilde{\mathcal{B}}$ is perpendicular to the mapped direction $\left(\mathrm{PCA}_{\mathcal{H}}\left(E_{l_1} - E_{l_2}\right) - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0})\right)$, and biased only by a constant $\left(\bar{h}\left(E_{l_1}^T - E_{l_2}^T\right) - C\right)$ on the classification criteria comparing to the original space. Specifically, in the two-dimensional case, it is a straight line that may not necessarily pass through the coordinate origin, as shown in Fig. 5.2.

C.1.5 Details of Experiment in §5.5.1

C.1.5.1 Calculation Details of Averaged Overlap

First, we divide the $|\mathcal{Y}|$-way classification task into $\mathbb{C}(|\mathcal{Y}|, 2)$ 2-way classification task[30], to allow us to use a scalar to characterize the classification criteria for each 2-combination (similar to what we do to the "positive" and "negative" examples in Fig. 5.2). Then, for each chosen 2-combination, w.l.o.g, given labels denoted as l_1 and l_2, we build prompt-label sets[31] as:

$$\mathcal{S}_{l_j} = \left\{T\left(\mathcal{D}^{de,(i)}, x^{(c_i)}\right) \middle| y^{(c_i)} = l_j\right\}_{l_j \in \{l_1, l_2\}}^{n_{l_j}}, \tag{C.11}$$

where c_i is the sampled query index. That is, we sample queries annotated with these two labels and build prompt sets, then collect the prompts with the same query label l_j into $\mathcal{S}_{l_j}$, with a size n_{l_j}.

Then, for each prompt $s^{(i)} = T\left(\mathcal{D}^{de,(i)}, x^{(i)}\right) \in \mathcal{S}_{l_j}$, we run decoders (vanilla, Con.C, Dom.C and Hidden Calibration) with probability normlization $f_{l_1}(\cdot)$ and $f_{l_2}(\cdot)$ to get the classification probabilities of assigning label l_1 and l_2 as $\alpha_1^{(i)} = f_{l_1}\left(s^{(i)}\right)$ and $\alpha_2^{(i)} = f_{l_2}\left(s^{(i)}\right)$. We calculate the difference between $\alpha_1^{(i)}$ and $\alpha_2^{(i)}$ and collect them into a set:

$$\mathcal{A}_{l_j} = \left\{\alpha_1^{(i)} - \alpha_2^{(i)} \middle| s^{(i)} \in \mathcal{S}_{l_j}\right\}_{i=1}^{n_{l_j}}. \tag{C.12}$$

Now, for the 2-combination of labels (l_1, l_2), we get $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$, whose elements are the probabilities difference between assigning l_1 and assigning l_2 to example $s^{(i)}$. The difference between $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$ is: the elements in $\mathcal{A}_{l_1}$ are from $s^{(i)}$s with queries labeled by ground-truth l_1, and vice versa. We obtain continuous probability density functions of $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$ as $p_{l_1}(\cdot)$ and $p_{l_2}(\cdot)$ by kernel density estimation, as the curves in Fig. 5.7.

Then, we calculate the overlap area of these curves:

$$S_{l_1,l_2} = \int_{-1}^{1} \min\left[p_{l_1}(x), p_{l_2}(x)\right] \mathrm{d}x. \tag{C.13}$$

[30] The $\mathbb{C}(m, n)$ is the n-combination number from m elements.

[31] Notice that the T is the prompting function.

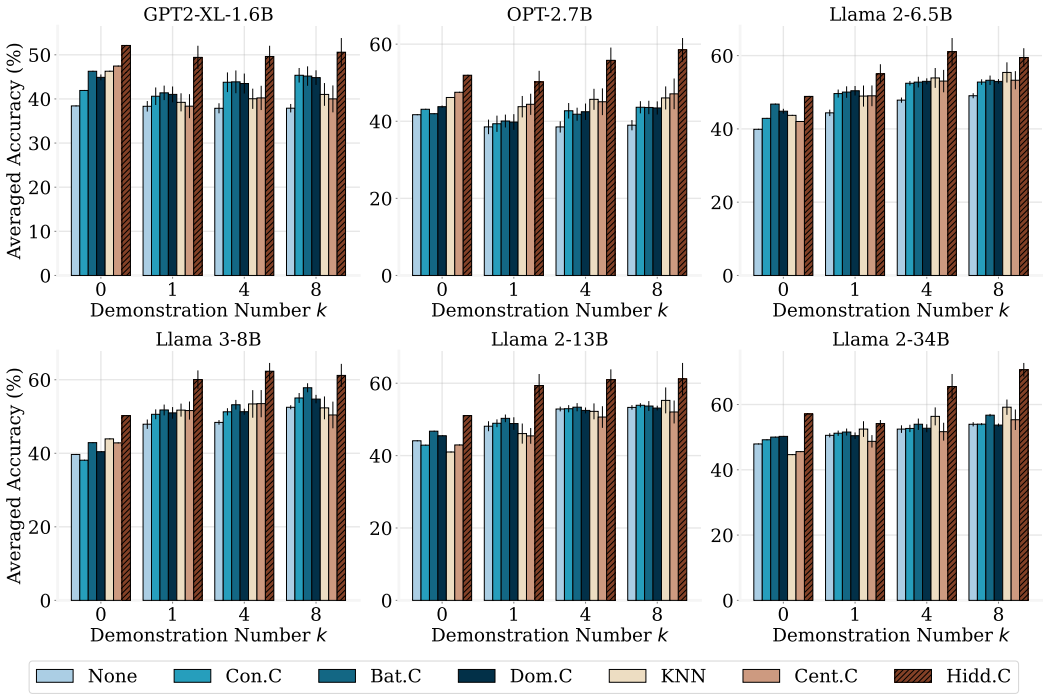


Figure C.1: The classification performance (Accuracy(%)) of 3 models averaged on 10 datasets.

For each combination[32] in the $\mathbb{C}(|\mathcal{Y}|, 2)$ 2-combinations, we repeat to calculate the $S_{\cdot,\cdot}$, and average them as the **Averaged Overlap** $\bar{S}$.

$$\bar{S} = \frac{1}{\mathbb{C}(|\mathcal{Y}|, 2)} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=i+1}^{|\mathcal{Y}|} S_{l_i, l_j}. \tag{C.14}$$

C.1.5.2 Experimental Details in §5.5.1

We conduct experiments resulting Fig. 5.8 on 3 models with `SemEval 2014-Task 4 Restaurant`, `SemEval 2014-Task 4 Laptops`, AGNews, Poem Sentiment, and fiancial_phrasebank, given the demonstration number $k = 4$ and calibration example numbers $m = 16$. We use the whole 512 examples on the test split for each dataset and repeat 5 times.

C.1.5.3 Proof: the Overlap Area is Double to the Error's Lower Bound

Suppose a label combination l_1 and l_2, w.l.o.g., we have a ground truth probability density function $p_{l_1}(x)$ and $p_{l_2}(x)$ on a criterion $x \in \mathbb{X}$, same as the curves in Fig. 5.7. Given a specific value of criterion x, the upper-bound classification performance is determined by majority vote, which is the most accurate method on such a point, resulting in a density of error classification:

$$e(x)_{l_1, l_2} \geqslant \min\left[p_{l_1}(x), p_{l_2}(x)\right]. \tag{C.15}$$

So, the integral error rate:

$$\mathcal{E}_{l_1, l_2} \geqslant \frac{\int_{x \in \mathbb{X}} \min\left[p_{l_1}(x), p_{l_2}(x)\right] \mathrm{d}x}{\int_{x \in \mathbb{X}} p_{l_1}(x) dx + \int_{x \in \mathbb{X}} p_{l_2}(x) \mathrm{d}x} \tag{C.16}$$

$$= \frac{1}{2} \int_{x \in \mathbb{X}} \min\left[p_{l_1}(x), p_{l_2}(x)\right] \mathrm{d}x \tag{C.17}$$

$$= \frac{1}{2} S_{l_1, l_2}. \tag{C.18}$$

[32] Notice that on $S_{\cdot,\cdot}$, the labels are rotational symmetry.

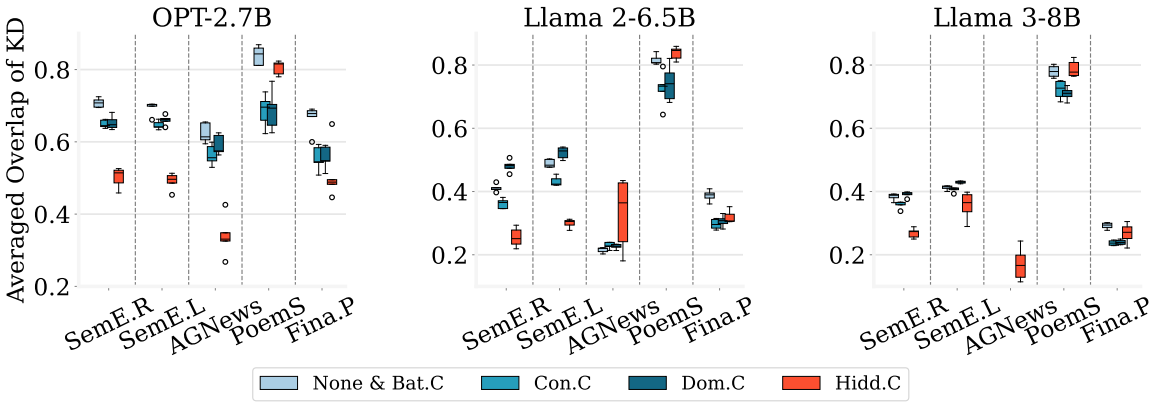


Figure C.2: The augmented results on 2 models of Fig. 5.8.

C.1.6 Details of Experiment in §5.5.2

C.1.6.1 Calculation of the Distance and Standard Error

Averaged Centroid Distance. Given a $|\mathcal{Y}|$-way classification task, for each label l we build its corresponding prompt set $\mathcal{S}_l = \left\{s^{(c_i)}|y^{(c_i)} = l\right\}_{i=1}^{n_l}$, where $s^{(c_i)}$ is the prompt with query labeled by l, and c_i is the sampled query index. We encode it into a hidden state set $\mathcal{H}_l = \left\{h^{(i)}\right\}_{i=1}^{n_l}$, and calculate its centroid $\bar{h}_l$, as what we do in Hidden Calibration:

$$\bar{h}_l = \frac{1}{n_l} \sum_{h^{(i)} \in \mathcal{H}_l} h^{(i)}. \tag{C.19}$$

For every 2-combination of labels l and l', we calculate the distance of their centroid, and the average among all the 2-combination is used as the Averaged Centroid Distance:

$$\text{ACD} = \frac{1}{\mathbb{C}(|\mathcal{Y}|, 2)} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=i+1}^{|\mathcal{Y}|} \left\|\bar{h}_i - \bar{h}_j\right\|_2. \tag{C.20}$$

Averaged Intra-class Standard Error. Given the hidden state set $\mathcal{H}_l = \left\{h^{(i)}\right\}_{i=1}^{n_l}$ w.r.t. the label l, we span all the hidden state vectors into a matrix $H_l \in \mathbb{R}^{n_l \times d}$. The covariance matrix is $\left(H_l - \bar{H}_l\right)^T \left(H_l - \bar{H}_l\right)$, where the $\bar{H}_l$ is the matrix spanned by the element-wise average vectors of hidden state set $\mathcal{H}_l$. Notice that the ACD is a first-order moment, for a proper comparison, we use the average on the diagonal elements of the element-wise square root of the covariance matrix as the intra-class standard error metric for label l. We average all the standard errors from all the classes as the Averaged Intra-class Standard Error:

$$\text{AIS} = \frac{1}{|\mathcal{Y}|d} \sum_{i=1}^{|\mathcal{Y}|} \text{tr}\left[\sqrt{\left(H_i - \bar{H}_i\right)^T \left(H_i - \bar{H}_i\right)}\right]. \tag{C.21}$$

C.1.6.2 Experimental Details in §5.5.2

We conduct experiments resulting Fig. 5.11 on 4 models with `SemEval 2014-Task 4 Restaurant`, `SemEval 2014-Task 4 Laptops`, AGNews, Poem Sentiment, and fiancial_phrasebank, given the calibration example numbers $m = 16$. We use the whole 512 examples on the test split for each dataset and repeat 5 times.

C.1.7 Experimental Details for Fig. 5.6

We conduct experiments resulting Fig. 5.6 on OPT-2.7B with 4 datasets: `SemEval 2014-Task 4 Restaurant`, `SemEval 2014-Task 4 Laptops`, AGNews, and Poem Sentiment, given the demonstration numbers $k = 4$ and repeat 5 times.

Table C.4: Performance of all 6 models on TREC for a more-way classification, and on Hate_Speech18 for a biased dataset. $k = 4$, top-2 results are in **bold**.

	GPT2-XL-1.6B	OPT-2.7B	Llama 2-6.5B	Llama 3-8B	Llama 2-13B	Llama 2-34B
			TREC			
None	13.02	15.85	23.12	16.79	23.22	21.13
Con.C	14.10	8.12	23.72	18.34	23.56	22.35
Bat.C	14.44	17.31	23.26	20.28	22.88	22.15
Dom.C	14.10	8.58	23.14	19.10	23.42	23.23
KNN	**27.53**	33.80	49.25	43.15	54.76	**53.66**
Cent.C	23.46	**33.80**	**49.18**	**46.12**	**56.79**	53.03
Hidd.C	**55.91**	**61.14**	**64.90**	**71.59**	**75.64**	**68.39**
			Hate_Speech18			
None	**23.96**	23.95	**23.96**	**23.96**	**23.96**	23.92
Con.C	21.57	25.47	14.68	**23.74**	21.49	21.25
Bat.C	20.59	21.25	20.54	18.25	20.66	23.68
Dom.C	21.23	23.68	20.73	23.67	**24.36**	23.28
KNN	17.30	16.74	16.50	16.26	16.23	**27.05**
Cent.C	16.32	**25.47**	15.42	5.46	9.15	22.23
Hidd.C	**24.47**	**27.03**	**23.44**	20.67	20.06	**29.94**

C.2 Detailed Results

C.2.1 Details of Main Results

Numerical details of Fig. 5.4 are shown in Table C.7, C.8, C.9, C.10, C.11 and C.12. Accuracy results are shown in Fig. C.1.

C.2.2 Details of Averaged Overlaps Results

The augmented results on the other 3 models (we skip this experiment on the quantitated model) of Fig. 5.8 are shown in Fig. C.2.

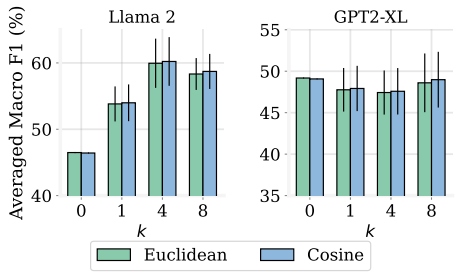


Figure C.3: The classification performance (Macro F1(%)) of Hidden Calibration with difference similarity measure.

C.3 Additional Discussion

C.3.1 The Similarity Measures Used in Hidden Calibration

In §5.3.2, we use the Euclidean distance as the similarity measure, while this is not the only option. Intuitively, we can choose other similarity measures as alternatives. Moreover, since we get inspired by observation with dot-production similarity, we should check the performance on such a measure instead of the Euclidean distance. This section uses cosine similarity as an example to illustrate that there is no significant performance difference between these measures. We use cosine similarity to repeat the results in §5.4.2 on Llama 2-6.9B and GPT 2-XL.

The results are shown in Fig. C.3, where the performance based on these two measures is close, without a statistical difference. This indicates that the hidden space has good properties of both metric and vector space, and Hidden Calibration acts equally on these measures.

C.3.2 Transferability of the Centroid

We have proven that it is not advisable to use the *common* token probability criteria, while, since the centroid criteria are proven to be better than token probability, we are curious: can the centroid calculated in one task be transferred to other tasks with the same label space? Among the datasets sharing the same label space "positive", "neutral", and "negative", we calculate centroids by one dataset and evaluate Hidden Calibration with it on another dataset, on OPT-2.7B, with $k = 4$, $m = 16|\mathcal{Y}|$. The results are shown in Table C.5, where only limited transferability is demonstrated in different domains of the same task (SemE.R and SemE.L), whose behavior is similar to *task vector* (Ilharco et al., 2023; Hendel et al., 2023), while other combinations

of datasets can not demonstrate considerable transferability. This further exacerbates our doubts about the token-based method: We find that the hidden state distributions have significant differences among various datasets, even if they share a common label space, then utilizing fixed token un-embedding vectors to decode these classification criteria is highly unreliable.

Moreover, we repeat this experiment on various k, instead of various datasets, as shown in Table C.6. The transferabilities among k are better than on datasets, but still worse than the un-transferred scenario. Notice that $4 \to 1$ results are much better than $0 \to 1$, which support our results in §5.5.2: hidden states with higher k are further converged.

Table C.5: Transferability of centroid among various datasets with the same label space. Big numbers are the average improvement (MF1) compared to vanilla ICL, small numbers are standard error. Statistically significant results ($p < 0.1$) are in **bold**.

Test \ Cali.	SemE.R	SemE.L	Fina.P	TES
SemE.R	**(+38.75)** ±2.28	**+29.24** ±3.19	+6.32 ±10.55	+7.54 ±8.96
SemE.L	**+20.78** ±7.37	**(+37.33)** ±3.47	-0.40 ±7.37	+8.94 ±8.93
Fina.P	**+7.42** ±4.98	+9.05 ±11.14	**(+37.29)** ±2.30	-4.35 ±6.34
TES	+6.95 ±7.00	**+9.73** ±5.68	-0.51 ±3.83	**(+11.83)** ±3.59

C.3.3 A Demonstration towards ICL Principles

Our findings may lead to an explanation of the principle of ICL and traditional calibrations. LMs generate distributed representations into separate clusters in the last hidden state. At this point, by dot-product, any non-collinear *arbitrary or plausible* mapping directions should be able to capture and classify these clusters to some extent. Note: The absolute distance in such a direction is not faithful (since the centroids of these hidden states and the coordinate origins in these mapping directions are not necessarily aligned), which leads to the generation of so-called bias, and calibrating these biases can improve the performance to a certain extent. However, in such a paradigm, high-dimensional features are discarded, resulting in overlapping originally linearly separable features in high-dimensional space, leading to a loss of classification accuracy, even if the calibration aligns the coordinate origin.

C.3.4 Applicability on More-way Classification and Biased Dataset

To further verify the applicability of Hidden Calibration on harder tasks, we test the performance of Hidden Calibration and baseline methods on TREC (Li & Roth, 2002; Hovy et al., 2001) for more-way ($|\mathcal{Y}| = 6$) classification and Hate_Speech18 (de Gibert et al., 2018) for a biased dataset (label frequency distribution: $[0.87, 0.11, 0.007, 0.015]$), shown in Table C.4. For results on TREC, Hidden Calibration produces a significant improvement compared to all the baselines on all the models. While, Hidden Calibration outperforms on Hate_Speech18 in most cases, and sometimes Hidden Calibration is weaker than baseline methods, but consistently produces competitive results. However, we believe that this slightly weaker result cannot be fully attributed to biased datasets, given that there are biased datasets among the 10 standard datasets where Hidden Calibration perform SotA results (e.g. `SemEval 2014-Task 4 Restaurant`, refer §C.2.1).

Table C.6: Transferability of centroid among various k on the same dataset. $k_1 \to k_2$ is to use centroids estimated by k_1 demonstrations for inference on test examples with k_2 demonstrations. Other annotations are the same as Table. C.5

	0→1	4→1	(1→1)	1→4	(4→4)
SemE.R	**+9.46** ±1.95	**+22.50** ±14.55	**(+26.14)** ±5.16	**+17.95** ±7.51	**(+38.75)** ±2.28
SemE.L	**+26.80** ±3.20	**+17.18** ±5.61	**(+26.65)** ±2.72	+10.79 ±14.86	**(+37.33)** ±3.47
AGNews	**+42.38** ±2.42	**+40.20** ±1.24	**(+41.02)** ±2.49	**+43.12** ±2.02	**(+46.66)** ±3.77
PoemS	+0.16 ±1.87	+2.12 ±6.18	**(+21.49)** ±2.54	**+8.79** ±1.84	**(+12.96)** ±1.52
Fina.P	-0.13 ±1.88	**+21.40** ±2.90	**(+16.70)** ±3.80	**+10.00** ±13.68	**(+37.30)** ±2.30

C.4 Statements

C.4.1 Author Contributions Statement

The vast majority of contributions of this work are attributed to Hakaze Cho (also called Yufeng Zhao), who provided ideas, designed and conducted experiments, collected and described data, wrote papers, and revised them.

Y.S., M.K., K.T., and A.I. participated in our discussion and provided some comments, with sequentially decreasing participation and contribution. They also helped write a non-peer-viewed seminar version of this paper in Japanese.

Naoya Inoue is their supervisor, he provides an excellent laboratory environment, funding, necessary guid-

ance, and paper revision.

C.4.2 License for Artifacts

Models. GPT 2-XL and OPT-2.7B is under the MIT license, Llama family is under its specific license.

Datasets. We list the open-source license for the datasets used in this work as follows:

- `CC-by-4.0`: `Poem Sentiment`, `SemEval 2014-Task 4 Restaurant`, `SemEval 2014-Task 4 Laptops`, `tweet_eval_emotion`, `tweet_eval_hate`, `tweet_eval_hate`
- `CC-by-SA-3.0`: `financial_phrasebank`, `GLUE-RTE`
- Unknown: `AGNews`, `rotten_tomatoes`

Consistency of Usage. Models and data are used with their original usage.

C.4.3 AI Agent Usage

AI Agents are only used for writing improving and grammar checking in this paper.

Table C.7: Classification performance (Macro F1(%)) on GPT 2-XL. mean_{std}, top-2 results are in **bold**.

GPT-2 XL		AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$16.53_{0.00}$	$9.87_{0.00}$	$12.31_{0.00}$	$8.75_{0.00}$	$48.31_{0.00}$	$19.40_{0.00}$	$37.56_{0.00}$	$21.14_{0.00}$	$25.36_{0.00}$	$34.16_{0.00}$	23.34
	Con.C	$30.04_{0.00}$	$9.87_{0.00}$	$12.31_{0.00}$	$8.11_{0.00}$	$47.46_{0.00}$	$9.68_{0.00}$	$42.97_{0.00}$	$27.01_{0.00}$	$33.14_{0.00}$	$60.75_{0.00}$	28.13
	Bat.C	$42.56_{0.00}$	$37.29_{0.00}$	**$46.21_{0.00}$**	$18.92_{0.00}$	**$51.29_{0.00}$**	$26.39_{0.00}$	**$44.45_{0.00}$**	**$42.52_{0.00}$**	$32.00_{0.00}$	**$66.96_{0.00}$**	40.86
	Dom.C	$38.00_{0.60}$	$24.76_{0.04}$	$26.58_{1.06}$	$20.52_{0.99}$	$38.00_{0.60}$	$14.34_{0.00}$	$37.56_{0.00}$	**$42.70_{1.00}$**	$30.14_{1.06}$	**$67.42_{2.19}$**	34.33
	KNN	$50.63_{0.00}$	**$39.62_{0.00}$**	$42.31_{0.00}$	$26.69_{0.00}$	**$49.57_{0.00}$**	$30.79_{0.00}$	**$44.68_{0.00}$**	$34.12_{0.00}$	$39.30_{0.00}$	$62.37_{0.00}$	42.01
	Cent.C	**$52.54_{0.00}$**	$39.53_{0.00}$	$45.37_{0.00}$	**$27.23_{0.00}$**	$44.92_{0.00}$	**$32.71_{0.00}$**	$41.93_{0.00}$	$32.15_{0.00}$	**$43.15_{0.00}$**	$63.32_{0.00}$	**42.29**
	Hidd.C	**$82.02_{0.00}$**	**$44.73_{0.00}$**	**$54.45_{0.00}$**	**$32.81_{0.00}$**	$47.36_{0.00}$	**$43.45_{0.00}$**	$42.95_{0.00}$	$36.78_{0.00}$	**$47.38_{0.00}$**	$59.83_{0.00}$	**49.18**
$k=1$	None	$20.95_{1.29}$	$36.72_{1.19}$	$31.60_{1.33}$	$21.21_{1.72}$	$49.47_{2.22}$	$22.81_{0.94}$	$37.56_{0.00}$	$31.74_{1.43}$	$30.15_{1.97}$	$36.20_{0.56}$	31.84
	Con.C	$24.15_{1.13}$	$41.88_{1.73}$	$38.92_{3.32}$	$24.82_{2.55}$	$47.64_{3.38}$	$20.70_{1.20}$	$37.56_{0.00}$	$33.06_{1.29}$	$34.93_{1.31}$	$60.21_{3.98}$	36.39
	Bat.C	$30.02_{1.49}$	**$45.04_{0.77}$**	**$41.10_{4.41}$**	$25.20_{0.48}$	**$49.58_{1.88}$**	$25.85_{2.51}$	**$48.02_{0.94}$**	**$34.92_{0.91}$**	**$35.02_{1.14}$**	**$64.72_{1.92}$**	**39.95**
	Dom.C	$22.17_{1.15}$	$44.86_{1.16}$	$41.02_{4.91}$	**$25.61_{1.27}$**	$46.81_{2.09}$	$18.92_{1.58}$	$37.56_{0.00}$	**$33.96_{0.90}$**	$34.72_{1.56}$	$62.27_{3.33}$	36.79
	KNN	**$32.14_{0.85}$**	$36.88_{2.76}$	$37.29_{3.20}$	$21.44_{1.57}$	$48.18_{2.09}$	**$27.44_{1.75}$**	$43.52_{3.73}$	$33.85_{1.08}$	**$36.23_{2.07}$**	$52.41_{1.44}$	36.94
	Cent.C	$26.74_{2.18}$	$33.00_{1.48}$	$32.02_{3.63}$	$18.75_{2.78}$	$47.82_{3.27}$	$23.82_{3.95}$	$44.07_{1.52}$	$30.68_{2.12}$	$29.58_{1.98}$	$58.41_{4.30}$	34.49
	Hidd.C	**$65.15_{1.77}$**	**$49.16_{3.43}$**	**$51.56_{2.83}$**	**$32.83_{2.32}$**	**$50.47_{1.41}$**	**$36.17_{2.74}$**	**$49.16_{1.47}$**	$33.55_{4.45}$	$44.02_{3.36}$	**$65.55_{2.47}$**	**47.76**
$k=4$	None	$21.87_{4.32}$	$33.14_{1.46}$	$41.03_{2.14}$	$20.11_{1.47}$	$40.48_{1.15}$	$17.98_{0.33}$	$38.19_{1.41}$	$29.06_{1.70}$	$28.86_{2.07}$	$33.81_{0.72}$	30.45
	Con.C	$24.22_{10.00}$	$44.76_{0.98}$	**$48.90_{2.71}$**	$21.95_{1.28}$	$36.33_{1.11}$	$24.40_{0.82}$	$37.51_{0.11}$	**$37.30_{2.49}$**	$41.65_{2.96}$	$69.57_{1.11}$	38.66
	Bat.C	$26.97_{9.22}$	$44.48_{1.67}$	$46.94_{1.72}$	$21.93_{1.24}$	$47.17_{1.96}$	**$28.86_{1.88}$**	$46.94_{2.76}$	**$36.49_{2.64}$**	**$44.82_{2.54}$**	**$71.94_{1.24}$**	**41.65**
	Dom.C	$25.30_{10.23}$	**$45.44_{1.54}$**	$47.01_{2.20}$	**$23.05_{1.05}$**	$36.79_{1.68}$	$27.11_{1.72}$	$37.79_{0.51}$	$34.67_{0.81}$	$42.54_{2.53}$	$67.00_{3.89}$	38.67
	KNN	**$33.93_{2.04}$**	$37.57_{3.54}$	$38.35_{2.35}$	$21.72_{3.05}$	**$48.55_{2.97}$**	$26.20_{1.96}$	**$48.71_{2.74}$**	$30.10_{2.12}$	$36.57_{2.56}$	$57.36_{1.72}$	37.91
	Cent.C	$32.98_{2.56}$	$37.24_{5.80}$	$32.71_{4.50}$	$18.47_{1.88}$	$45.78_{2.93}$	$24.21_{4.17}$	**$48.83_{3.57}$**	$29.75_{2.50}$	$33.30_{7.56}$	$58.81_{2.86}$	36.21
	Hidd.C	**$49.55_{3.29}$**	**$50.81_{2.16}$**	**$54.16_{3.62}$**	**$24.96_{2.56}$**	**$49.28_{2.42}$**	**$39.13_{2.80}$**	$48.48_{1.62}$	$34.43_{2.46}$	**$50.80_{3.51}$**	**$72.70_{2.19}$**	**47.43**
$k=8$	None	$19.23_{0.78}$	$32.79_{2.13}$	$33.36_{1.02}$	$17.93_{1.08}$	$37.37_{2.07}$	$15.75_{0.75}$	$37.56_{0.00}$	$25.15_{0.99}$	$26.45_{1.13}$	$33.55_{0.45}$	27.91
	Con.C	$18.38_{0.28}$	**$47.06_{3.77}$**	**$52.40_{0.96}$**	$19.70_{2.05}$	$35.98_{0.00}$	$26.80_{2.00}$	$37.56_{0.00}$	**$38.12_{3.15}$**	$41.05_{2.25}$	**$75.05_{2.12}$**	39.21
	Bat.C	$22.32_{1.22}$	$45.96_{2.08}$	$48.25_{0.79}$	**$21.20_{1.55}$**	**$45.82_{4.68}$**	**$29.23_{1.22}$**	$47.84_{1.36}$	**$38.30_{3.09}$**	**$48.78_{2.06}$**	$74.81_{1.58}$	**42.25**
	Dom.C	$21.85_{0.95}$	$45.78_{2.48}$	$49.91_{0.74}$	$20.80_{1.97}$	$35.98_{0.00}$	$28.50_{1.30}$	$37.56_{0.60}$	$36.20_{2.44}$	$47.99_{1.84}$	$69.87_{4.70}$	39.44
	KNN	**$39.81_{1.81}$**	$38.83_{4.38}$	$34.38_{3.72}$	$20.92_{2.98}$	$43.03_{5.31}$	$26.49_{3.84}$	**$49.00_{1.23}$**	$29.33_{3.74}$	$39.00_{1.91}$	$60.08_{3.64}$	38.09
	Cent.C	$41.52_{4.18}$	$37.73_{3.15}$	$32.21_{2.80}$	$14.66_{4.00}$	$45.11_{6.67}$	$21.83_{6.60}$	$48.97_{3.71}$	$25.81_{2.72}$	$31.32_{3.89}$	$55.23_{4.35}$	35.44
	Hidd.C	**$57.10_{5.54}$**	**$49.86_{2.55}$**	**$58.26_{3.15}$**	**$24.42_{1.28}$**	**$48.48_{4.48}$**	**$35.48_{5.04}$**	**$51.50_{2.18}$**	$27.46_{6.30}$	**$56.99_{1.72}$**	**$76.50_{3.14}$**	**48.60**

Table C.8: Classification performance (Macro F1(%)) on OPT-2.7B. mean_{std}, top-2 results are in **bold**.

OPT 2.7B		AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$27.67_{0.00}$	$24.62_{0.00}$	$31.07_{0.00}$	$25.00_{0.00}$	$51.19_{0.00}$	$24.43_{0.00}$	$39.31_{0.00}$	$34.08_{0.00}$	**$47.23_{0.00}$**	$40.33_{0.00}$	34.49
	Con.C	$20.72_{0.00}$	$20.77_{0.00}$	$29.33_{0.00}$	$16.51_{0.00}$	$34.63_{0.00}$	$16.25_{0.00}$	$31.37_{0.00}$	$30.65_{0.00}$	$44.88_{0.00}$	**$67.48_{0.00}$**	31.26
	Bat.C	$28.58_{0.00}$	$31.32_{0.00}$	$34.31_{0.00}$	$25.44_{0.00}$	**$53.23_{0.00}$**	$26.81_{0.00}$	$43.60_{0.00}$	$34.34_{0.00}$	$45.25_{0.00}$	**$66.51_{0.00}$**	38.94
	Dom.C	$27.55_{0.06}$	$20.53_{0.05}$	$30.90_{0.84}$	$16.72_{0.40}$	$45.07_{0.12}$	$18.01_{0.18}$	$43.30_{0.12}$	$31.11_{0.24}$	**$46.16_{0.47}$**	$66.23_{1.28}$	34.56
	KNN	$52.50_{0.00}$	$31.39_{0.00}$	**$44.16_{0.00}$**	$25.72_{0.00}$	**$52.08_{0.00}$**	$36.15_{0.00}$	$51.60_{0.00}$	**$39.45_{0.00}$**	$42.26_{0.00}$	$57.16_{0.00}$	43.25
	Cent.C	**$55.97_{0.00}$**	**$31.65_{0.00}$**	$43.23_{0.00}$	**$35.86_{0.00}$**	$41.98_{0.00}$	**$41.27_{0.00}$**	**$53.44_{0.00}$**	$35.25_{0.00}$	$37.54_{0.00}$	$58.60_{0.00}$	**43.48**
	Hidd.C	**$75.01_{0.00}$**	**$41.94_{0.00}$**	**$52.14_{0.00}$**	**$39.92_{0.00}$**	$45.64_{0.00}$	**$45.90_{0.00}$**	**$52.93_{0.00}$**	**$35.67_{0.00}$**	$43.24_{0.00}$	$61.71_{0.00}$	**49.41**
$k=1$	None	$24.16_{0.71}$	$18.76_{0.50}$	$24.98_{1.74}$	$12.23_{1.19}$	**$50.75_{3.01}$**	$20.72_{3.33}$	**$51.55_{4.28}$**	$23.64_{1.19}$	$31.40_{1.83}$	$48.94_{1.90}$	30.71
	Con.C	$22.57_{1.38}$	$20.15_{1.78}$	$24.48_{1.88}$	$13.17_{0.63}$	**$50.75_{3.01}$**	$24.30_{1.69}$	**$51.69_{4.03}$**	$23.02_{0.00}$	$28.88_{1.95}$	$65.63_{1.72}$	32.46
	Bat.C	$27.43_{1.13}$	$20.48_{0.90}$	$26.78_{1.17}$	$14.21_{1.05}$	$50.25_{3.25}$	$26.44_{1.41}$	$50.45_{1.76}$	$23.34_{2.30}$	$28.81_{0.24}$	**$70.88_{0.65}$**	33.91
	Dom.C	$24.06_{0.94}$	$19.68_{1.26}$	$24.24_{1.73}$	$13.78_{2.27}$	**$50.75_{3.01}$**	$23.58_{2.54}$	$51.54_{3.86}$	$23.14_{0.83}$	$28.81_{1.72}$	**$69.98_{1.81}$**	32.96
	KNN	$48.15_{2.50}$	**$42.35_{4.06}$**	$39.01_{5.24}$	$25.52_{1.94}$	**$53.07_{3.06}$**	$32.62_{3.02}$	$49.78_{0.58}$	**$31.90_{3.64}$**	$36.59_{3.67}$	$57.31_{4.89}$	**41.63**
	Cent.C	**$49.21_{2.81}$**	$39.46_{4.26}$	**$42.30_{4.65}$**	**$29.60_{5.30}$**	$48.48_{3.85}$	**$34.32_{2.51}$**	$50.69_{1.21}$	$31.00_{1.34}$	**$36.69_{2.28}$**	$53.70_{2.51}$	41.54
	Hidd.C	**$65.18_{2.39}$**	**$44.91_{5.14}$**	**$51.62_{2.09}$**	**$33.72_{2.24}$**	$50.40_{1.80}$	**$45.00_{2.79}$**	$49.53_{2.92}$	**$35.02_{0.47}$**	**$48.10_{3.33}$**	$57.72_{1.42}$	**48.12**
$k=4$	None	$22.91_{1.05}$	$20.84_{1.16}$	$25.44_{1.60}$	$12.46_{1.29}$	$49.70_{3.22}$	$14.83_{0.16}$	$40.68_{0.44}$	$23.78_{0.99}$	$28.62_{1.13}$	$53.73_{0.95}$	29.30
	Con.C	$22.81_{1.04}$	$20.70_{1.02}$	$27.74_{1.79}$	$12.62_{1.32}$	**$50.06_{3.86}$**	$21.92_{1.03}$	$41.21_{0.70}$	$25.86_{1.65}$	$36.74_{1.68}$	**$83.50_{1.82}$**	34.32
	Bat.C	$25.40_{1.04}$	$20.07_{1.40}$	$26.49_{0.86}$	$11.46_{1.24}$	$47.61_{2.78}$	$23.92_{0.96}$	$45.71_{1.33}$	$26.91_{1.24}$	$38.88_{1.70}$	$82.22_{1.13}$	34.87
	Dom.C	$22.32_{1.17}$	$20.71_{1.39}$	$25.96_{1.97}$	$12.64_{1.24}$	**$50.37_{3.22}$**	$21.74_{1.30}$	$41.30_{0.54}$	$26.15_{1.41}$	$38.01_{1.69}$	**$83.99_{1.45}$**	34.32
	KNN	$49.36_{2.28}$	$49.71_{5.26}$	$47.66_{3.87}$	$22.18_{1.17}$	$48.74_{3.76}$	**$32.05_{4.50}$**	**$47.35_{3.18}$**	**$30.76_{3.92}$**	**$40.78_{1.86}$**	$66.33_{4.19}$	**43.49**
	Cent.C	**$49.91_{4.46}$**	**$51.72_{2.80}$**	**$48.96_{3.00}$**	**$21.63_{3.55}$**	$47.02_{2.37}$	$29.84_{3.43}$	$46.99_{2.05}$	$25.39_{3.05}$	$40.23_{8.32}$	$65.80_{4.54}$	42.75
	Hidd.C	**$69.56_{3.62}$**	**$59.59_{1.97}$**	**$62.77_{3.08}$**	**$25.42_{0.79}$**	$47.94_{3.16}$	**$47.47_{5.86}$**	**$49.46_{1.80}$**	**$35.61_{3.45}$**	**$65.91_{2.01}$**	$79.47_{5.88}$	**54.32**
$k=8$	None	$21.52_{1.03}$	$21.24_{1.48}$	$24.11_{1.51}$	$12.84_{0.73}$	$48.05_{3.65}$	$14.54_{0.00}$	$37.54_{0.03}$	$21.85_{1.07}$	$28.47_{1.48}$	$59.86_{1.63}$	29.00
	Con.C	$22.26_{0.33}$	$23.84_{1.26}$	$30.27_{1.27}$	$13.02_{0.89}$	$47.58_{3.27}$	$23.06_{1.10}$	$37.80_{0.48}$	$25.91_{0.79}$	$37.67_{1.44}$	$83.42_{1.74}$	34.48
	Bat.C	$24.38_{1.49}$	$22.28_{1.04}$	$28.05_{1.65}$	$13.31_{0.50}$	$47.50_{2.86}$	$24.46_{0.50}$	$49.30_{1.63}$	$27.96_{1.51}$	$40.39_{1.53}$	$84.03_{1.31}$	36.16
	Dom.C	$22.38_{0.70}$	$21.33_{1.57}$	$26.63_{1.36}$	$13.18_{1.08}$	$47.48_{3.87}$	$23.63_{1.08}$	$37.59_{0.25}$	$25.25_{1.11}$	$39.27_{2.20}$	**$86.67_{1.55}$**	34.34
	KNN	$47.45_{1.64}$	$55.08_{1.89}$	$48.16_{3.38}$	$22.61_{2.20}$	**$49.56_{2.68}$**	**$34.80_{3.18}$**	$49.60_{2.26}$	**$30.77_{2.79}$**	$38.73_{3.52}$	$66.65_{5.67}$	44.34
	Cent.C	**$47.50_{4.46}$**	**$55.83_{4.46}$**	**$51.06_{8.42}$**	**$22.95_{2.22}$**	$48.22_{2.35}$	$33.57_{4.12}$	**$49.66_{2.73}$**	$29.18_{4.37}$	**$43.87_{2.57}$**	$67.49_{8.07}$	**44.93**
	Hidd.C	**$65.27_{1.64}$**	**$63.95_{2.51}$**	**$66.48_{2.53}$**	**$23.70_{1.76}$**	**$49.86_{2.86}$**	**$52.24_{2.39}$**	**$50.57_{3.28}$**	**$41.94_{6.42}$**	**$71.16_{4.13}$**	**$85.78_{4.39}$**	**57.09**

Table C.9: Classification performance (Macro F1(%)) on Llama 2-6.9B. mean_{std}, top-2 results are in **bold**.

	Llama 2	AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$23.74_{0.00}$	$44.36_{0.00}$	$32.48_{0.00}$	$16.23_{0.00}$	$35.88_{0.00}$	$17.01_{0.00}$	$46.06_{0.00}$	$30.59_{0.00}$	$19.64_{0.00}$	$32.90_{0.00}$	29.89
	Con.C	$22.76_{0.00}$	$44.50_{0.00}$	$42.83_{0.00}$	$10.53_{0.00}$	$43.88_{0.00}$	$24.06_{0.00}$	$42.42_{0.00}$	$33.32_{0.00}$	$54.92_{0.00}$	**$58.97_{0.00}$**	37.82
	Bat.C	$37.26_{0.00}$	$49.03_{0.00}$	**$54.33_{0.00}$**	$19.70_{0.00}$	**$47.34_{0.00}$**	**$31.64_{0.00}$**	**$51.06_{0.00}$**	**$39.82_{0.00}$**	**$55.76_{0.00}$**	**$63.07_{0.00}$**	**44.90**
	Dom.C	$28.11_{1.43}$	$47.88_{0.45}$	$49.74_{1.40}$	**$26.58_{1.27}$**	$39.99_{0.13}$	$23.00_{1.11}$	$37.48_{0.00}$	$33.33_{0.21}$	**$61.03_{0.76}$**	$54.96_{2.78}$	40.21
	KNN	**$40.24_{0.00}$**	$47.65_{0.00}$	$49.15_{0.00}$	$20.56_{0.00}$	**$50.93_{0.00}$**	$26.65_{0.00}$	$42.35_{0.00}$	$33.63_{0.00}$	$46.44_{0.00}$	$53.26_{0.00}$	41.09
	Cent.C	$34.36_{0.00}$	**$50.87_{0.00}$**	$46.36_{0.00}$	$17.57_{0.00}$	$42.15_{0.00}$	$25.09_{0.00}$	$42.31_{0.00}$	$28.21_{0.00}$	$43.81_{0.00}$	$49.38_{0.00}$	38.01
	Hidd.C	**$62.46_{0.00}$**	**$50.90_{0.00}$**	**$55.77_{0.00}$**	**$22.49_{0.00}$**	$46.94_{0.00}$	**$34.06_{0.00}$**	**$47.37_{0.00}$**	**$34.29_{0.00}$**	$54.84_{0.00}$	$56.37_{0.00}$	**46.49**
$k=1$	None	$13.97_{1.15}$	$51.57_{0.30}$	$51.50_{0.47}$	$22.28_{0.98}$	$36.26_{0.80}$	$27.80_{1.58}$	$40.16_{0.83}$	$30.78_{2.77}$	$27.89_{2.18}$	$55.03_{1.91}$	35.72
	Con.C	$13.29_{1.62}$	$52.36_{0.58}$	$54.12_{0.50}$	$24.16_{1.29}$	$38.53_{0.92}$	$27.77_{1.16}$	$40.63_{1.52}$	$37.39_{1.86}$	$37.23_{0.68}$	**$75.91_{2.44}$**	40.14
	Bat.C	$22.05_{1.07}$	$52.24_{0.44}$	$53.77_{1.27}$	**$24.75_{1.51}$**	**$49.77_{3.25}$**	$28.94_{0.90}$	$47.72_{1.19}$	**$40.25_{2.30}$**	$38.43_{1.91}$	$75.77_{1.37}$	43.37
	Dom.C	$11.89_{1.01}$	$51.80_{0.38}$	$53.98_{0.45}$	$24.23_{1.13}$	**$49.33_{3.82}$**	$29.17_{1.44}$	$38.23_{0.85}$	$39.02_{1.72}$	$34.95_{1.35}$	**$79.44_{0.58}$**	41.20
	KNN	**$52.50_{3.58}$**	$56.19_{5.59}$	$61.24_{2.42}$	$21.91_{0.91}$	$48.82_{1.90}$	**$30.66_{3.56}$**	**$49.65_{3.15}$**	$34.86_{1.11}$	**$40.26_{3.40}$**	$75.26_{3.34}$	**46.87**
	Cent.C	$46.17_{0.93}$	**$62.09_{3.04}$**	**$64.50_{1.76}$**	$23.29_{2.57}$	$46.66_{3.16}$	$27.80_{2.36}$	$45.63_{2.33}$	$33.10_{2.42}$	$38.99_{3.56}$	$75.16_{2.33}$	46.34
	Hidd.C	**$61.88_{0.68}$**	**$64.83_{1.85}$**	**$69.05_{2.29}$**	**$24.26_{1.37}$**	$48.32_{2.35}$	**$42.77_{2.38}$**	**$51.55_{1.27}$**	**$40.72_{1.91}$**	**$63.79_{0.93}$**	$71.00_{5.23}$	**53.82**
$k=4$	None	$9.57_{0.26}$	$53.45_{0.34}$	$53.76_{0.62}$	$19.82_{0.51}$	$35.90_{0.18}$	$29.30_{1.47}$	$37.56_{0.00}$	$36.55_{0.97}$	$37.86_{0.78}$	$69.31_{1.43}$	38.31
	Con.C	$9.35_{0.23}$	$53.49_{0.33}$	$54.21_{0.30}$	**$27.01_{1.15}$**	$37.65_{1.74}$	**$31.37_{1.07}$**	$37.56_{0.00}$	$39.34_{0.47}$	$39.28_{0.55}$	$90.86_{0.35}$	42.01
	Bat.C	$19.33_{1.51}$	$52.58_{0.64}$	$54.40_{0.69}$	$24.62_{0.99}$	$46.81_{5.22}$	$31.26_{1.58}$	$48.81_{2.93}$	$38.47_{0.33}$	$37.05_{0.60}$	$86.62_{1.15}$	44.00
	Dom.C	$9.49_{0.32}$	$52.12_{0.79}$	$54.14_{0.28}$	**$26.47_{0.87}$**	$39.59_{2.03}$	$30.75_{1.06}$	$37.56_{1.43}$	**$39.52_{0.22}$**	$34.71_{0.58}$	**$91.84_{0.88}$**	41.62
	KNN	**$66.47_{6.82}$**	$60.00_{0.92}$	$62.87_{2.54}$	$21.76_{3.20}$	$54.41_{4.95}$	$29.96_{2.19}$	$49.46_{2.22}$	$31.90_{3.97}$	**$49.09_{5.20}$**	$88.18_{1.17}$	**51.41**
	Cent.C	$60.62_{4.62}$	**$63.61_{2.06}$**	**$64.55_{3.77}$**	$21.42_{4.73}$	**$55.50_{5.61}$**	$28.41_{2.51}$	**$50.72_{3.29}$**	$21.12_{5.70}$	$42.48_{2.81}$	$89.55_{0.74}$	49.80
	Hidd.C	**$68.22_{6.82}$**	**$65.64_{1.38}$**	**$71.38_{2.00}$**	$25.47_{2.10}$	**$58.00_{4.38}$**	**$48.48_{4.32}$**	**$51.83_{3.03}$**	**$49.44_{3.20}$**	**$68.79_{9.71}$**	**$92.26_{0.18}$**	**59.95**
$k=8$	None	$8.03_{0.00}$	$53.90_{0.31}$	$54.12_{0.38}$	$20.47_{0.49}$	$35.98_{0.00}$	$30.77_{1.50}$	$37.56_{0.00}$	$38.59_{1.26}$	$39.25_{1.04}$	$73.77_{0.60}$	39.24
	Con.C	$8.15_{0.26}$	$53.72_{0.30}$	$54.26_{0.37}$	**$28.72_{1.91}$**	$35.98_{0.00}$	$31.40_{1.94}$	$37.56_{0.00}$	$39.33_{0.53}$	$38.80_{0.46}$	$91.52_{0.38}$	41.94
	Bat.C	$17.58_{1.19}$	$53.06_{0.52}$	$54.15_{0.28}$	$25.62_{0.85}$	$51.85_{3.70}$	$30.66_{1.48}$	$48.45_{2.63}$	$38.44_{0.46}$	$36.28_{0.37}$	$88.11_{0.62}$	44.42
	Dom.C	$8.03_{0.00}$	$52.26_{0.43}$	$54.26_{0.47}$	$26.50_{0.99}$	$36.44_{1.05}$	$29.28_{2.06}$	$37.56_{0.00}$	**$39.72_{0.41}$**	$35.21_{0.86}$	**$91.91_{0.49}$**	41.12
	KNN	**$69.41_{4.38}$**	$59.12_{4.16}$	$62.69_{3.74}$	$23.15_{2.09}$	**$54.86_{4.00}$**	**$34.89_{2.06}$**	$48.44_{3.62}$	$36.27_{4.36}$	**$50.27_{5.38}$**	$89.60_{1.32}$	**52.87**
	Cent.C	**$60.04_{8.07}$**	**$61.80_{1.44}$**	**$66.16_{2.05}$**	$23.17_{1.31}$	$53.64_{3.23}$	$32.16_{2.62}$	**$51.74_{1.94}$**	$30.29_{3.91}$	$36.55_{6.71}$	$90.12_{0.52}$	50.57
	Hidd.C	$59.17_{4.08}$	**$65.88_{0.84}$**	**$70.07_{2.32}$**	**$27.85_{2.56}$**	**$55.79_{3.05}$**	**$46.70_{5.37}$**	**$53.95_{2.36}$**	**$43.90_{0.48}$**	**$67.69_{2.29}$**	**$92.26_{0.63}$**	**58.33**

Table C.10: Classification performance (Macro F1(%)) on Llama 3-8B. mean_{std}, top-2 results are in **bold**.

	Llama 3	AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$19.55_{0.00}$	$27.61_{0.00}$	$24.38_{0.00}$	$11.25_{0.00}$	$43.68_{0.00}$	$26.33_{0.00}$	$46.69_{0.00}$	$30.67_{0.00}$	$44.45_{0.00}$	$61.19_{0.00}$	33.58
	Con.C	$23.20_{0.00}$	$17.16_{0.00}$	$14.55_{0.00}$	$2.33_{0.00}$	**$47.50_{0.00}$**	$21.66_{0.00}$	$33.34_{0.00}$	$29.66_{0.00}$	$28.78_{0.00}$	$60.71_{0.00}$	27.89
	Bat.C	$26.59_{0.00}$	$43.52_{0.00}$	$38.62_{0.00}$	$17.40_{0.00}$	$47.25_{0.00}$	**$32.02_{0.00}$**	**$49.12_{0.00}$**	**$38.17_{0.00}$**	**$50.32_{0.00}$**	**$67.19_{0.00}$**	41.02
	Dom.C	$19.17_{0.00}$	$25.92_{0.00}$	$24.36_{0.00}$	$3.97_{0.00}$	$32.86_{0.00}$	$26.46_{0.00}$	$39.35_{0.00}$	**$37.95_{0.00}$**	**$48.39_{0.00}$**	$59.50_{0.00}$	31.79
	KNN	**$49.70_{0.00}$**	$51.20_{0.00}$	$47.55_{0.00}$	$20.74_{0.00}$	$47.25_{0.00}$	$28.97_{0.00}$	$51.08_{0.00}$	$31.26_{0.00}$	$47.80_{0.00}$	$52.64_{0.00}$	**42.82**
	Cent.C	$39.40_{0.00}$	**$53.41_{0.00}$**	**$48.67_{0.00}$**	**$23.68_{0.00}$**	**$47.41_{0.00}$**	$27.00_{0.00}$	**$49.58_{0.00}$**	$25.19_{0.00}$	$40.38_{0.00}$	$51.14_{0.00}$	40.58
	Hidd.C	**$81.52_{0.00}$**	**$52.75_{0.00}$**	**$49.43_{0.00}$**	**$27.67_{0.00}$**	$44.53_{0.00}$	**$34.04_{0.00}$**	$44.53_{0.00}$	$37.20_{0.00}$	$47.74_{0.00}$	**$63.04_{0.00}$**	**48.25**
$k=1$	None	$19.97_{0.70}$	$53.47_{0.13}$	$53.75_{0.74}$	**$25.41_{1.06}$**	$48.60_{2.81}$	$17.02_{1.72}$	**$50.68_{2.19}$**	$28.30_{1.50}$	$39.28_{1.36}$	$81.04_{2.00}$	41.75
	Con.C	$15.57_{1.02}$	$53.37_{0.69}$	$54.42_{0.44}$	$20.37_{5.23}$	$48.60_{3.06}$	$19.27_{1.28}$	$49.71_{3.07}$	$37.32_{1.11}$	$39.07_{0.33}$	**$90.32_{0.74}$**	42.80
	Bat.C	$27.50_{1.91}$	$52.87_{0.17}$	$55.71_{0.60}$	$23.73_{0.65}$	$49.23_{2.95}$	$26.48_{0.99}$	$49.80_{2.77}$	**$39.54_{2.26}$**	$39.06_{0.76}$	$87.01_{0.79}$	45.09
	Dom.C	$12.96_{1.62}$	$52.23_{1.07}$	$54.41_{0.56}$	$19.94_{4.35}$	**$49.37_{3.03}$**	$20.24_{0.86}$	$49.73_{3.05}$	$37.86_{1.05}$	$36.40_{0.36}$	**$90.27_{0.56}$**	42.34
	KNN	**$61.64_{0.94}$**	$54.58_{3.37}$	$61.94_{1.25}$	$21.49_{1.40}$	$48.15_{2.60}$	$34.73_{2.96}$	$48.13_{2.61}$	$39.11_{1.54}$	**$43.88_{1.85}$**	$87.67_{1.78}$	**50.13**
	Cent.C	$60.16_{4.68}$	**$56.81_{5.45}$**	**$62.72_{1.87}$**	$22.74_{3.75}$	$45.77_{4.80}$	**$35.10_{3.40}$**	$44.59_{5.76}$	$35.95_{2.01}$	$38.15_{2.52}$	$88.06_{1.14}$	49.00
	Hidd.C	**$83.37_{2.76}$**	**$60.59_{5.42}$**	**$64.12_{1.73}$**	**$27.82_{1.99}$**	**$49.24_{1.88}$**	**$53.37_{3.10}$**	**$51.77_{2.59}$**	**$41.87_{1.07}$**	**$65.62_{5.94}$**	$90.17_{0.58}$	**58.79**
$k=4$	None	$9.75_{0.22}$	$53.90_{0.25}$	$54.36_{0.52}$	**$24.45_{1.23}$**	$37.09_{1.14}$	$14.54_{0.00}$	$39.42_{1.76}$	$33.23_{1.55}$	$40.28_{0.18}$	$84.80_{0.93}$	39.18
	Con.C	$9.17_{0.00}$	$53.72_{0.64}$	$54.42_{0.51}$	**$23.58_{4.38}$**	$37.20_{1.21}$	$15.56_{0.72}$	$43.43_{4.03}$	$38.31_{1.40}$	$40.17_{1.84}$	**$92.71_{0.63}$**	40.83
	Bat.C	$23.64_{0.50}$	$52.70_{0.67}$	$54.35_{0.33}$	$22.95_{1.60}$	$46.73_{5.45}$	$27.62_{1.01}$	**$51.46_{0.44}$**	**$38.65_{0.99}$**	$37.11_{0.41}$	$90.51_{1.13}$	44.57
	Dom.C	$9.17_{0.01}$	$52.51_{0.47}$	$54.35_{0.38}$	$21.85_{1.37}$	$37.52_{1.69}$	$15.94_{0.73}$	$43.01_{5.37}$	$38.04_{1.01}$	$37.76_{0.82}$	**$92.57_{0.48}$**	40.27
	KNN	**$74.94_{3.56}$**	**$54.73_{5.26}$**	**$59.53_{2.92}$**	$23.01_{3.28}$	**$49.54_{5.90}$**	**$40.01_{1.41}$**	$49.43_{3.14}$	$37.00_{4.50}$	$42.38_{2.07}$	$86.28_{1.79}$	**51.69**
	Cent.C	$73.26_{3.26}$	$54.31_{3.12}$	$61.26_{4.08}$	$21.02_{2.43}$	$45.64_{10.22}$	$38.73_{5.55}$	$50.74_{5.16}$	$37.05_{3.14}$	**$44.61_{2.96}$**	$85.45_{2.18}$	51.21
	Hidd.C	**$84.53_{1.35}$**	**$61.73_{4.56}$**	**$69.28_{1.43}$**	$23.03_{1.82}$	**$53.49_{3.61}$**	**$54.70_{2.03}$**	**$52.96_{3.84}$**	**$40.71_{2.35}$**	**$72.18_{3.19}$**	$92.18_{1.39}$	**60.48**
$k=8$	None	-	$53.77_{0.42}$	$54.82_{0.12}$	**$27.10_{1.62}$**	$35.98_{0.00}$	$14.54_{0.00}$	$37.69_{0.27}$	$35.96_{1.51}$	$42.65_{0.85}$	$88.40_{0.59}$	39.09
	Con.C	-	$53.52_{0.53}$	$54.42_{0.53}$	**$25.07_{4.81}$**	$35.98_{0.00}$	$14.54_{0.00}$	$47.99_{9.84}$	$38.74_{1.35}$	$41.19_{2.12}$	**$92.81_{1.09}$**	40.43
	Bat.C	-	$52.40_{0.72}$	$54.82_{0.35}$	$24.68_{0.67}$	**$60.38_{5.04}$**	$31.50_{0.44}$	**$55.00_{0.54}$**	$38.59_{0.78}$	$37.95_{0.35}$	$91.59_{0.53}$	44.69
	Dom.C	-	$52.03_{0.35}$	$54.11_{0.66}$	$21.95_{0.83}$	$35.98_{0.00}$	$14.67_{0.22}$	$45.15_{9.11}$	$38.61_{1.13}$	$37.30_{1.39}$	$93.11_{0.74}$	39.29
	KNN	-	$54.47_{2.60}$	**$58.17_{2.84}$**	$21.49_{1.69}$	$50.75_{4.92}$	**$47.22_{2.50}$**	$51.75_{2.65}$	**$41.66_{3.19}$**	**$46.47_{4.53}$**	$84.53_{1.91}$	**45.65**
	Cent.C	-	**$56.03_{3.55}$**	$49.78_{5.61}$	$22.15_{1.36}$	$48.30_{9.65}$	$45.41_{6.52}$	$48.34_{4.57}$	$38.65_{1.60}$	$39.58_{7.11}$	$85.23_{2.05}$	43.35
	Hidd.C	-	**$63.34_{1.61}$**	**$67.48_{1.00}$**	$23.84_{1.22}$	**$60.89_{3.38}$**	**$56.67_{7.36}$**	**$53.45_{4.92}$**	**$39.74_{2.66}$**	**$79.28_{4.45}$**	**$91.64_{1.59}$**	**53.63**

Table C.11: Classification performance (Macro F1(%)) on Llama 2-13B. mean_{std}, top-2 results are in **bold**.

	Llama 2	AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$9.61_{0.00}$	$47.67_{0.00}$	$51.12_{0.00}$	$8.01_{0.00}$	$36.04_{0.00}$	$31.92_{0.00}$	$37.56_{0.00}$	$46.19_{0.00}$	$30.94_{0.00}$	$54.98_{0.00}$	35.40
	Con.C	$22.36_{0.00}$	$48.44_{0.00}$	$\mathbf{57.96}_{0.00}$	$15.71_{0.00}$	$41.45_{0.00}$	$23.57_{0.00}$	$37.56_{0.00}$	$39.59_{0.00}$	$44.97_{0.00}$	$42.61_{0.00}$	37.42
	Bat.C	$30.84_{0.00}$	$\mathbf{52.16}_{0.00}$	$\mathbf{61.45}_{0.00}$	$20.50_{0.00}$	$43.54_{0.00}$	$\mathbf{37.88}_{0.00}$	$43.73_{0.00}$	$\mathbf{47.31}_{0.00}$	$\mathbf{50.83}_{0.00}$	$\mathbf{61.26}_{0.00}$	**44.95**
	Dom.C	$23.21_{0.00}$	$46.54_{0.00}$	$51.51_{0.00}$	$8.88_{0.00}$	$41.87_{0.00}$	$29.11_{0.00}$	$37.56_{0.00}$	$\mathbf{47.22}_{0.00}$	$31.98_{0.00}$	$\mathbf{60.59}_{0.00}$	37.85
	KNN	$\mathbf{46.44}_{0.00}$	$42.06_{0.00}$	$43.18_{0.00}$	$\mathbf{24.03}_{0.00}$	$\mathbf{50.33}_{0.00}$	$28.36_{0.00}$	$\mathbf{55.76}_{0.00}$	$35.27_{0.00}$	$45.74_{0.00}$	$51.95_{0.00}$	42.31
	Cent.C	$43.48_{0.00}$	$44.64_{0.00}$	$41.05_{0.00}$	$21.23_{0.00}$	$34.02_{0.00}$	$30.41_{0.00}$	$49.02_{0.00}$	$30.44_{0.00}$	$43.17_{0.00}$	$52.75_{0.00}$	39.02
	Hidd.C	$\mathbf{69.78}_{0.00}$	$\mathbf{54.08}_{0.00}$	$50.67_{0.00}$	$\mathbf{33.01}_{0.00}$	$\mathbf{54.41}_{0.00}$	$\mathbf{41.23}_{0.00}$	$\mathbf{51.14}_{0.00}$	$35.45_{0.00}$	$\mathbf{48.87}_{0.00}$	$57.61_{0.00}$	**49.63**
$k=1$	None	$9.17_{0.00}$	$50.67_{0.80}$	$53.51_{0.08}$	$15.49_{0.67}$	$35.11_{0.00}$	$33.79_{1.09}$	$37.56_{0.00}$	$37.11_{0.78}$	$34.73_{0.90}$	$81.61_{0.84}$	38.88
	Con.C	$9.17_{0.00}$	$53.38_{0.30}$	$\mathbf{54.04}_{0.20}$	$\mathbf{25.20}_{0.64}$	$35.11_{0.00}$	$33.61_{0.46}$	$37.52_{0.05}$	$\mathbf{38.00}_{0.53}$	$35.73_{0.58}$	$\mathbf{87.40}_{1.46}$	40.92
	Bat.C	$24.81_{2.72}$	$49.21_{0.87}$	$53.72_{0.14}$	$23.39_{0.33}$	$47.24_{1.16}$	$\mathbf{35.68}_{0.19}$	$\mathbf{51.66}_{1.75}$	$36.21_{0.81}$	$34.35_{0.39}$	$85.99_{1.19}$	44.23
	Dom.C	$9.55_{0.38}$	$51.02_{0.55}$	$52.92_{0.29}$	$19.66_{1.43}$	$35.11_{0.00}$	$32.55_{2.39}$	$37.56_{0.00}$	$38.43_{0.72}$	$35.55_{0.68}$	$\mathbf{86.42}_{1.75}$	39.88
	KNN	$\mathbf{58.35}_{3.83}$	$\mathbf{54.98}_{5.17}$	$48.86_{0.44}$	$24.03_{1.43}$	$\mathbf{47.50}_{2.83}$	$32.11_{1.50}$	$50.95_{3.28}$	$33.33_{4.95}$	$\mathbf{37.93}_{3.72}$	$82.19_{1.11}$	**47.02**
	Cent.C	$51.96_{0.88}$	$49.83_{4.34}$	$46.51_{2.10}$	$25.20_{2.94}$	$43.15_{3.01}$	$27.11_{1.80}$	$47.81_{6.69}$	$31.46_{6.15}$	$31.22_{4.39}$	$82.31_{1.10}$	43.65
	Hidd.C	$\mathbf{76.45}_{5.43}$	$\mathbf{64.70}_{1.97}$	$\mathbf{65.70}_{0.16}$	$\mathbf{30.22}_{3.11}$	$\mathbf{49.42}_{1.78}$	$\mathbf{48.97}_{0.32}$	$\mathbf{52.81}_{4.66}$	$\mathbf{41.05}_{4.32}$	$\mathbf{59.29}_{4.43}$	$84.02_{0.88}$	**57.26**
$k=4$	None	$9.17_{0.00}$	$49.99_{0.32}$	$53.69_{0.44}$	$19.29_{0.86}$	$35.11_{0.00}$	$\mathbf{36.23}_{2.22}$	$37.56_{0.00}$	$37.63_{0.58}$	$36.02_{0.42}$	$91.82_{0.41}$	40.65
	Con.C	$9.37_{0.22}$	$53.12_{0.62}$	$\mathbf{54.25}_{0.26}$	$\mathbf{26.98}_{1.34}$	$35.11_{0.00}$	$34.64_{2.69}$	$37.56_{0.00}$	$37.64_{1.28}$	$36.38_{0.15}$	$\mathbf{92.52}_{0.76}$	41.76
	Bat.C	$19.64_{1.36}$	$49.33_{0.12}$	$53.69_{0.13}$	$23.93_{0.41}$	$46.56_{3.27}$	$34.29_{1.84}$	$46.95_{1.79}$	$35.92_{0.83}$	$34.14_{0.46}$	$91.68_{0.52}$	43.61
	Dom.C	$9.56_{0.31}$	$49.61_{0.38}$	$53.79_{0.35}$	$16.85_{0.80}$	$35.11_{0.00}$	$33.62_{1.59}$	$37.56_{0.00}$	$\mathbf{38.20}_{1.38}$	$35.95_{0.35}$	$\mathbf{92.46}_{0.83}$	40.27
	KNN	$\mathbf{76.62}_{0.69}$	$\mathbf{53.18}_{1.01}$	$53.38_{4.35}$	$22.73_{1.17}$	$47.94_{1.65}$	$35.09_{5.00}$	$\mathbf{50.17}_{1.01}$	$38.26_{1.25}$	$40.40_{2.69}$	$83.18_{3.32}$	**50.09**
	Cent.C	$74.72_{7.17}$	$52.52_{2.90}$	$48.20_{3.83}$	$22.08_{2.73}$	$\mathbf{48.59}_{2.55}$	$28.15_{3.24}$	$49.53_{1.53}$	$29.35_{4.83}$	$\mathbf{43.69}_{5.11}$	$82.73_{1.48}$	47.96
	Hidd.C	$\mathbf{80.76}_{4.75}$	$\mathbf{66.91}_{1.22}$	$\mathbf{68.44}_{2.54}$	$\mathbf{25.39}_{1.27}$	$\mathbf{52.69}_{3.96}$	$\mathbf{53.41}_{3.40}$	$\mathbf{52.29}_{4.93}$	$\mathbf{44.47}_{4.05}$	$\mathbf{59.48}_{1.42}$	$91.36_{0.79}$	**59.52**
$k=8$	None	$9.17_{0.00}$	$50.01_{0.25}$	$53.82_{0.37}$	$19.54_{0.37}$	$35.11_{0.00}$	$36.93_{1.76}$	$37.56_{0.00}$	$36.64_{0.75}$	$36.49_{0.04}$	$\mathbf{92.80}_{0.63}$	40.81
	Con.C	$9.17_{0.00}$	$\mathbf{52.89}_{0.18}$	$\mathbf{54.31}_{0.29}$	$\mathbf{29.11}_{0.49}$	$35.11_{0.00}$	$36.44_{1.99}$	$37.56_{0.00}$	$38.24_{0.79}$	$38.10_{0.70}$	$92.72_{0.85}$	42.36
	Bat.C	$17.94_{0.26}$	$48.95_{0.69}$	$53.64_{0.45}$	$24.35_{0.44}$	$\mathbf{52.35}_{3.27}$	$34.94_{2.46}$	$47.18_{3.05}$	$36.11_{0.40}$	$35.04_{0.09}$	$92.23_{0.64}$	44.27
	Dom.C	$9.17_{0.00}$	$50.55_{0.37}$	$53.95_{0.41}$	$18.12_{1.11}$	$35.11_{0.00}$	$34.14_{2.53}$	$37.56_{0.00}$	$38.43_{0.17}$	$35.70_{0.27}$	$\mathbf{92.91}_{0.56}$	40.56
	KNN	$\mathbf{79.57}_{4.10}$	$52.03_{2.71}$	$51.68_{8.67}$	$23.48_{0.62}$	$50.35_{3.07}$	$\mathbf{39.90}_{6.24}$	$\mathbf{53.54}_{4.10}$	$\mathbf{41.95}_{2.23}$	$50.11_{3.56}$	$86.47_{1.81}$	**52.91**
	Cent.C	$76.52_{2.63}$	$51.46_{5.58}$	$42.86_{5.16}$	$22.13_{4.60}$	$49.28_{0.85}$	$34.97_{5.39}$	$53.09_{4.32}$	$33.61_{7.13}$	$\mathbf{50.29}_{2.71}$	$84.48_{2.15}$	49.87
	Hidd.C	$\mathbf{80.47}_{2.80}$	$\mathbf{66.49}_{2.19}$	$\mathbf{62.69}_{8.43}$	$\mathbf{25.58}_{3.44}$	$\mathbf{53.02}_{4.91}$	$\mathbf{50.88}_{5.15}$	$\mathbf{57.19}_{7.28}$	$\mathbf{41.99}_{3.69}$	$\mathbf{69.04}_{5.48}$	$91.31_{0.25}$	**59.87**

Table C.12: Classification performance (Macro F1(%)) on Llama 2-34B. mean_{std}, top-2 results are in **bold**.

	Llama 2	AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
$k=0$	None	$18.28_{0.00}$	$50.09_{0.00}$	$51.30_{0.00}$	$15.50_{0.00}$	$39.75_{0.00}$	$24.91_{0.00}$	$40.14_{0.00}$	$31.05_{0.00}$	$34.67_{0.00}$	**$71.67_{0.00}$**	37.73
	Con.C	$23.06_{0.00}$	$50.75_{0.00}$	**$58.71_{0.00}$**	$8.81_{0.00}$	$35.11_{0.00}$	$30.96_{0.00}$	$44.55_{0.00}$	$34.93_{0.00}$	$48.68_{0.00}$	$69.32_{0.00}$	40.49
	Bat.C	$26.52_{0.00}$	**$54.51_{0.00}$**	**$60.19_{0.00}$**	$18.91_{0.00}$	$48.47_{0.00}$	$31.62_{0.00}$	$45.90_{0.00}$	**$40.37_{0.00}$**	**$53.45_{0.00}$**	$70.69_{0.00}$	**45.06**
	Dom.C	$22.58_{0.00}$	$52.91_{0.00}$	$55.96_{0.00}$	$24.06_{0.00}$	$47.61_{0.00}$	$31.18_{0.00}$	$37.30_{0.00}$	$36.45_{0.00}$	$50.42_{0.00}$	**$71.77_{0.00}$**	43.02
	KNN	$31.79_{0.00}$	$45.85_{0.00}$	$45.03_{0.00}$	**$27.38_{0.00}$**	**$53.04_{0.00}$**	$31.64_{0.00}$	**$55.29_{0.00}$**	$34.10_{0.00}$	$46.57_{0.00}$	$57.60_{0.00}$	**42.83**
	Cent.C	**$35.28_{0.00}$**	$50.83_{0.00}$	$51.72_{0.00}$	$23.44_{0.00}$	$51.15_{0.00}$	**$32.02_{0.00}$**	$53.12_{0.00}$	$35.33_{0.00}$	$44.25_{0.00}$	$48.47_{0.00}$	42.56
	Hidd.C	**$65.31_{0.00}$**	**$56.22_{0.00}$**	$54.83_{0.00}$	**$31.82_{0.00}$**	**$54.76_{0.00}$**	**$50.49_{0.00}$**	**$65.71_{0.00}$**	**$45.46_{0.00}$**	**$69.22_{0.00}$**	$65.14_{0.00}$	**55.90**
$k=1$	None	$9.59_{0.03}$	$53.48_{0.49}$	$54.97_{0.04}$	$22.51_{0.39}$	$35.53_{0.18}$	**$33.32_{1.18}$**	$37.56_{0.38}$	$36.38_{1.24}$	$35.41_{0.18}$	$81.78_{0.31}$	40.05
	Con.C	$10.38_{0.91}$	$53.97_{0.80}$	$54.97_{0.55}$	$25.44_{0.20}$	$35.11_{2.42}$	$32.92_{0.56}$	$37.56_{0.00}$	$37.08_{2.18}$	$38.73_{0.28}$	**$85.60_{0.25}$**	41.17
	Bat.C	$21.41_{0.28}$	$51.86_{0.43}$	$54.81_{0.10}$	$23.47_{1.50}$	$52.97_{1.18}$	$33.02_{1.00}$	$48.06_{1.46}$	$37.07_{0.67}$	$35.23_{0.21}$	$84.63_{0.44}$	44.25
	Dom.C	$10.00_{0.29}$	$53.46_{0.90}$	$54.96_{0.11}$	**$29.23_{2.93}$**	$37.32_{0.40}$	$32.70_{0.82}$	$37.56_{0.00}$	$37.21_{1.47}$	$39.96_{0.16}$	**$85.80_{1.51}$**	41.82
	KNN	**$65.38_{1.25}$**	**$58.91_{1.13}$**	**$61.28_{2.68}$**	$29.09_{4.68}$	**$62.63_{3.40}$**	$30.21_{1.64}$	$54.69_{0.85}$	**$40.78_{2.03}$**	**$68.78_{1.13}$**	$68.77_{3.17}$	**54.05**
	Cent.C	$63.03_{3.16}$	$55.45_{2.61}$	$54.41_{4.82}$	$24.23_{1.29}$	$62.47_{0.04}$	$31.16_{0.39}$	**$56.13_{2.05}$**	$30.55_{3.73}$	$55.08_{3.23}$	$58.61_{3.26}$	49.11
	Hidd.C	**$79.39_{0.48}$**	**$67.23_{2.91}$**	**$69.03_{0.53}$**	**$33.12_{0.35}$**	**$74.89_{1.33}$**	**$52.50_{4.09}$**	**$61.88_{0.91}$**	**$46.63_{1.04}$**	**$76.83_{2.01}$**	$73.14_{0.06}$	**63.46**
$k=4$	None	$9.59_{0.55}$	$53.48_{0.39}$	$54.97_{0.64}$	$22.51_{0.69}$	$35.53_{0.06}$	**$33.32_{3.35}$**	$37.56_{0.00}$	$36.38_{1.92}$	$35.41_{0.51}$	$81.78_{0.55}$	40.05
	Con.C	$10.38_{0.01}$	$53.97_{0.16}$	$54.97_{0.61}$	$25.44_{0.06}$	$35.11_{0.00}$	$32.92_{3.78}$	$37.56_{0.00}$	$37.08_{0.97}$	$38.73_{0.55}$	**$85.60_{2.37}$**	41.17
	Bat.C	$21.41_{1.08}$	$51.86_{0.34}$	$54.81_{0.44}$	$23.47_{0.33}$	$52.97_{0.10}$	$33.02_{4.05}$	$48.06_{2.65}$	$37.07_{1.89}$	$35.23_{0.77}$	$84.63_{1.97}$	44.25
	Dom.C	$10.00_{0.55}$	$53.46_{0.36}$	$54.96_{0.66}$	**$29.23_{0.90}$**	$37.32_{1.28}$	$32.70_{4.18}$	$37.56_{0.00}$	$37.21_{0.80}$	$39.96_{0.04}$	**$85.80_{2.09}$**	41.82
	KNN	**$65.38_{3.35}$**	**$58.91_{0.83}$**	**$61.28_{0.90}$**	$29.09_{2.60}$	**$62.63_{2.71}$**	$30.21_{3.04}$	$54.69_{2.18}$	**$40.78_{2.16}$**	**$68.78_{1.44}$**	$68.77_{8.22}$	**54.05**
	Cent.C	$63.03_{10.02}$	$55.45_{0.33}$	$54.41_{2.47}$	$24.23_{0.35}$	$62.47_{6.66}$	$31.16_{2.87}$	**$56.13_{0.93}$**	$30.55_{0.73}$	$55.08_{1.63}$	$58.61_{1.20}$	49.11
	Hidd.C	**$79.39_{3.49}$**	**$67.23_{1.13}$**	**$69.03_{9.36}$**	**$33.12_{2.59}$**	**$74.89_{1.75}$**	**$52.50_{2.17}$**	**$61.88_{1.74}$**	**$46.63_{5.67}$**	**$76.83_{9.16}$**	$73.14_{11.06}$	**63.46**
$k=8$	None	$9.18_{0.01}$	$53.56_{0.99}$	$55.06_{0.70}$	$26.48_{0.16}$	$35.11_{0.00}$	$35.06_{0.43}$	$37.56_{0.00}$	$37.93_{0.81}$	$36.12_{0.27}$	$88.65_{2.13}$	41.47
	Con.C	$11.00_{1.59}$	$52.04_{0.51}$	$54.55_{0.11}$	$27.01_{0.23}$	$35.11_{0.00}$	$34.60_{0.17}$	$37.56_{0.00}$	$38.81_{0.29}$	$38.68_{0.74}$	**$89.87_{0.41}$**	41.92
	Bat.C	$18.25_{0.02}$	$52.17_{0.44}$	$54.91_{0.11}$	$25.89_{0.60}$	**$66.77_{0.63}$**	$35.08_{0.61}$	$57.00_{0.26}$	$36.73_{0.86}$	$35.85_{0.25}$	$90.17_{1.13}$	47.28
	Dom.C	$9.39_{0.30}$	$53.02_{1.61}$	$55.18_{0.52}$	**$32.92_{1.16}$**	$35.11_{0.00}$	$33.10_{2.08}$	$37.56_{0.00}$	$37.79_{0.23}$	$41.86_{1.38}$	**$92.08_{0.41}$**	42.80
	KNN	**$63.23_{1.12}$**	**$60.79_{0.34}$**	**$61.09_{4.12}$**	$30.84_{0.06}$	$63.67_{1.20}$	**$37.53_{0.81}$**	$55.26_{5.26}$	**$42.30_{4.48}$**	**$80.93_{0.92}$**	$78.58_{0.41}$	**57.42**
	Cent.C	$57.73_{6.27}$	$57.02_{0.78}$	$54.73_{5.85}$	$26.35_{2.44}$	$62.27_{5.14}$	$35.94_{1.68}$	**$59.23_{6.62}$**	$38.25_{6.66}$	$58.83_{10.95}$	$74.64_{0.80}$	52.50
	Hidd.C	**$83.00_{0.52}$**	**$68.71_{0.42}$**	**$77.09_{0.85}$**	**$31.59_{8.39}$**	**$78.91_{0.40}$**	**$60.97_{0.90}$**	**$64.20_{1.65}$**	**$50.52_{0.85}$**	**$91.98_{0.17}$**	$84.78_{3.82}$	**69.17**

Appendix D

Supplementary Materials for Chapter 6

D.1 Detailed Experiment Implementation

D.1.1 Model & Dataset Details

Models. All the models in this chapter are loaded from `huggingface`. In detail, we list the `huggingface` repository name to keep the repeatability of this chapter, as shown in Table D.2.

Dataset Split. As also described by Cho & Inoue (2025), we randomly sample 1024 data samples from the original dataset to build the inputs for training, and sample $4096 + 512$ (especially, $512 + 512$ for FP dataset, $3192 + 512$ for TEH dataset) data samples for the demonstrations+queries for the testing, respectively.

Demonstration Sampling. To generate the training examples of k demonstrations, we randomly sample $(k + 1)$ data examples from the aforementioned 1024 data, and concatenate them into the inputs, with the prompt templates shown in §D.1.2. To generate the testing examples, for each query in the 512 samples, we sample two sequences of demonstrations from the 4096 data samples, and concatenate them into testing inputs, 2 for one query sample, so that 1024 for one dataset.

Table D.2: `Huggingface` repository name for models used in this work.

Model	Repository
GPT 2-L	`openai-community/gpt2-large`
GPT 2-XL	`openai-community/gpt2-xl`
Falcon 3	`tiiuae/Falcon3-7B-Base`
Llama 3-8B	`meta-llama/Meta-Llama-3-8B`
DeepSeek-R1	`deepseek-ai/DeepSeek-R1-Distill-Qwen-14B`
Qwen 2.5	`Qwen/Qwen2.5-32B`
SimpleScaling s1.1	`simplescaling/s1.1-32B`
Llama 3-43B	`chargoddard/llama3-42b-v0`
Llama 3-56B	`nyunai/nyun-c2-llama3-56B`

D.1.2 Prompt Templates

We utilize the default prompt templates of STAICC, as shown in Table D.1. For the sake of simplicity, we reduce the label tokens into one token, as also shown in Table D.1.

D.1.3 Details of PID Algorithm

On each model update step $t > 2$ (i.e., when the gradients from all the samples of the t-th pseudo batch (of n_b data samples) are propagated), we calculated the identified induction head numbers from the induction head filter described in §6.3 and Fig. 6.1 averaged on the n_b data samples as $\bar{n}_t$. Given the similar averaged induction head number on the previous time step $(t - 1)$ as $\bar{n}_{t-1}$, we can calculate the updated A_t term[33] by standard PID algorithm as:

$$\begin{aligned} A_t &= C_p \left(\bar{n}_t - \bar{n}_{t-1}\right) \\ &+ C_i \left(\sum_{i=2}^{t} \bar{n}_i - \bar{n}_{i-1}\right) \\ &+ C_d \left(\bar{n}_t - 2\bar{n}_{t-1} + \bar{n}_{t-2}\right) \\ &+ A_{t-1}, \end{aligned} \tag{D.1}$$

where the $C_p = 0.03$, $C_i = 0.005$, and $C_d = 0.005$ are hyperparameters. By such calculation, we implement a feedback control to stabilize the number of induction heads among training steps.

[33] Remind that for the sake of simplicity, we only control A, given the findings of A and B are antagonistic, as shown in §6.5.3.

Table D.1: Prompt templates used in this work.

Dataset	Prompt Template (Unit)	Label Tokens
SST-2	`sentence: [input sentence] sentiment: [label token] \n`	negative, positive
MR	`review: [input sentence] sentiment: [label token] \n`	negative, positive
FP	`sentence: [input sentence] sentiment: [label token] \n`	negative, neutral, positive
SST-5	`sentence: [input sentence] sentiment: [label token] \n`	poor, bad, neutral, good, great
TREC	`question: [input sentence] target: [label token] \n`	short, entity, description, person, location, number
SUBJ	`review: [input sentence] subjectiveness: [label token] \n`	objective, subjective
TEE	`tweet: [input sentence] emotion: [label token] \n`	anger, joy, positive, sad
TEH	`tweet: [input sentence] hate speech: [label token] \n`	normal, hate

D.1.4 Experiment Protocol of Unseen Label

In Fig. 6.3, we examine that ABFT model can utilize the demonstration with wrong label to improve the ICL performance. Here we introduce the experiment protocol.

First, we train a Llama 3-8B on TREC (a 6-way classification dataset) with the ABFT method. Then, to test the trained ABFT model on the unseen label condition, we build special test inputs: for each query with label l^*, we choose $k = 4$ demonstrations with label $l \neq l^*$, and utilize the standard template shown in Table D.1 to build the inputs, then test the accuracy. Notice that during the training, no special sampling for the inputs is conducted, i.e., the training is not under the unseen label setting, so that such an experiment protocol also confirms the generalization of ABFT methods on a different distribution. Moreover, we repeat the induction head visualization shown in Fig. 6.4 on the unseen label condition in Fig. D.1, where the induction heads in the ABFT model are almost fully suppressed but with considerable inference accuracy, which implies a new inference mechanism.

D.1.5 Experiment Protocol of Stability against Prompting

In Table 6.3, we test whether the prediction of ICL is stable against various **(1)** prompt templates and **(2)** demonstration sampling, on the STAICC-DIAG (Cho & Inoue, 2025) benchmark, whose method is described briefly below.

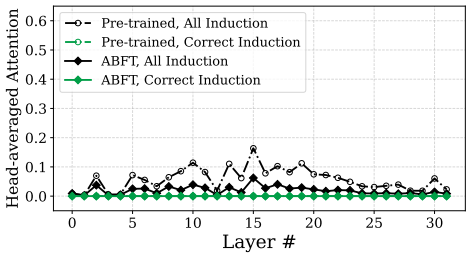


Figure D.1: Visualization of induction attention scores on unseen label settings (Llama 3-8B, TREC).

Method. To test the prediction robustness against prompt templates / demonstration sampling, we repeat several predictions for each query on various prompt templates / demonstration sampling, and calculate the ratio of the maximum consistent group (e.g., we get 6 positive and 3 negative predictions on one query, then the ratio is $\max(6, 3)/(6 + 3) = 2/3$). The robustness metrics are the average value of the whole dataset. Refer to Cho & Inoue (2025) for the detailed implementation. Notice that only the consistency is tested in these experiments, without observing the accuracy.

Result. The robustness metrics among prompt templates / demonstration sampling averaged on all 8 datasets before and after ABFT are shown in Table 6.3, where both terms of the robustness are significantly improved after ABFT, suggesting that ABFT stabilizes ICL for various contexts, providing higher efficiency on prompt designing. Also, given the results with mitigating prediction sensitivity and bias against prompt templates / demonstration sampling, which is consistent with the objective of output calibration (Zhao et al., 2021; Fei et al., 2023; Han et al., 2023c; Zhou et al., 2024a; Jiang et al., 2023; Cho et al., 2025c), ABFT can be regarded as an implicit calibration inside the LLM.

D.2 Parameter Shift after ABFT against Layers

We utilize the Frobenius norm to visualize the shifting distance of the parameter matrix θ before and after ABFT (θ') as $\|\theta - \theta'\|_2$. The results are shown in Fig. D.8, D.9, D.10, where, although each model exhibits its own pattern in terms of distance across layer numbers, certain layers consistently show significantly lower distances within every model.

Moreover, even though the early layers accumulate more gradients (since the gradients from each later layer propagate backward to them), the peak of the shifting distance typically appears in the middle to later layers.

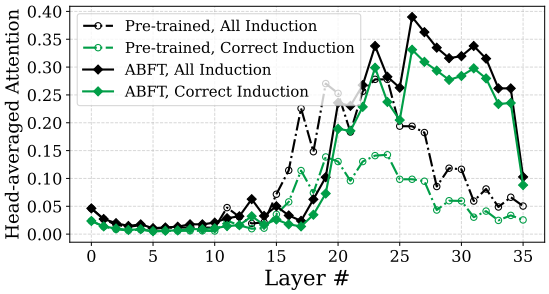


Figure D.3: Augmentation results of Fig. 6.4 on GPT 2-L.

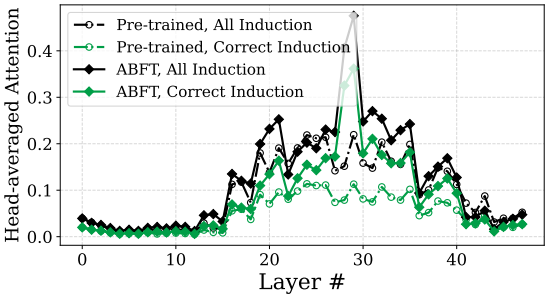


Figure D.4: Augmentation results of Fig. 6.4 on GPT 2-XL.

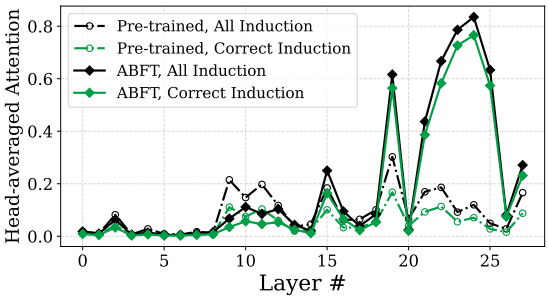


Figure D.5: Augmentation results of Fig. 6.4 on Falcon 3-7B.

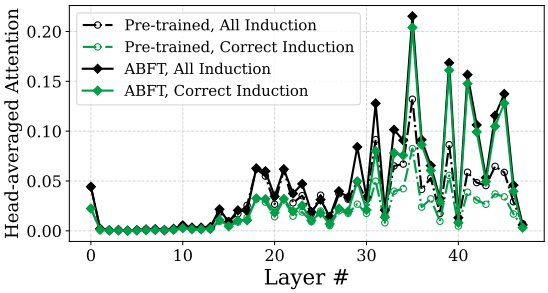


Figure D.6: Augmentation results of Fig. 6.4 on Llama 3-42B.

This observation is consistent with previous works on Induction Heads (Cho et al., 2025a).

D.3 Augmentation Results

D.3.1 Augmentation Results for Data Efficiency (Fig. 6.2)

We repeat the data efficiency experiments shown in Fig. 6.2 on Qwen 2.5-32B, as shown in Fig. D.2. The results are globally consistent with Fig. 6.2.

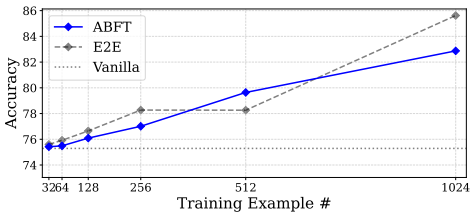


Figure D.2: Accuracy against training set size as a metric of data efficiency, for Qwen 2.5-32B.

D.3.2 Augmentation Results for Number of Induction Heads against Training Processing (Fig. 6.5)

We repeat the visualization of the number of induction heads against the training processing under various settings on Llama 3-8B and Falcon 3-7B as shown in Fig. D.11 and D.12. The results are globally consistent with Fig. 6.5.

Moreover, we visualize the number of induction heads on only standard settings, as shown in Fig. D.13-D.19 for reference.

D.3.3 Augmentation Results for Attention Visualization (Fig. 6.4 and 6.8)

As shown in Fig. D.7, we visualize the attention score on the last token of the given input example in the validation set on Llama 3-8B, and repeat this visualization on more input cases for Fig. D.7 and Fig. 6.8 in Fig. D.20. Moreover, we expand the Fig. D.7 towards more layers in Fig. D.21, and Fig. 6.8 in Fig. D.22. We observe that ABFT significantly modifies the attention distribution in the middle layers, while in the early and late layers, neither ABFT nor E2E has a substantial impact on attention scores.

Moreover, we repeat the attention score visualization similar to Fig. 6.4 on more models and SST-2, as shown

in Fig. D.3, D.4, D.5, and D.6.

D.4 Statements

Author Contributions Statement. Hakaze Cho, also known as Yufeng Zhao, handled the entire workload in this work. He provided ideas, designed / conducted experiments, collected / described data, and wrote / revised the paper. Peng Luo suggested the PID algorithm, which is an essential part of this work, and provided some technical support for it. Naoya Inoue, as the supervisor, provided valuable feedback, revisions, and financial support for this work. M.K. and R.K. made minor contributions that placed them on the borderline of authorship criteria: they participated in discussions and offered comments, some of which were incorporated.

License for Artifacts. Models and datasets used in this chapter are used in their original usage, and are open-sourced with the following license:

- `mit`: GPT 2-L, GPT 2-XL, DeepSeek-R1
- Individual License or Unknown: Falcon 3, Llama 3-8B, Llama 3-43B, Llama 3-56B, SST-5, MR, TREC, SUBJ, TEE, TEH
- `Apache 2.0`: Qwen 2.5, SimpleScaling s1.1
- `cc-by-sa-3.0`: SST-2, FP,

AI Agent Usage. AI Agents are used and only used for writing improvement in this work.

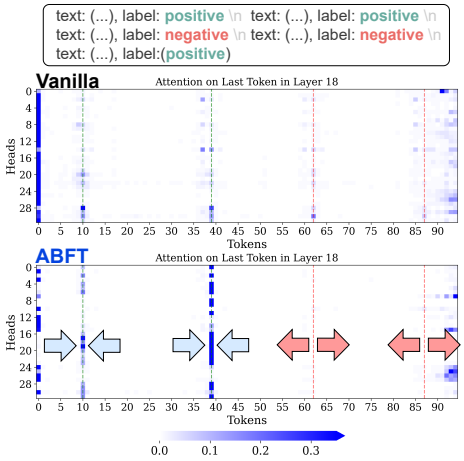


Figure D.7: Attention score visualization on the last token of ICL input of every attention head (vertical axis) towards each token. Label tokens and their contents are marked with dotted lines. Refer to Appendix D.3.3 for more examples and layers. **ABFT successfully focuses attention scores to correct labels**.

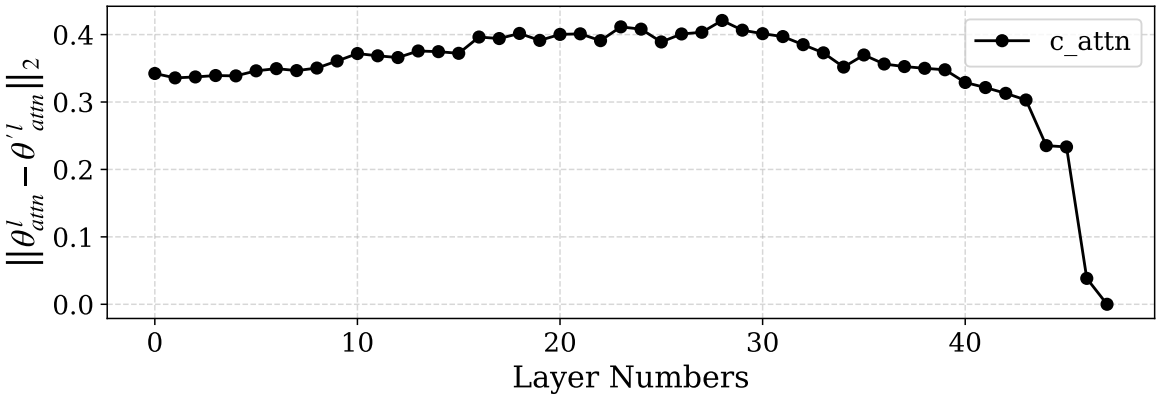


Figure D.8: Shifting distance before and after ABFT on the `c_attn` matrix of GPT 2-XL and SST-2.

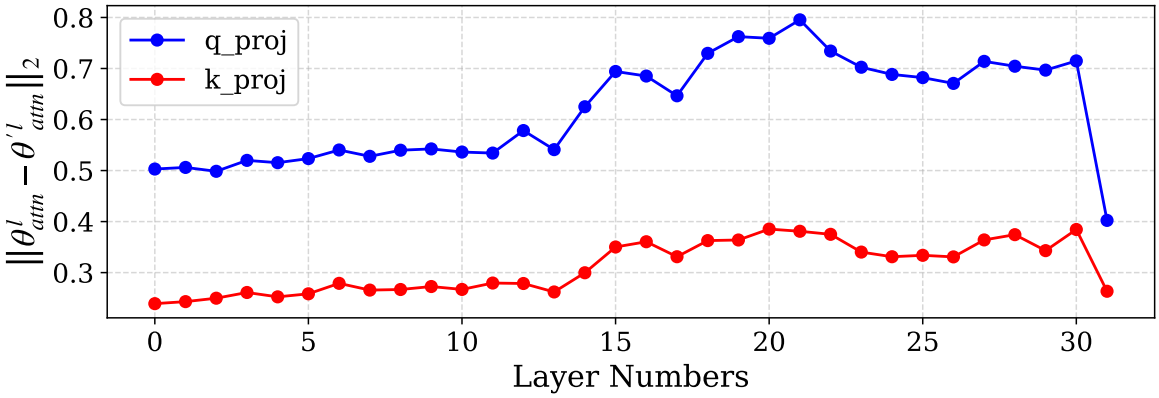


Figure D.9: Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama 3-8B and SST-2.

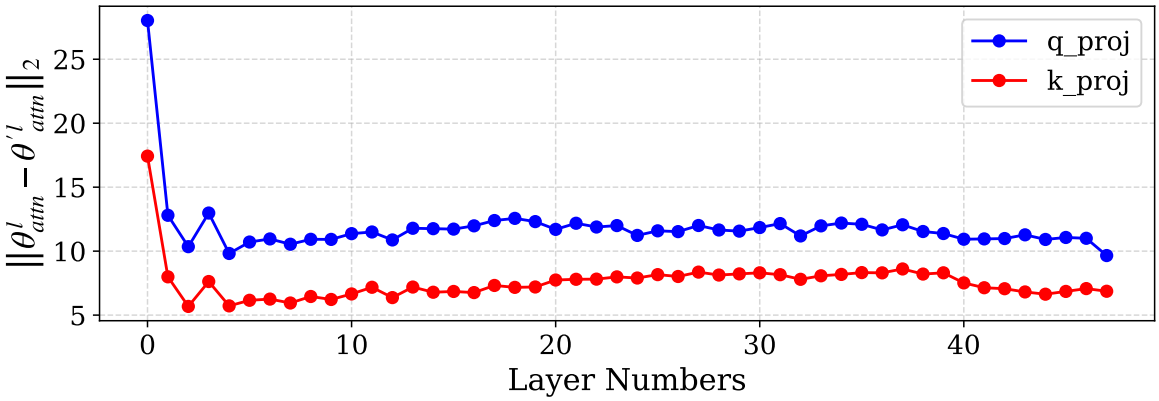
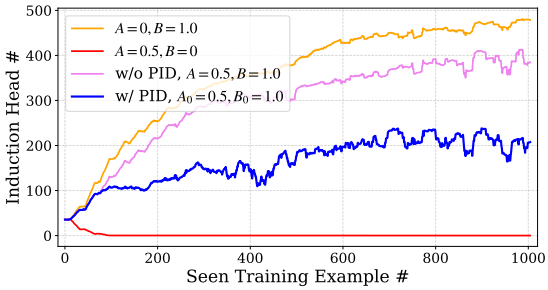
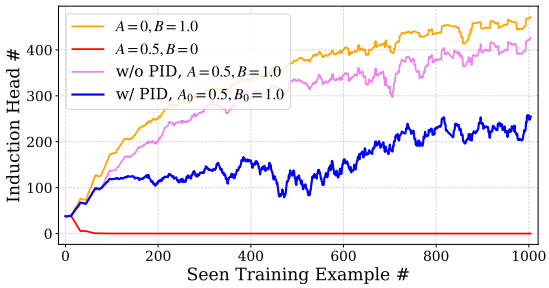


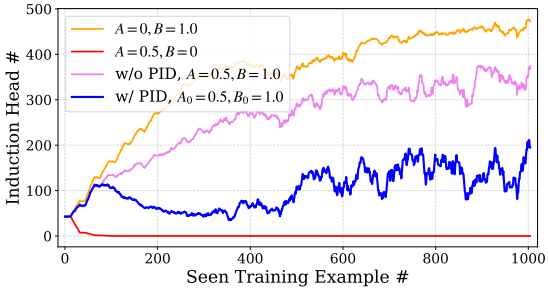
Figure D.10: Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama 3-42B and SST-2.



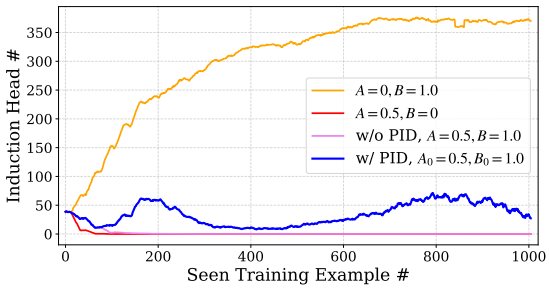
Llama 3-8B SST2 (Fig. 6.5)



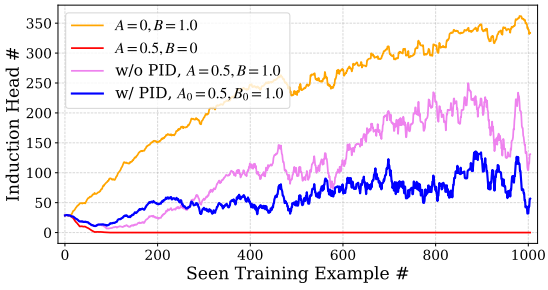
Llama 3-8B MR



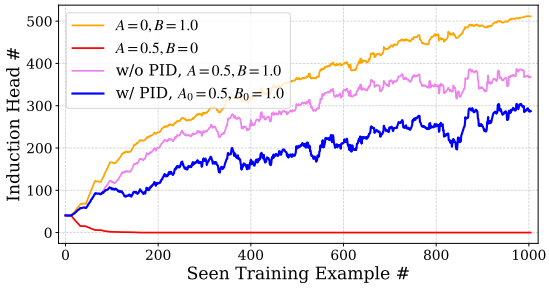
Llama 3-8B FP



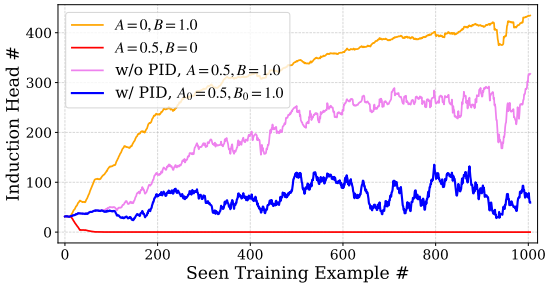
Llama 3-8B SST5



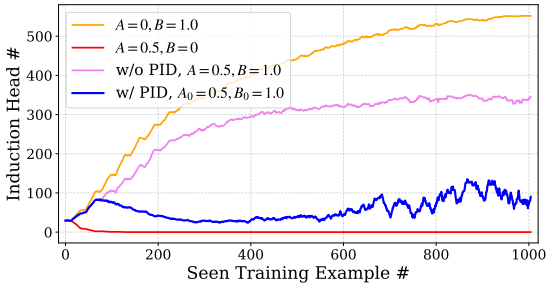
Llama 3-8B TREC



Llama 3-8B SUBJ

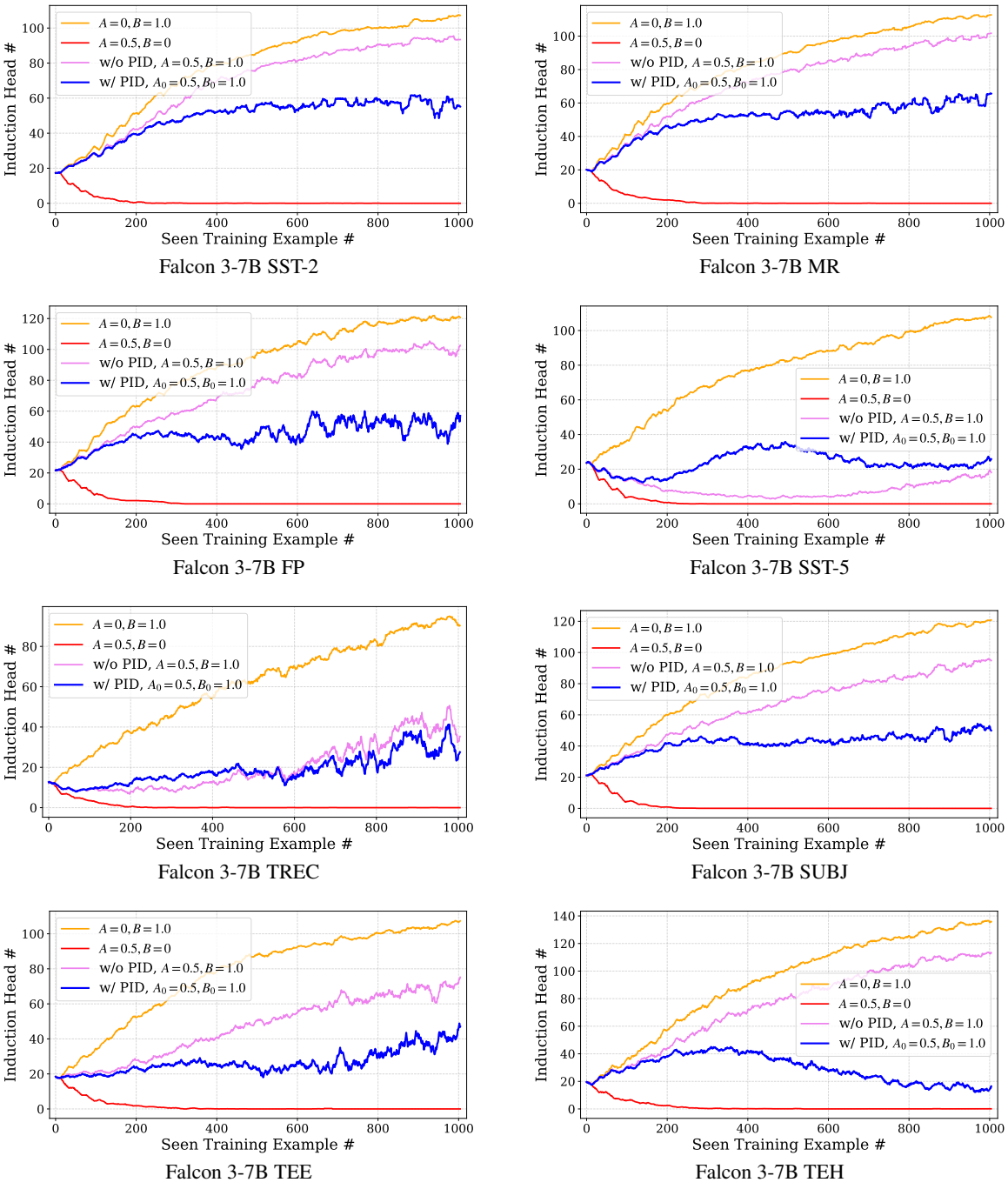


Llama 3-8B TEE



Llama 3-8B TEH

Figure D.11: Induction head numbers along training dynamics on Llama 3-8B and all 8 datasets.



Falcon 3-7B SST-2

Falcon 3-7B MR

Falcon 3-7B FP

Falcon 3-7B SST-5

Falcon 3-7B TREC

Falcon 3-7B SUBJ

Falcon 3-7B TEE

Falcon 3-7B TEH

Figure D.12: Induction head numbers along training dynamics on Falcon 3-7B and all 8 datasets.

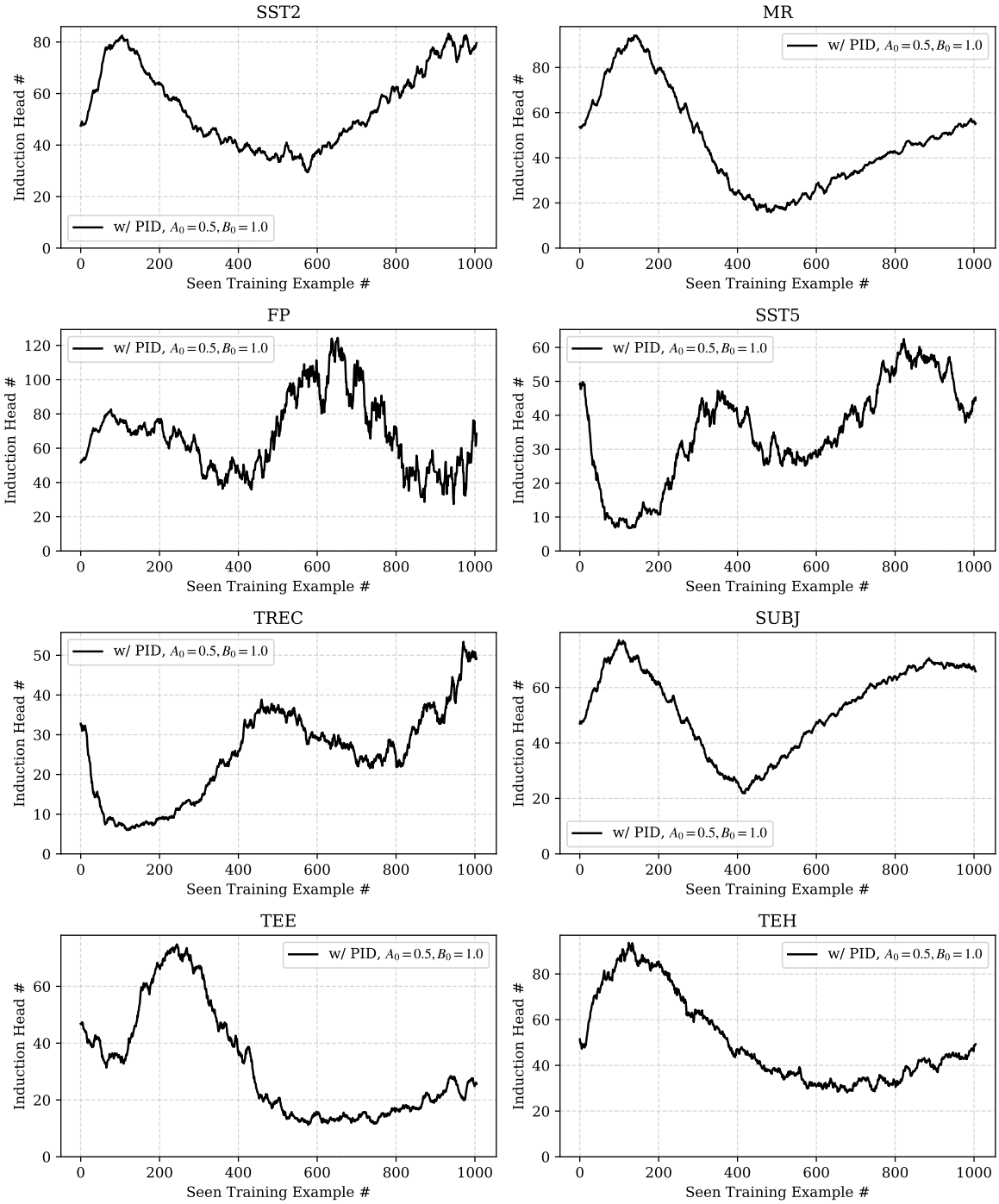


Figure D.13: Induction head numbers along training dynamics on GPT 2-Large and all 8 datasets.

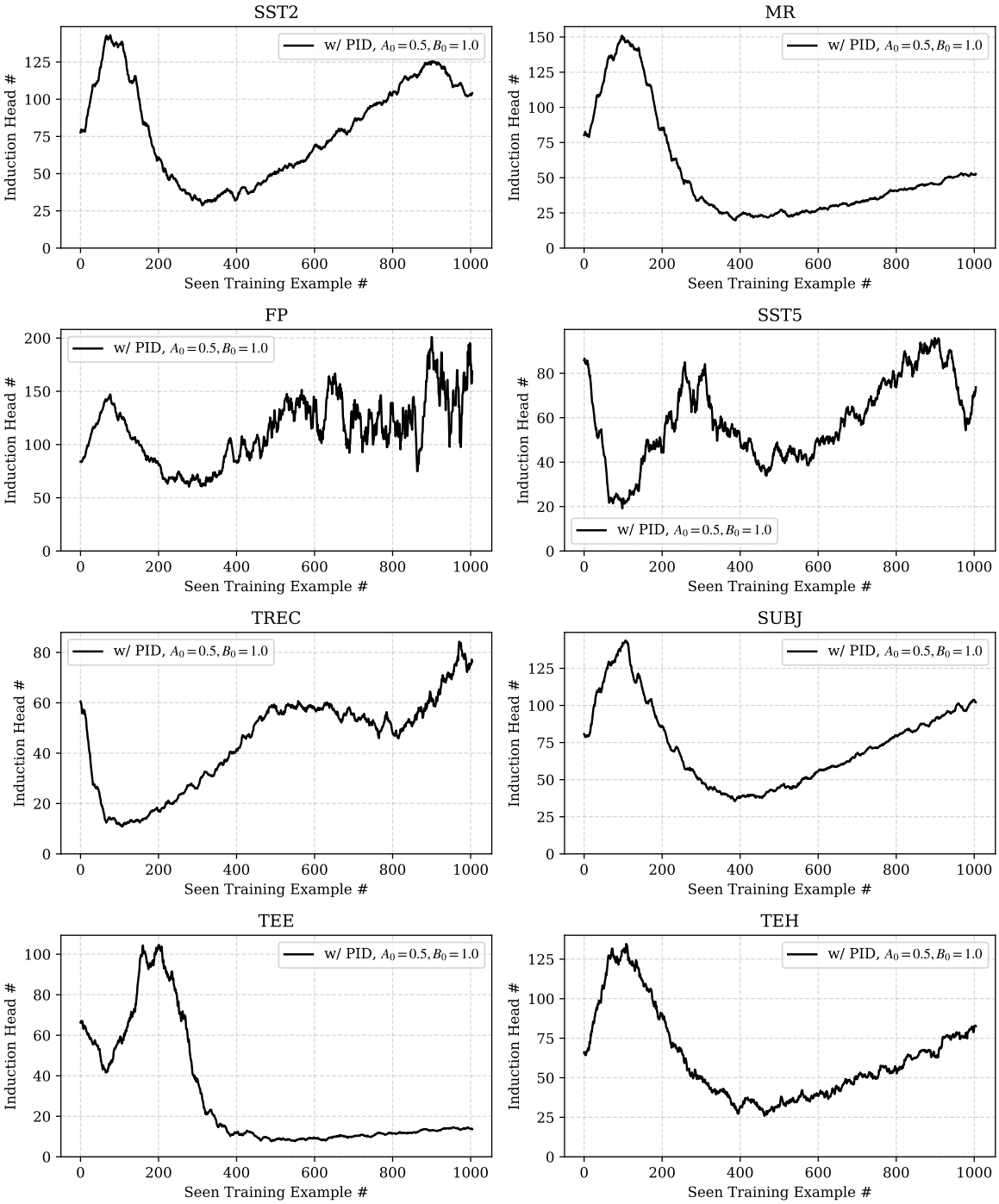


Figure D.14: Induction head numbers along training dynamics on GPT 2-XL and all 8 datasets.

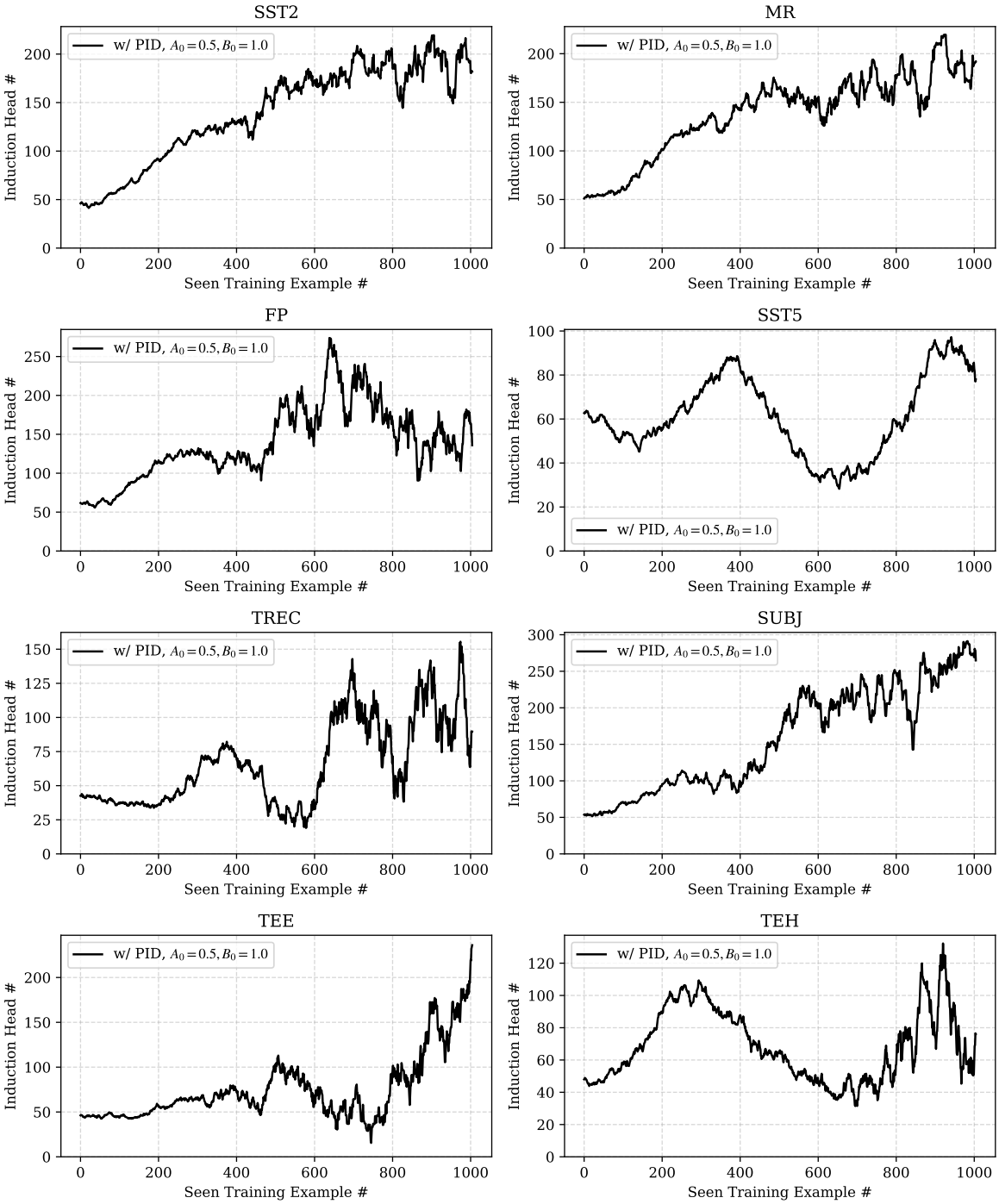


Figure D.15: Induction head numbers along training dynamics on DeepSeek-R1 and all 8 datasets.

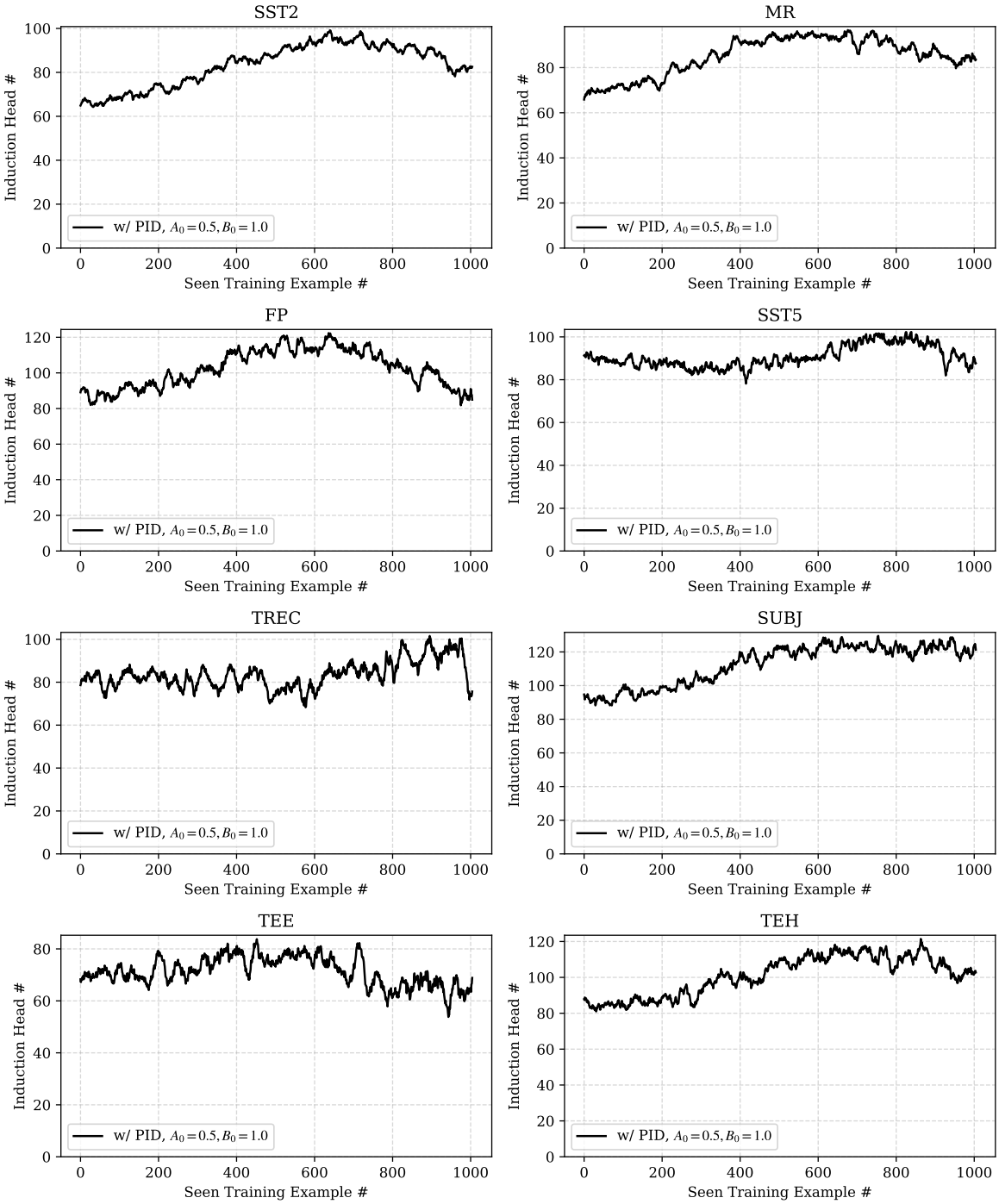


Figure D.16: Induction head numbers along training dynamics on Qwen 2.5-32B and all 8 datasets.

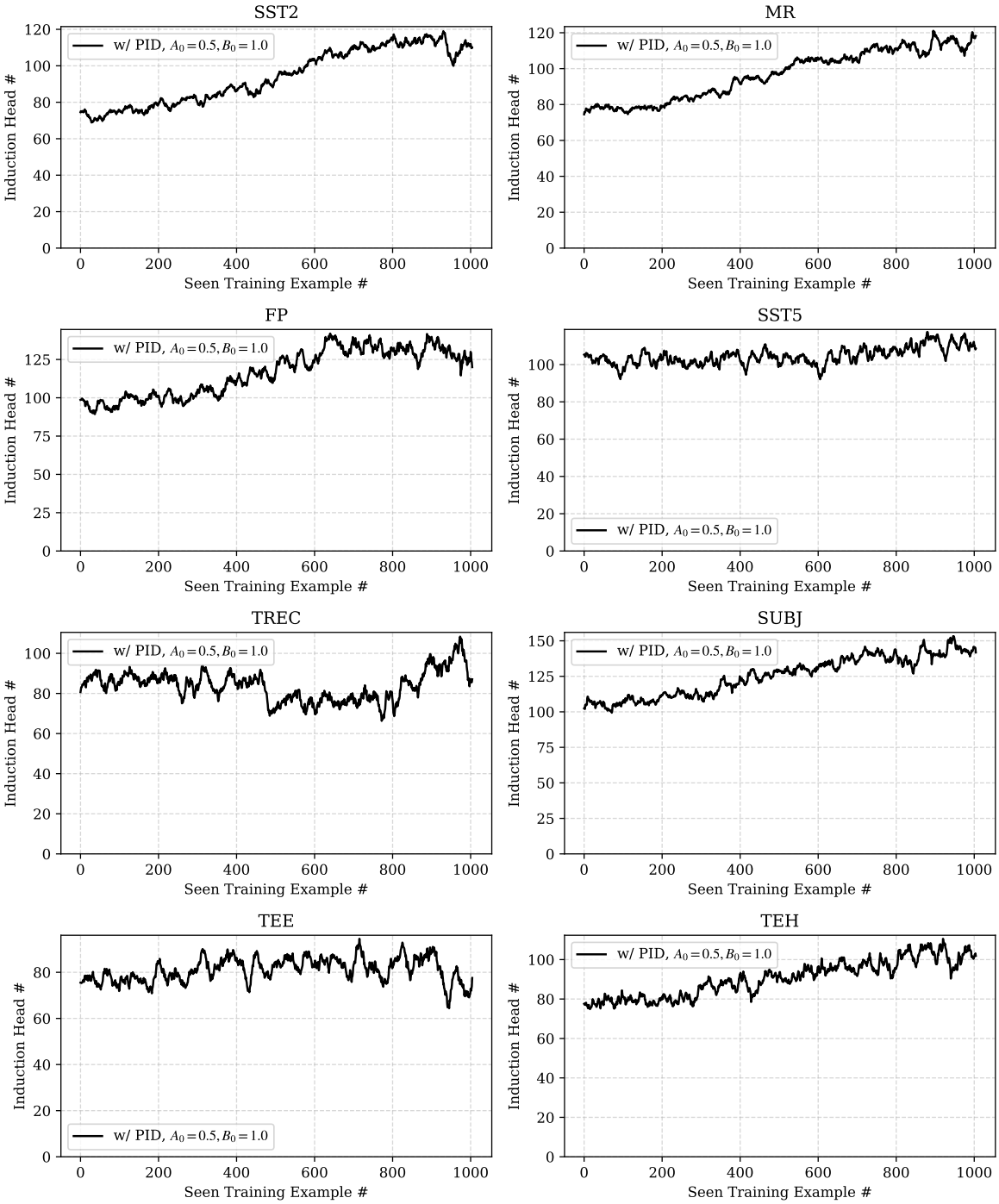


Figure D.17: Induction head numbers along training dynamics on SimpleScaling s1.1 and all 8 datasets.

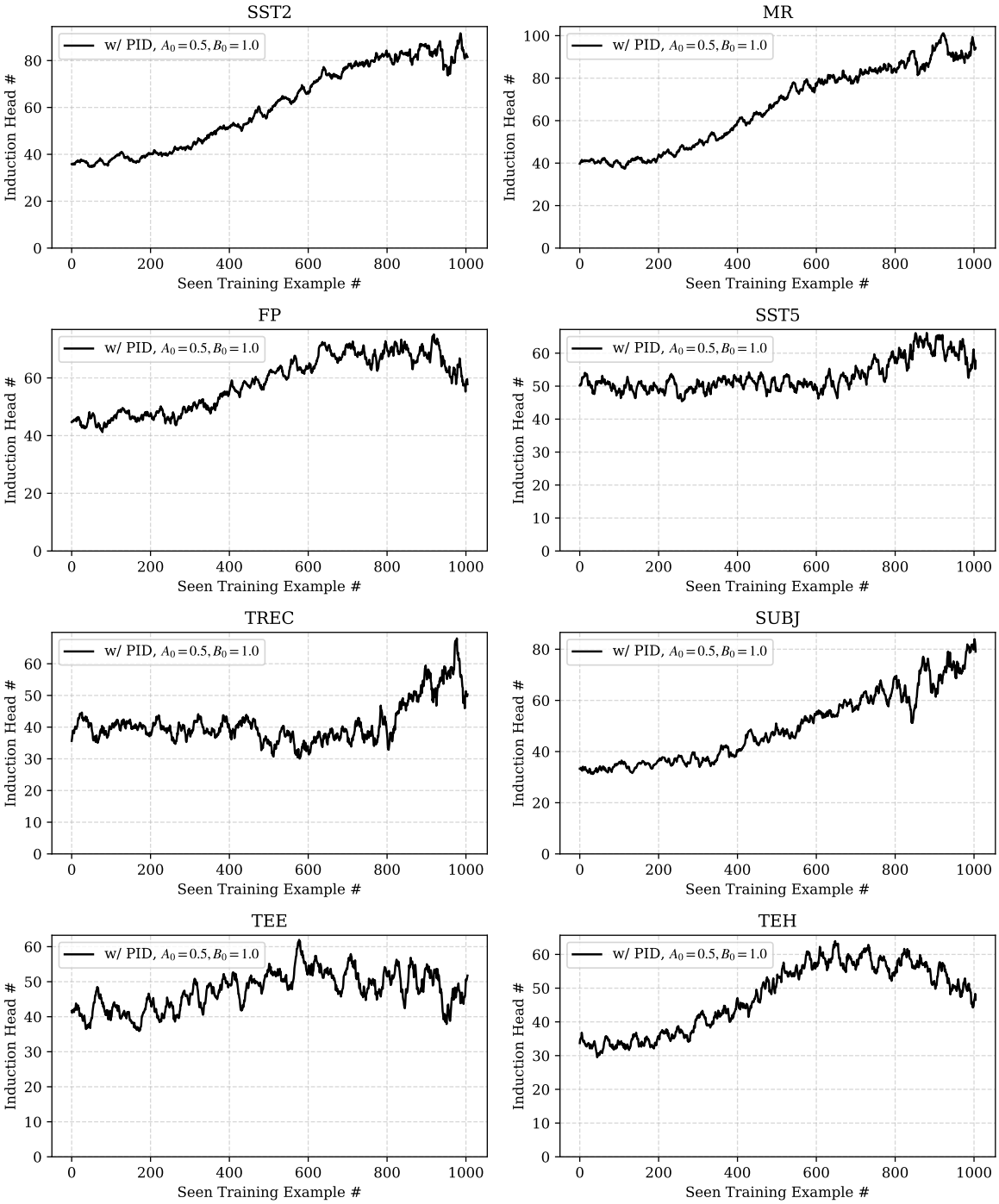


Figure D.18: Induction head numbers along training dynamics on Llama 3-42B and all 8 datasets.

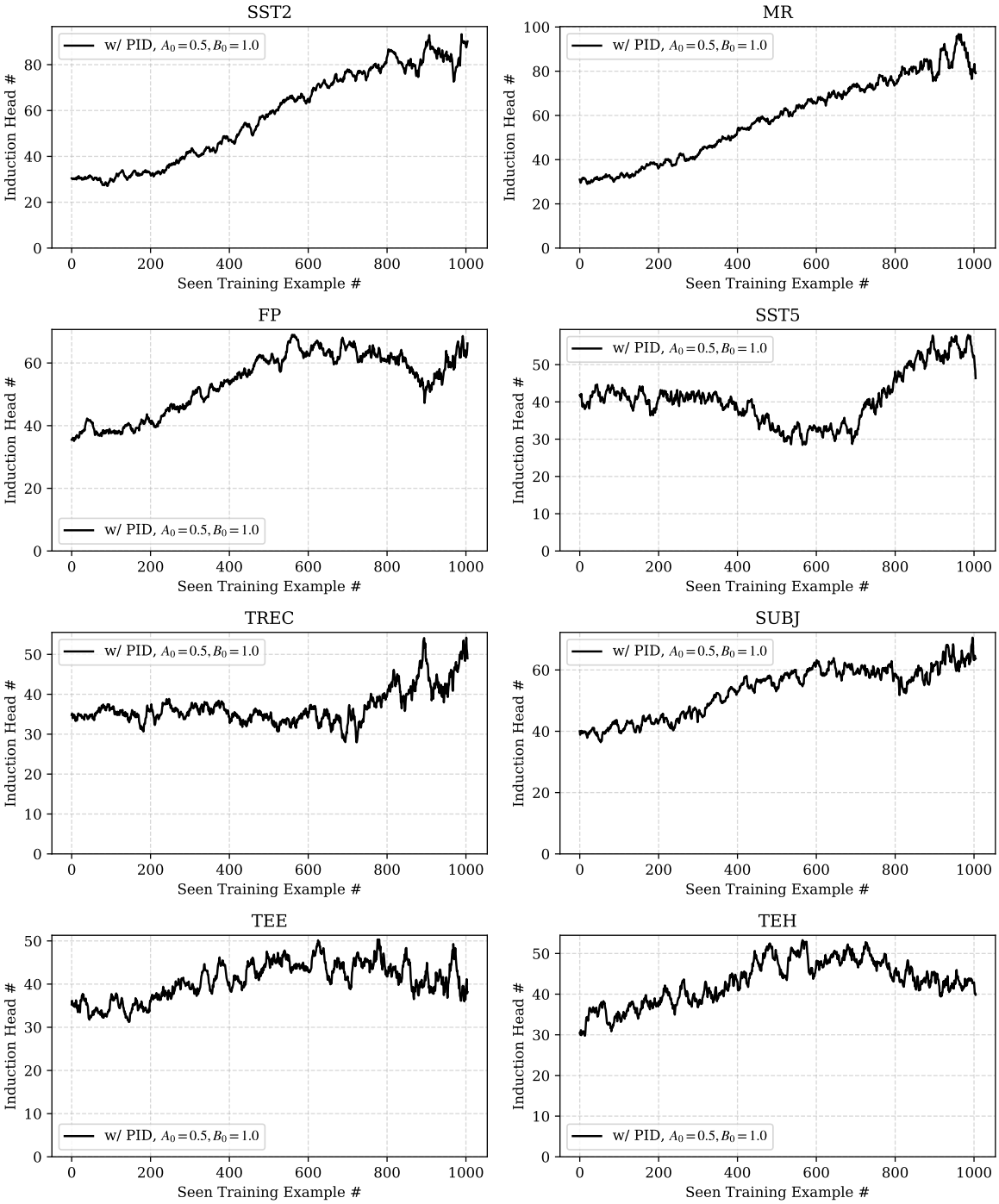


Figure D.19: Induction head numbers along training dynamics on Llama 3-56B and all 8 datasets.

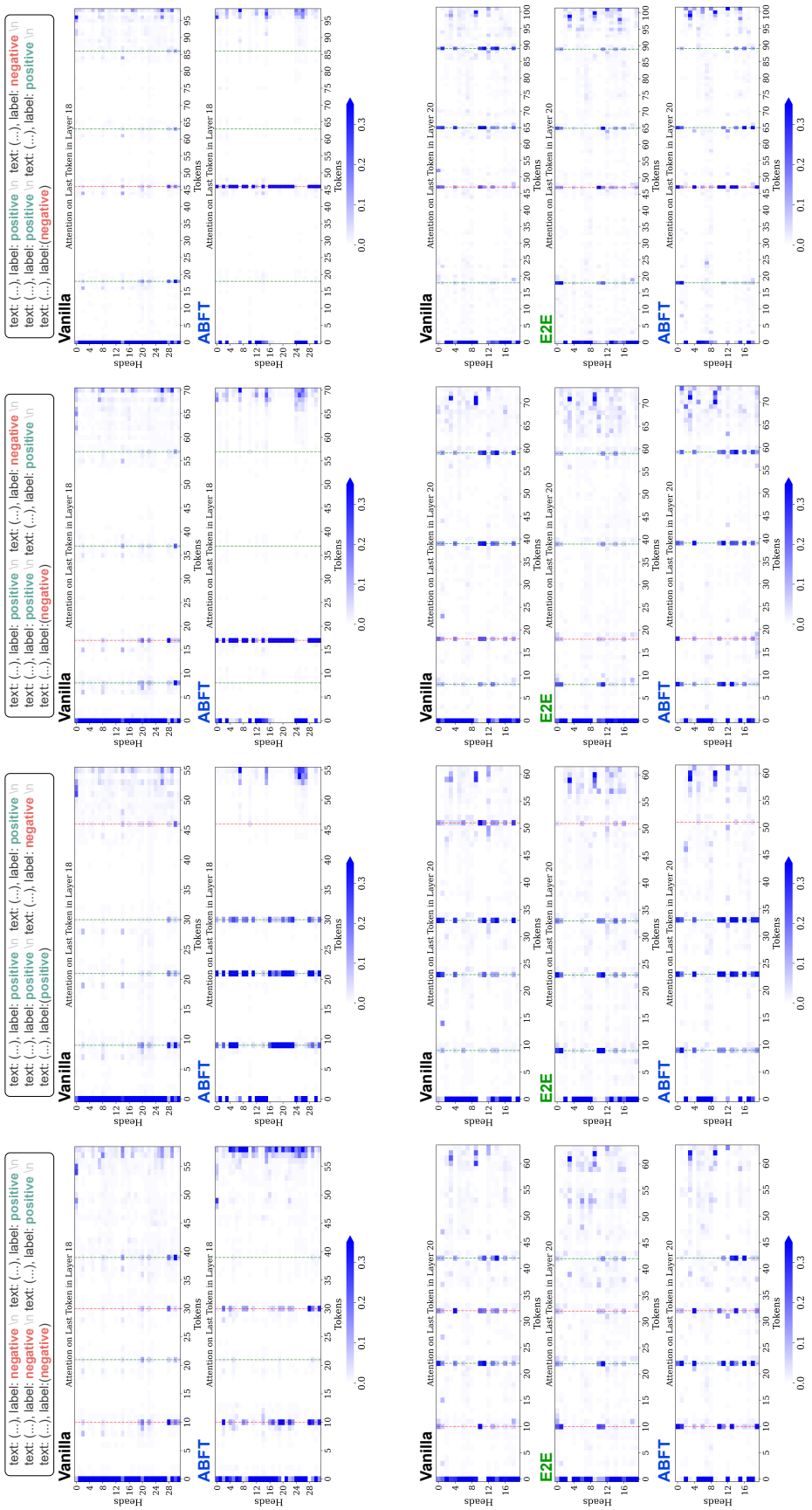


Figure D.20: Augmentation results (4 cases) for attention visualization. **Upper**: results for Fig. D.7, **lower**: results for Fig. 6.8.

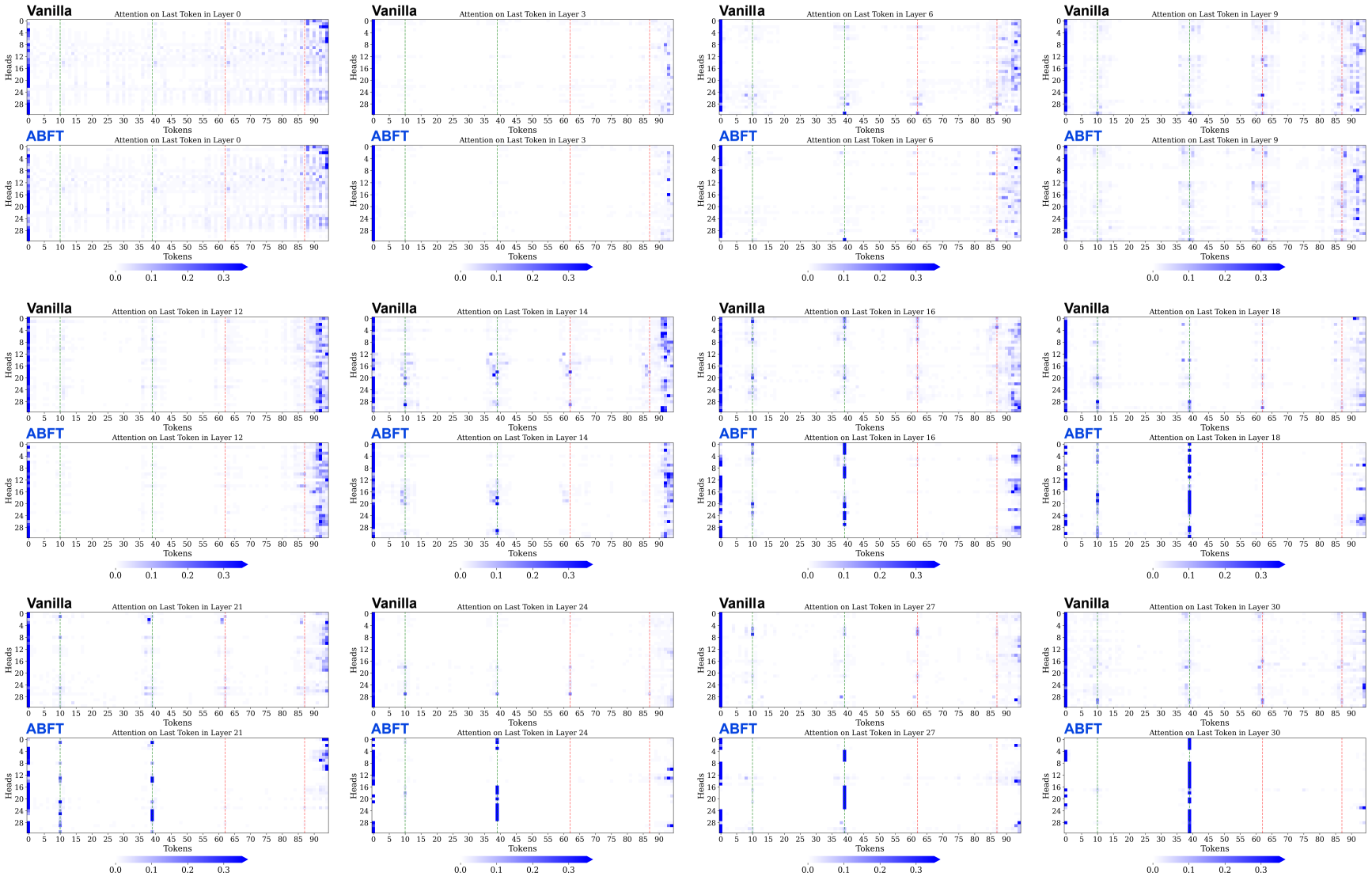


Figure D.21: Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. D.7.

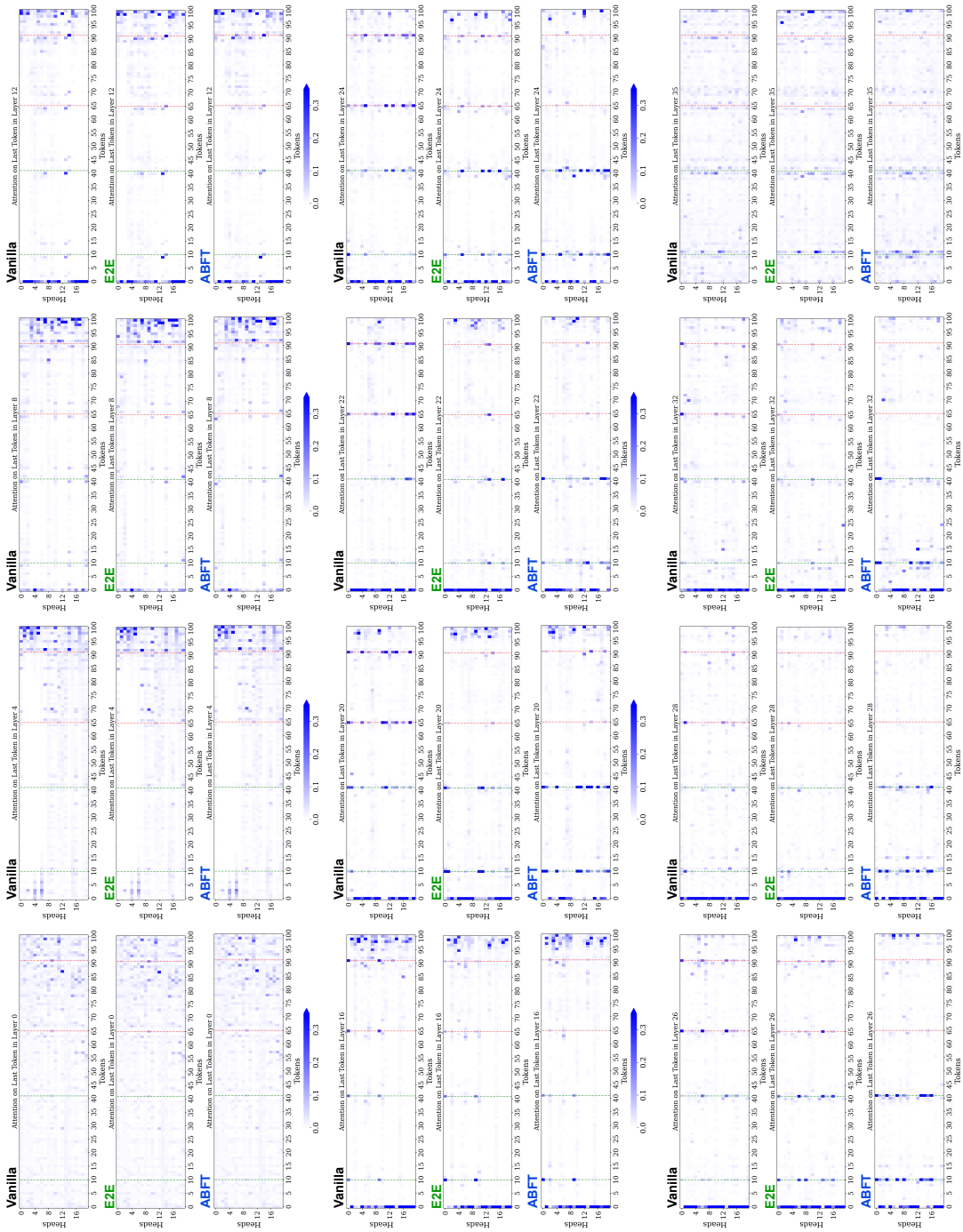


Figure D.22: Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. 6.8.

Authorship, Contribution, and AI Usage Statement

The author confirms that, although the main part of this doctoral dissertation consists of several conference papers with multiple co-authors, Yufeng Zhao (alias: Hakaze Cho) has played the principal and leading role in all of them. While co-authors have made valuable contributions through discussions, feedback, and auxiliary experiments, the conception, core methodology, main analyses, and writing were primarily undertaken by Yufeng Zhao. The overall structure and intellectual direction of this dissertation would remain coherent and substantially intact even if co-authors' specific contributions were abstracted away. In other words, Yufeng Zhao has been chiefly responsible for formulating research questions, designing methods, conducting key experiments, interpreting results, drafting manuscripts, responding to reviews, and presenting the work. The essential theoretical framework and major findings thus reflect Yufeng Zhao's central research vision and sustained effort, while properly acknowledging the collaborative nature of academic research.

Chapter 3 Author Contributions: Hakaze Cho, also known as Yufeng Zhao, handled the entire workload in this paper. He provided ideas, designed / conducted experiments, collected / described data, and wrote / revised the paper. Naoya Inoue, as the supervisor, provided valuable feedback, revisions, and financial support for this paper. M.K. and Y.S. made minor contributions that placed them on the borderline of authorship criteria: they participated in discussions and offered comments, some of which were incorporated.

Chapter 4 Author Contributions: Hakaze Cho, also known as Yufeng Zhao, handled almost the entire workload in this paper. He provided ideas, designed / conducted experiments, collected / described data, and wrote / revised the paper. Naoya Inoue, as the supervisor, provided valuable feedback, revisions, and financial support for this paper. Haolin Yang and Gouki Minegishi participated in our discussions and provided numerous important comments, as well as several major revision suggestions.

Chapter 5 Author Contributions: The vast majority of contributions of this paper are attributed to Hakaze Cho (also called Yufeng Zhao), who provided ideas, designed and conducted experiments, collected and described data, wrote papers, and revised them. Y.S., M.K., K.T., and A.I. participated in our discussion and provided some comments, with sequentially decreasing participation and contribution. They also helped write a non-peer-viewed seminar version of this paper in Japanese. Naoya Inoue is their supervisor, he provides an excellent laboratory environment, funding, necessary guidance, and paper revision.

Chapter 6 Author Contributions: Hakaze Cho, also known as Yufeng Zhao, handled the entire workload in this paper. He provided ideas, designed / conducted experiments, collected / described data, and wrote / revised the paper. Peng Luo suggested the PID algorithm, which is an essential part of this paper, and provided some technical support for it. Naoya Inoue, as the supervisor, provided valuable feedback, revisions, and financial support for this paper. M.K. and R.K. made minor contributions that placed them on the borderline of authorship criteria: they participated in discussions and offered comments, some of which were incorporated.

AI agents are used only to polish the English writing of this dissertation.

Data Release & Reproducibility Statement

The experimental code related to parts of this dissertation is publicly available as follows:

Chapter 3: The official code implementation of this paper by the author can be found at https://github.com/hc495/ICL_Circuit. Please follow the instructions in this GitHub repository to reproduce the experiments. If any final data or intermediate results are required, please get in touch with the author by contact information shown in .

Chapter 4: Source code of this paper is available at https://github.com/hc495/Verb_subspace.

Chapter 5: The official code implementation can be found at https://github.com/hc495/Hidden_Calibration.

Chapter 6: Source code of this paper is available at https://github.com/hc495/ICL_head_tuning.

All the results and intermediate results of this dissertation have been archived safely. If any verification is needed, please contact the author.

Publications During Ph.D. Program

From October 2023 to March 2026

Important note: Yufeng Zhao publishes under the Japanese spell "**Hakaze Cho**" to address pronunciation challenges of "**Yufeng Zhao**". To ensure the correct author attribution, in the following papers, it is explicitly stated that "**Hakaze Cho**" is an alias for "**Yufeng Zhao**". For verification, you may contact the email address provided in these papers or contact Prof. Naoya Inoue. Additionally, we translate the information into English for Japanese publications (**I**mpact **F**actors (IF) extracted from (Eickhoff, 2023)).

International Conferences (Refereed)

[1] **Hakaze Cho**, Haolin Yang, Brian M. Kurkoski, Naoya Inoue. Binary Autoencoder for Mechanistic Interpretability of Large Language Models. *The Thirteenth International Conference on Learning Representations* (**ICLR**). 2026. [h5-index=362, IF=48.9, core rank: A*] (Cho et al., 2025e)

[2] Haolin Yang, **Hakaze Cho**, Naoya Inoue. Localizing Task Recognition and Task Learning in In-Context Learning via Attention Head Analysis. *The Thirteenth International Conference on Learning Representations* (**ICLR**). 2026. [h5-index=362, IF=48.9, core rank: A*] (Yang et al., 2025b)

[3] Haolin Yang, **Hakaze Cho**, Kaize Ding, Naoya Inoue. Task Vectors, Learned Not Extracted: Performance Gains and Mechanistic Insight. *The Thirteenth International Conference on Learning Representations* (**ICLR**). 2026. [h5-index=362, IF=48.9, core rank: A*] (Yang et al., 2025a)

[4] Haolin Yang, **Hakaze Cho**, Yiqiao Zhong, Naoya Inoue. Unifying Attention Heads and Task Vectors via Hidden State Geometry in In-Context Learning. *The Thirty-Ninth Annual Conference on Neural Information Processing Systems* (**NeurIPS**). 2025. [h5-index=371, IF=23.3, core rank: A*] (Yang et al., 2025c)

[5] **Hakaze Cho**, Peng Luo, Mariko Kato, Rin Kaenbyou, Naoya Inoue. Mechanistic Fine-tuning for In-context Learning. *BlackboxNLP Workshop @ EMNLP*. 2025. (Cho et al., 2025b)

[6] **Hakaze Cho**, Mariko Kato, Yoshihiro Sakai, Naoya Inoue. Revisiting In-context Learning Inference Circuit in Large Language Models. *The Thirteenth International Conference on Learning Representations* (**ICLR**). 2025. [h5-index=362, IF=48.9, core rank: A*] (Cho et al., 2025a)

[7] **Hakaze Cho**, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, Naoya Inoue. Token-based Decision Criteria Are Suboptimal in In-context Learning. In proceedings of *the 2025 Annual Conference of the Nations of the Americas Chapter of the ACL* (**NAACL main conference**). 2025. [h5-index=126, IF=16.5, core rank: A] (Cho et al., 2025c)

[8] **Hakaze Cho**, Yoshihiro Sakai, Kenshiro Tanaka, Mariko Kato, Naoya Inoue. Understanding Token Probability Encoding in Output Embeddings. In proceeding of *the 31st International Conference on Computational Linguistics* (**COLING**). 2025. [h5-index=81, IF=7.7, core rank: B] (Cho et al., 2025d)

[9] Yuting Shi, Houjing Wei, Tao Jin, **Yufeng Zhao**, Naoya Inoue. Find-the-Common: A Benchmark for Explaining Visual Patterns from Images. In proceedings of *the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation* (**LREC**). 2024. [h5-index=68]

Pre-print

[1] **Hakaze Cho**, Haolin Yang, Gouki Minegishi, Naoya Inoue. Mechanism of Task-oriented Information Removal in In-context Learning. *Pre-print*. 2025. (Cho et al., 2025f)

[2] Takuya Kataiwa, **Hakaze Cho**, Tetsushi Ohki. Measuring Intrinsic Dimension of Token Embeddings. *Pre-print*. 2025.

[3] Mariko Kato, **Hakaze Cho**, Yoshihiro Sakai, Naoya Inoue. Affinity and Diversity: A Unified Metric for Demonstration Selection via Internal Representations. *Pre-print*. 2025. (Kato et al., 2025)

[4] **Hakaze Cho**, Naoya Inoue. StaICC: Standardized Evaluation for Classification Task in In-context Learning. *Pre-print*. 2025. (Cho & Inoue, 2025)

[5] **Yufeng Zhao**, Yoshihiro Sakai, Naoya Inoue. NoisyICL: A Little Noise in Model Parameters Calibrates In-context Learning. *Pre-print*. 2024. (Zhao et al., 2024)

Domestic Conferences / Journals / Miscellaneous
(English translated, mainly non-peer-reviewed)

[1] **Hakaze Cho**, Haolin Yang, Brian M. Kurkoski, Naoya Inoue. Binary Autoencoder for Mechanistic Interpretability of Large Language Models. *The 32nd Annual Conference of the Association for Natural Language Processing* (**NLP2026**). 2026.

[2] **Hakaze Cho**, Haolin Yang, Gouki Minegishi, Naoya Inoue. Mechanism of Task-oriented Information Removal in In-context Learning. *The 32nd Annual Conference of the Association for Natural Language Processing* (**NLP2026**). 2026.

[3] Koshiro Aoki, Ryota Takatsuki, Gouki Minegishi, **Hakaze Cho**. Mechanistic Interpretability: A New Trend in Interpretability Research. *Transactions of the Japanese Society for Artificial Intelligence.* 2025. (Under Review)

[4] **Hakaze Cho**, Naoya Inoue. Conference Note: Token-based Decision Criteria Are Suboptimal in In-context Learning. *Journal of Natural Language Processing.* 2025.

[5] Takuya Kataiwa, **Hakaze Cho**, Tetsushi Ohki. Measuring Intrstic Dimension of Embedding Layer. *The 39th Annual Conference of the Japanese Society for Artificial Intelligence.* 2025.

[6] Kenshiro Tanaka, Yoshihiro Sakai, **Hakaze Cho**, Naoya Inoue, Kai Sato, Ryosuke Takahashi, Benjamin Heinzerling, Kentaro Inui. Analysis of Internal Representations of Knowledge with Expressions of Familiarity. *The 39th Annual Conference of the Japanese Society for Artificial Intelligence.* 2025.

[7] Kai Sato, Ryosuke Takahashi, Benjamin Heinzerling, Kenshiro Tanaka, **Hakaze Cho**, Yoshihiro Sakai, Naoya Inoue, Kentaro Inui. Internal Representations of Knowledge Recognition in Language Models. *The 39th Annual Conference of the Japanese Society for Artificial Intelligence.* 2025.

[8] **Hakaze Cho**, Mariko Kato, Yoshihiro Sakai, Naoya Inoue. Revisiting In-context Learning Inference Circuit in Large Language Models. *The 31st Annual Conference of the Association for Natural Language Processing* (**NLP2025**). 2025. **Outstanding Paper Awards**.

[9] **Hakaze Cho**, Naoya Inoue. Beyond the Induction Circuit: A Mechanistic Prototype for Out-of-domain In-context Learning. *The 31st Annual Conference of the Association for Natural Language Processing* (**NLP2025**). 2025.

[10] Takuya Kataiwa, **Hakaze Cho**, Tetsushi Ohki. Measuring Intrstic Dimension of Embedding Layer. *The 31st Annual Conference of the Association for Natural Language Processing* (**NLP2025**). 2025.

[11] Kato Mariko, **Hakaze Cho**, Yoshihiro Sakai, Naoya Inoue. Affinity and Diversity in Demonstrations Selection of In-context Learning. *The 31st Annual Conference of the Association for Natural Language Processing* (**NLP2025**). 2025.

[12] **Hakaze Cho**, Naoya Inoue. StaICC: Standardized Evaluation for Classification Task in In-context Learning. *The 19th Symposium of Young Researcher Association for NLP Studies* (**YANS2024**). 2024.

[13] Mariko Kato, **Hakaze Cho**, Zhenzhu Yan, Yuting Shi, Naoya Inoue. Image Feature Vectors are Frozen Informative Tokens for Language Models. *The 19th Symposium of Young Researcher Association for NLP Studies* (**YANS2024**). 2024.

[14] **Hakaze Cho**, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, Naoya Inoue. Token-based Decision Criteria Are Suboptimal in In-context Learning. *The 260th NLP Workshop of the Information Processing Society of Japan* (**IPSJ-NL**). 2024. **Awards for Young Scholars**.

[15] **Yufeng Zhao**, Yoshihiro Sakai, Naoya Inoue. NoisyICL: A Little Noise in Model Parameters Calibrates In-context Learning. *The 30th Annual Conference of the Association for Natural Language Processing* (**NLP2024**). 2024.

[16] Yoshihiro Sakai, **Hakaze Cho**, Naoya Inoue. Can LLM Learn Prompt Format in In-context Learning? *The 30th Annual Conference of the Association for Natural Language Processing* (**NLP2024**). 2024. **SB Intuitions Awards**.

[17] Yuting Shi, Houjing Wei, Jin Tao, **Yoshihiro Zhao**, Naoya Inoue. Find-the-Common: Benchmarking Inductive Reasoning Ability on Vision-Language Models. *The 30th Annual Conference of the Association for Natural Language Processing* (**NLP2024**). 2024.

Other Academic Activities

Peer Review: ICLR 2026; NeurIPS 2025; ACL ARR 2025: May (EMNLP 2025), July (AACL 2025), October (EACL 2025); ACL ARR 2026: January (ACL 2026); ICML 2026; ICML 2025 Actionable Interpretability Workshop; ACL 2025 Student Research Workshop.

Copyright © 2026 by Yufeng Zhao (Hakaze Cho). All rights reserved.

LaTeX template designed by Yufeng Zhao (mainly based on ICLR official and JAIST rules).