# Mechanistic Fine-tuning for In-context Learning

Anonymous ACL submission

## Abstract

**I**n-**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 paper, 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.

## 1 Introduction

**I**n-**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. 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 §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 paper, 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. 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. 1 (C, D), we propose **A**ttention **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 mas-

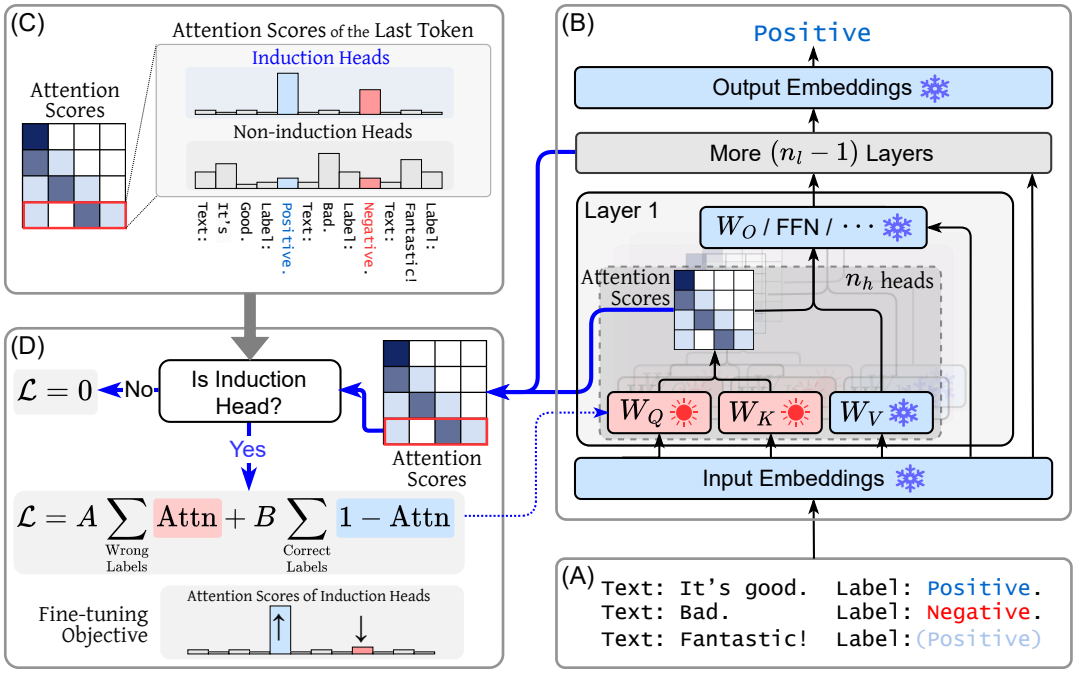



Figure 1: Diagram of ABFT framework. (A) **An example of ICL-style inputs.** We build datasets from such examples to fine-tune models. (B) **Feed-forward inference of ICL.** We collect the attention scores of every attention head in every layer to calculate the training objective. and we only enable 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.

sive 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.

## 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. 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., 2022; Mishra et al., 2022; Wang et al., 2022; Wei et al., 2023). 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. 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., 2024; 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. 1 (A), as the attention query) concentrate on the label tokens presented in the demonstrations (e.g., "positive"s and "negative"s in Fig. 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.

## 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 paper, as shown in Fig. 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. 1 (A). **(2) Feed-forward Calculation**: For each training sample, as shown in Fig. 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. 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. 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 $S = \sum_{j \in \mathcal{I}} \alpha_j$.[1] 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 §5.2.

**Loss Function.** As shown in Fig. 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 $\mathcal{I}^- = \mathcal{I} \backslash \mathcal{I}^+$,[2] we calculate the loss from the last row ($\alpha$) of $\mathcal{A}$ as:

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

That is, as shown in Fig. 1 (D), we punish the "wrong" attention scores towards the label tokens different 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 §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 2) to highlight the efficiency and harmlessness of the ABFT method.

[1] For the case shown in Fig. 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.

[2] For the case shown in Fig. 1 (C), $\mathcal{I}^+ = \{4\}$ (the position of "positive"), $\mathcal{I}^- = \{8\}$ (the position of "negative").

Table 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **GPT2-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** |
| **GPT2-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** |
| **Falcon3** | 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** |
| **Llama3** | 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** |
| **Qwen2.5** 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** |
| **Llama3** 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** |
| **Llama3** 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** |

# 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.

## 4.1 Experiment Settings

**Models and Datasets.** We conduct our experiment on **9 modern LLMs**: GPT2 (Large, XL) (Radford et al., 2019), Falcon3 7B (Team, 2024b), Llama3 (8B, 43B, 56B) (AI@Meta, 2024), DeepSeek-R1 Distill Qwen 14B (DeepSeek-AI, 2025), Qwen2.5 32B (Team, 2024a; Yang et al., 2024), and SimpleScaling s1.1 32B (Muennighoff et al., 2025); and **8 datasets**: SST2, SST5 (Socher et al., 2013), MR (Pang and Lee, 2005), **F**inancial **P**hrasebank (Malo et al., 2014), TREC (Li and Roth, 2002; Hovy et al., 2001), **Subj**ective (Wang and Manning, 2012), **T**weet **E**val **E**motion (Mohammad et al., 2018), **T**weet **E**val **H**ate (Basile et al., 2019) (Refer Appendix A.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 and Ba, 2014) is used with learning rate $\text{lr} = 2 \times 10^{-5}$ 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 A.3), stabilizing the number of attention heads identified as induction heads (see §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., 2022), and PICL re-pre-training with 80M samples (Gu et al., 2023).

Table 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 | $\text{Acc}_{\text{ID}}$ | $\text{Acc}_{\text{OD}}$ |
|---|---|---|---|---|---|
| **Llama3** 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 |
| **Qwen2.5** 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 |
| **Llama3** 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 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 |
| GPT2-XL | 81.28 | **91.74** | 68.38 | **82.75** |
| Llama3 8B | 86.93 | **90.32** | 76.99 | **92.00** |
| DeepSeek-R1 14B | 89.64 | **92.79** | 81.30 | **85.97** |
| Qwen2.5 32B | 88.97 | **92.78** | 84.52 | **87.94** |
| Llama3 56B | 92.78 | **93.90** | 82.10 | **87.49** |

**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 and Inoue, 2025).

### 4.2 Main Results

The test results are shown in Table 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.

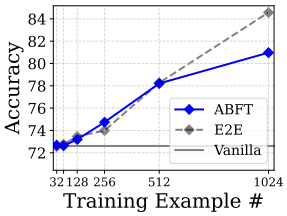



Figure 2: Data efficiency: accuracy against data size on DeepSeek-R1 Distill Qwen 14B / 8 datasets.

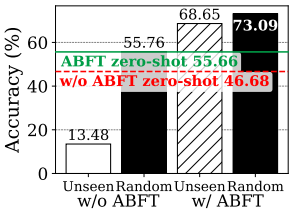



Figure 3: Accuracy on unseen label inputs and randomly sampled inputs, w/ and w/o ABFT.

**Prediction Consistency.** We evaluate the prediction consistency against variations in (1) prompt templates and (2) demonstration sampling on StaICC-Diag (Cho and 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 and Inoue (2025) and Appendix A.5 for implementation details. As shown in Table 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. 3 shows that (see Appendix A.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[3].

### 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

[3]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. 10, 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.

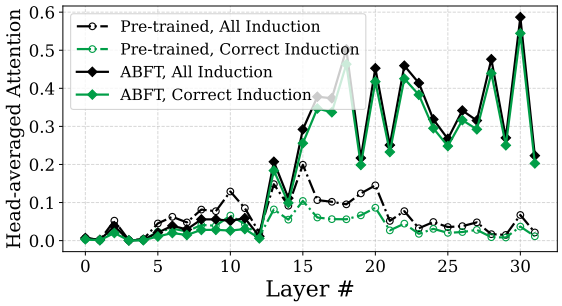



Figure 4: Induction attention scores and correct (on the correct labels) induction attention score averaged among heads on each layer. **ABFT enhances induction attention, and improves the correctness**.

compared with E2E fine-tuning, ABFT is more efficient, and more harmless on tasks out of the fine-tuning domain.

**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 2. E2E fine-tuning slightly outperforms ABFT in in-domain accuracy ($\text{ACC}_{\text{ID}}$), 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 2, $\text{ACC}_{\text{OD}}$). 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.

**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. 2 (refer to Appendix C.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.

Table 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. |
|---|---|---|---|
| **Falcon3** 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 |
| **Llama3** 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 |

## 5 Analysis

### 5.1 Attention Visualization after ABFT

As shown in Fig. 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 C.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 middle-to-last layers. Such visualization indicates that ABFT successfully generalizes to correct the behavior of attention heads.

### 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 4, where:

**ABFT should be Localized.** In Table 4, disabling the head filter ((D) in Fig. 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

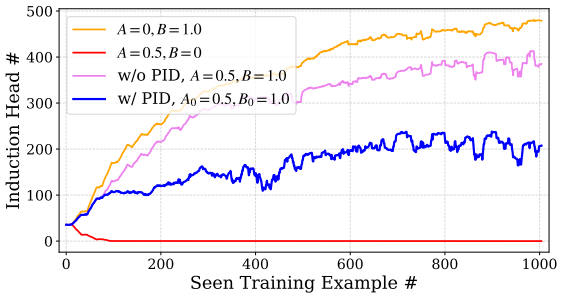



Figure 5: Number of attention heads that are judged as induction heads on 4 settings, against the training processing (the number of seen data samples). **PID effectively stabilizes the induction head numbers**.

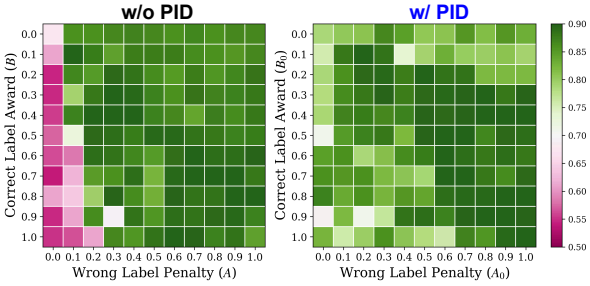



Figure 6: Accuracies with various settings on hyperparameter $A$ and $B$, with and without PID algorithm. **PID weakens the sensitivity to initial parameters**.

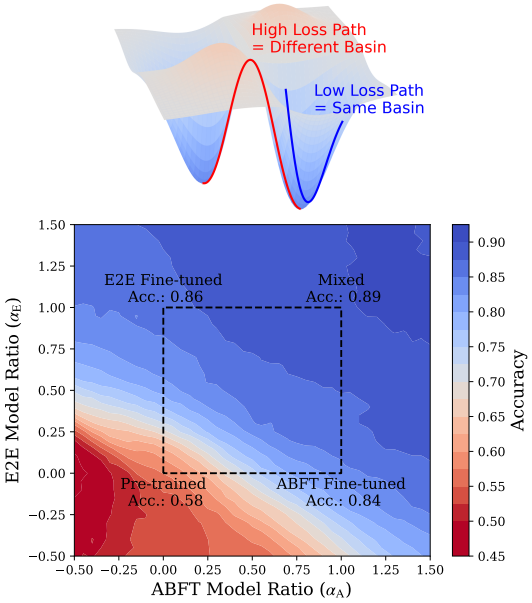



Figure 7: **Upper**: A better accuracy (lower loss) in the interpolation path suggests the same basin, and vice versa. **Lower**: Contour map of accuracies against the coefficient $\alpha_{\mathrm{E}}$ and $\alpha_{\mathrm{A}}$ in Eq. 2. **E2E fine-tuned and ABFT models are located in the same low-loss area.**

functions other than the induction head are still necessary for ICL, aligning with and enhancing the previous work (Reddy, 2024; 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 §4.1, we use the PID algorithm to adaptively balance the value of $A$ and $B$ in the loss calculation. In Table 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 §3, and we will discuss this in-depth in the next section (§5.3).

### 5.3 Balance the Loss Factor $A$ and $B$

**Punish and Reward Influence Induction Heads Antagonisticly.** As shown in Table 4, removing either loss term in Eq. 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 Llama3 8B and SST2 (see Appendix C.2 for more cases), as shown in Fig. 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. 1 using a classical PID controller (Appendix A.3)[4]. As shown in Fig. 5, PID stabilizes induction head count, improves ABFT performance (Table 4), and reduces sensitivity to hyperparameters $A_0$ and $B_0$, with accuracy remaining stable across settings (Fig. 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;

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

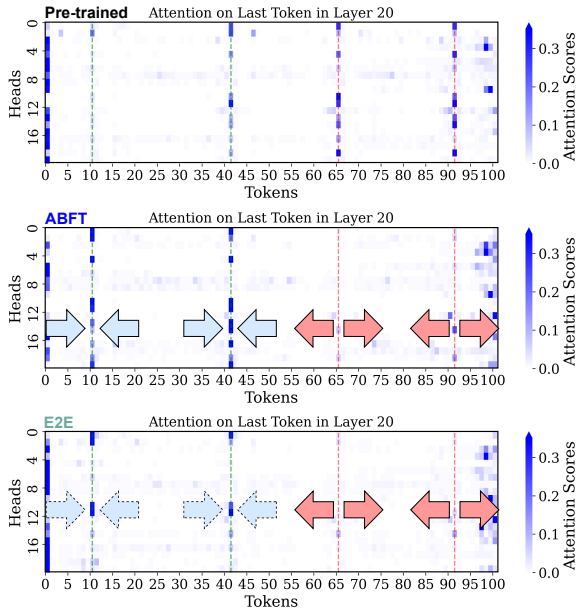



Figure 8: Attention score visualization of the pre-trained model, ABFT model, and E2E fine-tuned model, on the same input as Fig. 16, and the same models as Fig. 7 (Refer to Appendix C.3 for details and more cases).

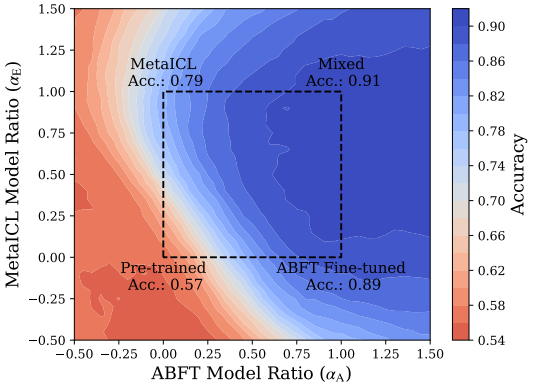



Figure 9: Contour map of accuracies against the $\alpha_E$ and $\alpha_A$ on GPT2-L, and the $\theta_E$ is set as MetaICL model.

Singh et al., 2024)—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 $\theta_0$ (as the start point of the fine-tuning) and fine-tuned parameters $\theta_E$ for E2E fine-tuning, and $\theta_A$ for ABFT. In detail, we mix these three parameters into a new model parameter set $\theta$ in the following form:

$$\theta = \theta_0 + \alpha_E(\theta_E - \theta_0) + \alpha_A(\theta_A - \theta_0), \quad (2)$$

and then test the accuracy of $\theta$ as an anti-metric of model loss. As shown in Fig. 7 (upper), if the accuracy of mixed $\theta$ is better (or at least, not significantly worse) than the accuracy of $\theta_E$ and $\theta_A$, we can infer that the $\theta_E$ and $\theta_A$ are in the same loss basin, with linear low-loss path observed.

**Experiment and Result.** We conduct the aforementioned experiment protocol on SST2 and GPT2-XL. The results are shown in Fig. 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. 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. 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*.

## 7 Discussion

**Conclusion.** In this paper, 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 paper, 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.

## 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., 2024; Cho et al., 2025a), so that our proposed ABFT approach, applies only to classification tasks with a finite label set. As mentioned in §4.2 and Fig. 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 paper 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.

**Towards Better Performance.** It can be considered that some hyperparameters (see §4.1 and Appendix A.3), and the induction head filter (see §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. 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 $\theta = \theta_0 + \alpha_{\mathrm{A}}(\theta_{\mathrm{A}} - \theta_0), \alpha_{\mathrm{A}} > 1$ to further improve accuracy without any gradient-based cost.

**Towards Further Efficiency.** As shown in Appendix B, 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).


## Acknowledgements

Not available during the anonymous review.

## References


AI@Meta. 2024. Llama 3 model card.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. 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*, pages 3558–3573.

Hakaze Cho and Naoya Inoue. 2025. Staicc: Standardized evaluation for classification task in in-context learning. *arXiv preprint arXiv:2501.15708*.

Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. 2025a. Revisiting in-context learning inference circuit in large language models. In *The Thirteenth International Conference on Learning Representations*.

Hakaze Cho, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, and Naoya Inoue. 2025b. Token-based decision criteria are suboptimal in in-context learning. In *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)*, pages 5378–5401, Albuquerque, New Mexico. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4849–4870, Toronto, Canada. Association for Computational Linguistics.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2023. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. *Advances in neural information processing systems*, 29.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. 2023. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2312–2333, Singapore. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

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

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. 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*, pages 2791–2809.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 3470–3487. Association for Computational Linguistics (ACL).

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523.

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

Bo Pang and Lillian Lee. 2005. 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)*, pages 115–124.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

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

Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie CY Chan, and Andrew M Saxe. 2024. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *Forty-first International Conference on Machine Learning*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Qwen Team. 2024a. Qwen2.5: A party of foundation models.

TII Team. 2024b. Falcon 3 family of open foundation models.

Sida Wang and Christopher D. Manning. 2012. 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, page 90¨C94, USA. Association for Computational Linguistics.

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

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

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. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*.

# A Detailed Experiment Implementation

## A.1 Model & Dataset Details

**Models.** All the models in this paper are loaded from `huggingface`. In detail, we list the `huggingface` repository name to keep the repeatability of this paper, as shown in Table 6.

**Dataset Split.** As also described by Cho and 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 §A.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.

## A.2 Prompt Templates

We utilize the default prompt templates of StaICC, as shown in Table 5. For the sake of simplicity, we reduce the label tokens into one token, as also shown in Table 5.

## A.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 §3 and Fig. 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[5] by standard PID algorithm as:

$$\begin{aligned} A_t &= C_p\left(\bar{n}_t - \bar{n}_{t-1}\right) \\ &+ C_i\left(\sum_{i=2}^{t} \bar{n}_i - \bar{n}_{i-1}\right) \\ &+ C_d\left(\bar{n}_t - 2\bar{n}_{t-1} + \bar{n}_{t-2}\right) \\ &+ A_{t-1}, \end{aligned} \tag{3}$$

[5]Remind that for the sake of simplicity, we only control $A$, given the findings of $A$ and $B$ are antagonistic, as shown in §5.3.

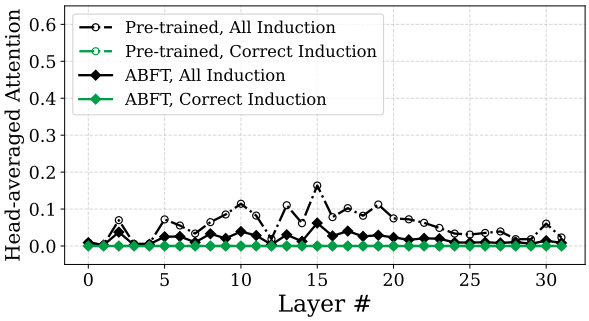



Figure 10: Visualization of induction attention scores on unseen label settings (Llama3 8B, TREC).

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.

## A.4 Experiment Protocol of Unseen Label

In Fig. 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 Llama3 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 5 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. 4 on the unseen label condition in Fig. 10, where the induction heads in the ABFT model are almost fully suppressed but with considerable inference accuracy, which implies a new inference mechanism.

## A.5 Experiment Protocol of Stability against Prompting

In Table 3, we test whether the prediction of ICL is stable against various **(1)** prompt templates and **(2)** demonstration sampling, on the StaICC-Diag (Cho and Inoue, 2025) benchmark, whose method is described briefly below.

**Method.** To test the prediction robustness against prompt templates / demonstration sampling, we repeat several predictions for each query on various prompt templates / demonstration sampling,

Table 5: Prompt templates used in this paper.

| Dataset | Prompt Template (Unit) | Label Tokens |
|---|---|---|
| SST2 | `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 |
| SST5 | `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 |

Table 6: `Huggingface` repository name for models used in this paper.

| Model | Repository |
|---|---|
| GPT2-L | `openai-community/gpt2-large` |
| GPT2-XL | `openai-community/gpt2-xl` |
| Falcon3 | `tiiuae/Falcon3-7B-Base` |
| Llama3 (8B) | `meta-llama/Meta-Llama-3-8B` |
| DeepSeek-R1 | `deepseek-ai/DeepSeek-R1-Distill-Qwen-14B` |
| Qwen2.5 | `Qwen/Qwen2.5-32B` |
| SimpleScaling s1.1 | `simplescaling/s1.1-32B` |
| Llama3 (43B) | `chargoddard/llama3-42b-v0` |
| Llama3 (56B) | `nyunai/nyun-c2-llama3-56B` |

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 and 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 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., 2023; Zhou et al., 2024; Jiang et al., 2023; Cho et al., 2025b), ABFT can be regarded as an implicit calibration inside the LLM.

# B Parameter Shift after ABFT against Layers

We utilize the Frobenius norm to visualize the shifting distance of the parameter matrix $\theta$ before and after ABFT ($\theta'$) as $\|\theta - \theta'\|_2$. The results are shown in Fig. 17, 18, 19, 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. This observation is consistent with previous works on Induction Heads (Cho et al., 2025a).

# C Augmentation Results

## C.1 Augmentation Results for Data Efficiency (Fig. 2)

We repeat the data efficiency experiments shown in Fig. 2 on Qwen2.5 32B, as shown in Fig. 15. The results are globally consistent with Fig. 2.

## C.2 Augmentation Results for Number of Induction Heads against Training Processing (Fig. 5)

We repeat the visualization of the number of induction heads against the training processing under various settings on Llama3 8B and Falcon3 7B as shown in Fig. 20 and 21. The results are globally consistent with Fig. 5.

Moreover, we visualize the number of induction heads on only standard settings, as shown in Fig. 22-28 for reference.

## C.3 Augmentation Results for Attention Visualization (Fig. 4 and 8)

As shown in Fig. 16, we visualize the attention score on the last token of the given input example in the validation set on Llama3 8B, and repeat this visualization on more input cases for Fig. 16 and Fig. 8 in Fig. 29. Moreover, we expand the Fig. 16 towards more layers in Fig. 30, and Fig. 8 in Fig. 31. 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.

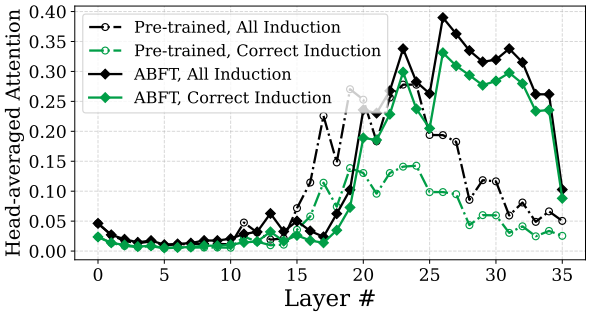



Figure 11: Augmentation results of Fig. 4 on GPT2-L.

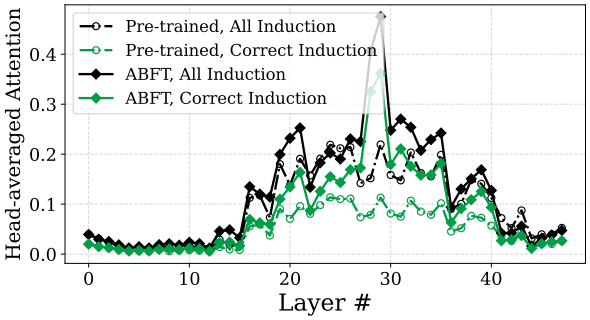



Figure 12: Augmentation results of Fig. 4 on GPT2-XL.

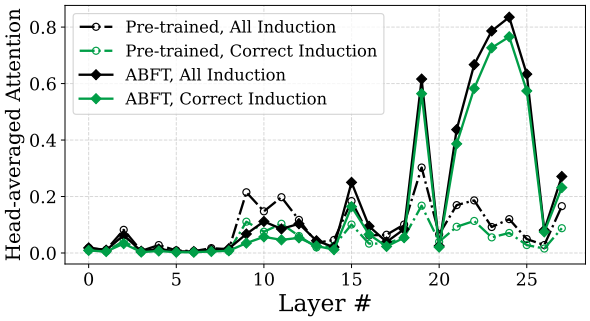



Figure 13: Augmentation results of Fig. 4 on Falcon3 7B.

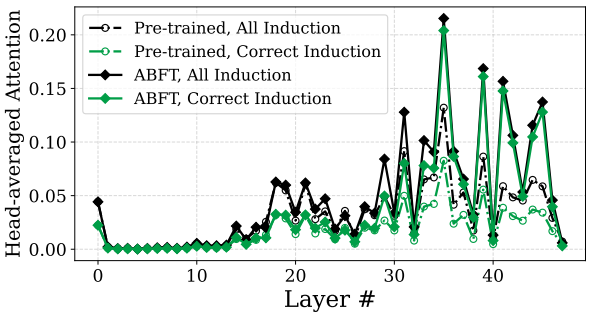



Figure 14: Augmentation results of Fig. 4 on Llama3 42B.

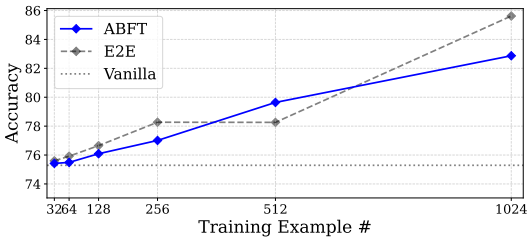



Figure 15: Accuracy against training set size as a metric of data efficiency, for Qwen2.5 32B.

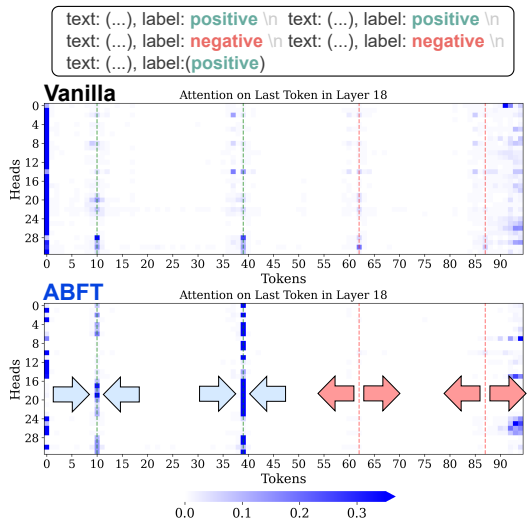



Figure 16: Attention score visualization on the last token of ICL input of every attention head (vertical axis) towards each token. Label tokens and their contents are marked with dotted lines. Refer to Appendix C.3 for more examples and layers. **ABFT successfully focuses attention scores to correct labels**.

Moreover, we repeat the attention score visualization similar to Fig. 4 on more models and SST2, as shown in Fig. 11, 12, 13, and 14.

## D Statements

**Author Contributions Statement.** Not available during the anonymous review.

**License for Artifacts.** Models and datasets used in this paper are used in their original usage, and are open-sourced with the following license:

- `mit`: GPT2-L, GPT2-XL, DeepSeek-R1
- Individual License or Unknown: Falcon3, Llama3 8B, Llama3 43B, Llama3 56B, SST5, MR, TREC, SUBJ, TEE, TEH
- `Apache 2.0`: Qwen2.5, SimpleScaling s1.1
- `cc-by-sa-3.0`: SST2, FP,

**AI Agent Usage.** AI Agents are used and only used for writing improvement in this paper.

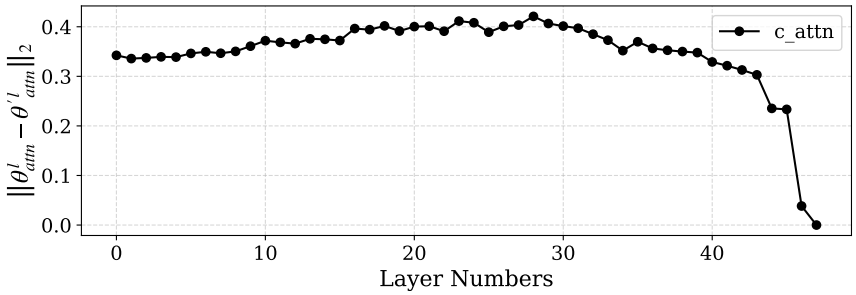



Figure 17: Shifting distance before and after ABFT on the `c_attn` matrix of GPT2-XL and SST2.

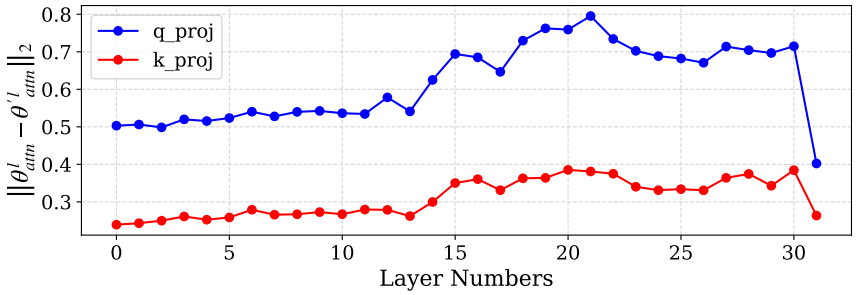



Figure 18: Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama3-8B and SST2.

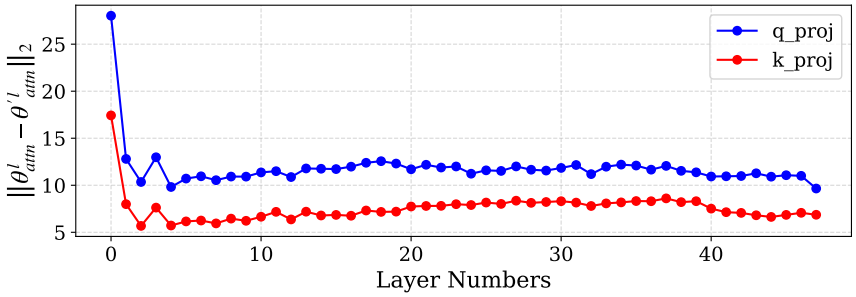
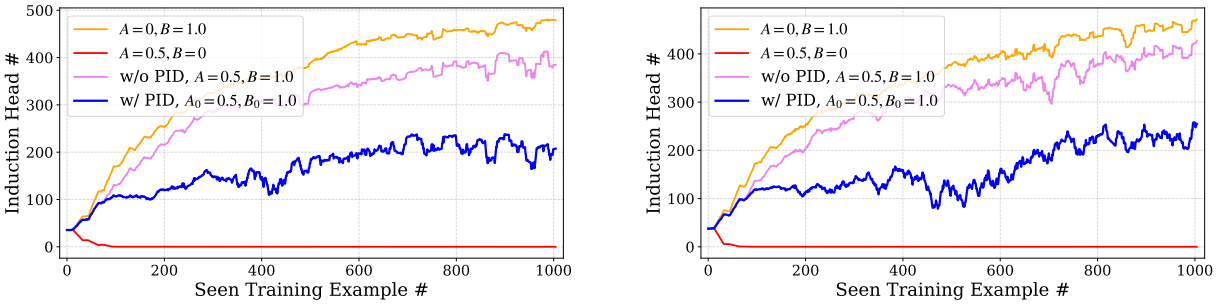


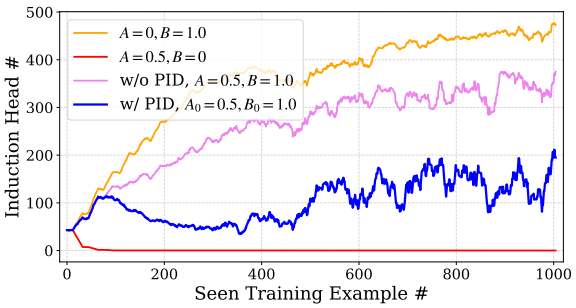

Figure 19: Shifting distance before and after ABFT on the `q_proj` and `k_proj` matrix of Llama3-42B and SST2.



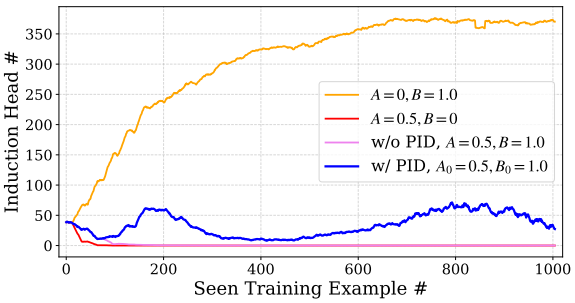

(a) Llama3-8B SST2 (Fig. 5)



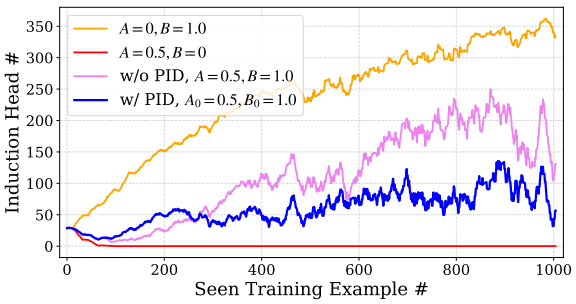

(b) Llama3-8B MR



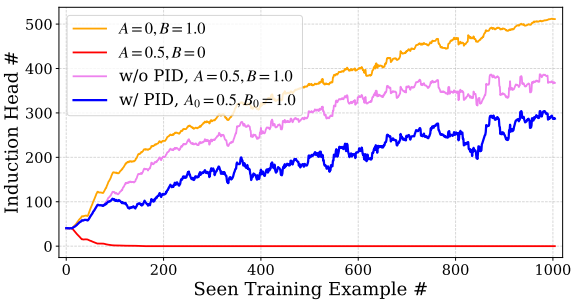

(c) Llama3-8B FP



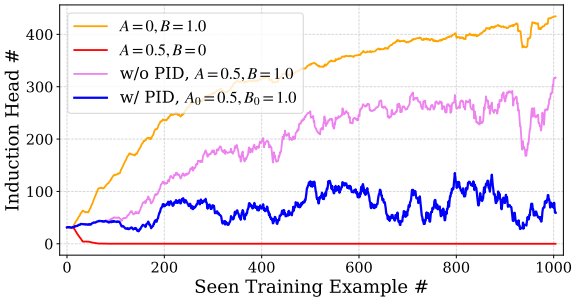

(d) Llama3-8B SST5



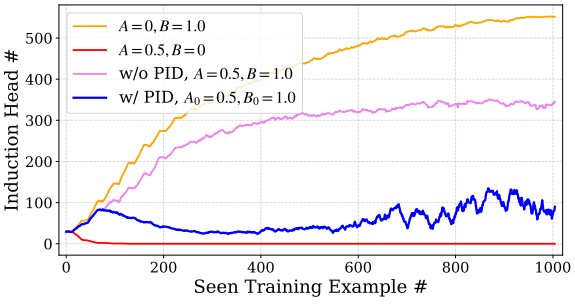

(e) Llama3-8B TREC




(f) Llama3-8B SUBJ

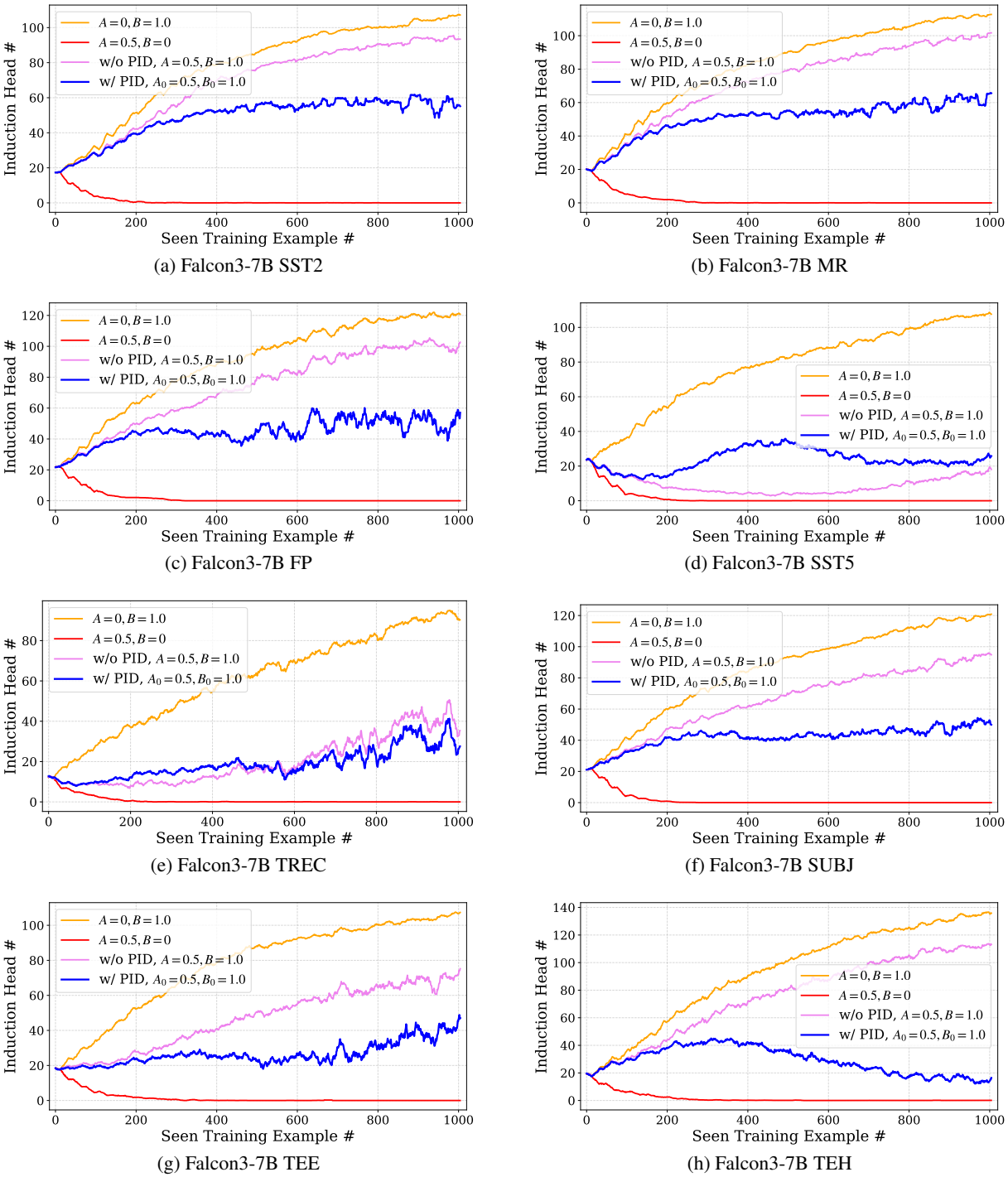



(g) Llama3-8B TEE




(h) Llama3-8B TEH

Figure 20: Induction head numbers along training dynamics on Llama3-8B and all 8 datasets.




(a) Falcon3-7B SST2

(b) Falcon3-7B MR

(c) Falcon3-7B FP

(d) Falcon3-7B SST5

(e) Falcon3-7B TREC

(f) Falcon3-7B SUBJ

(g) Falcon3-7B TEE

(h) Falcon3-7B TEH

Figure 21: Induction head numbers along training dynamics on Falcon3-7B and all 8 datasets.

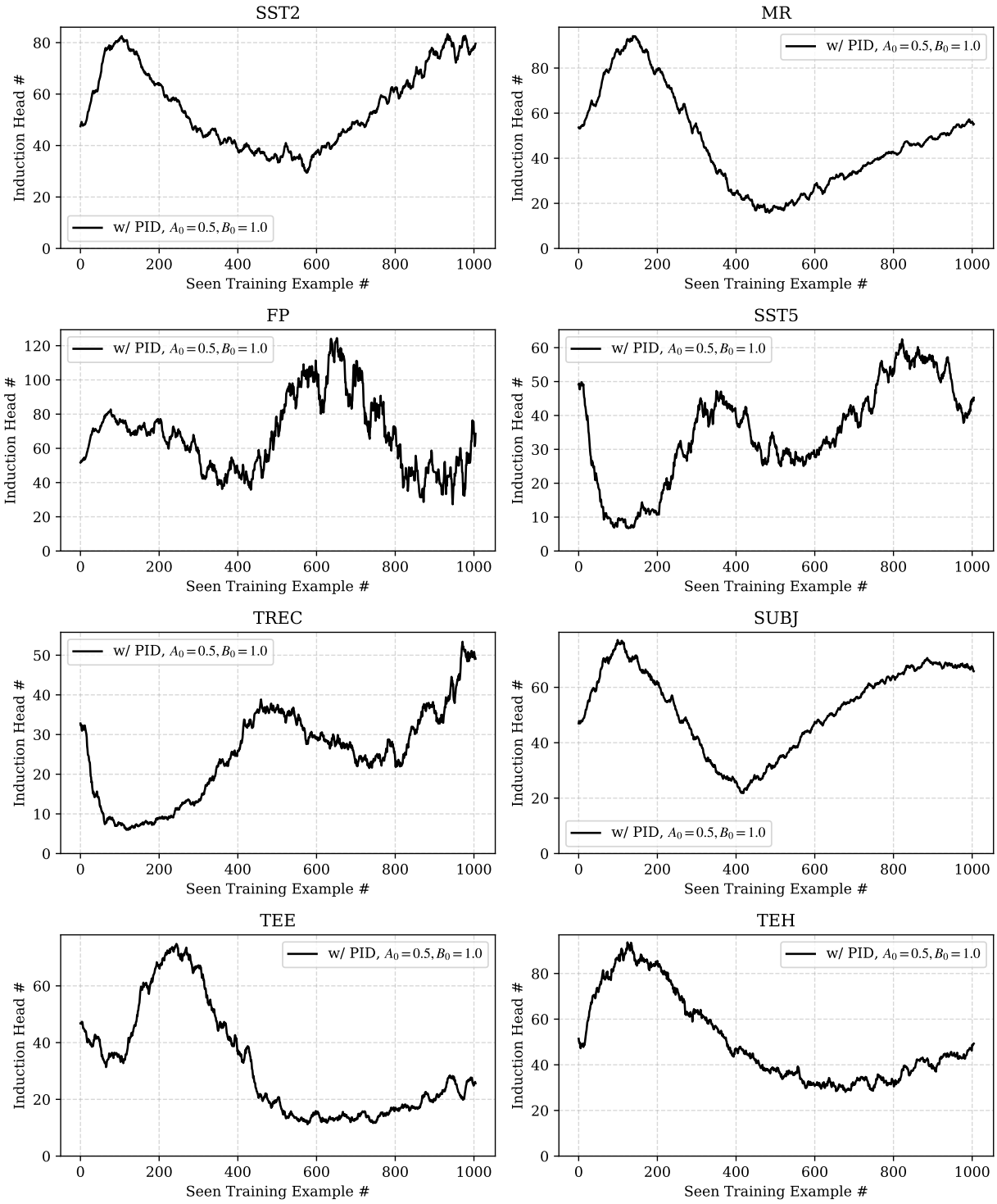



Figure 22: Induction head numbers along training dynamics on GPT2-Large and all 8 datasets.

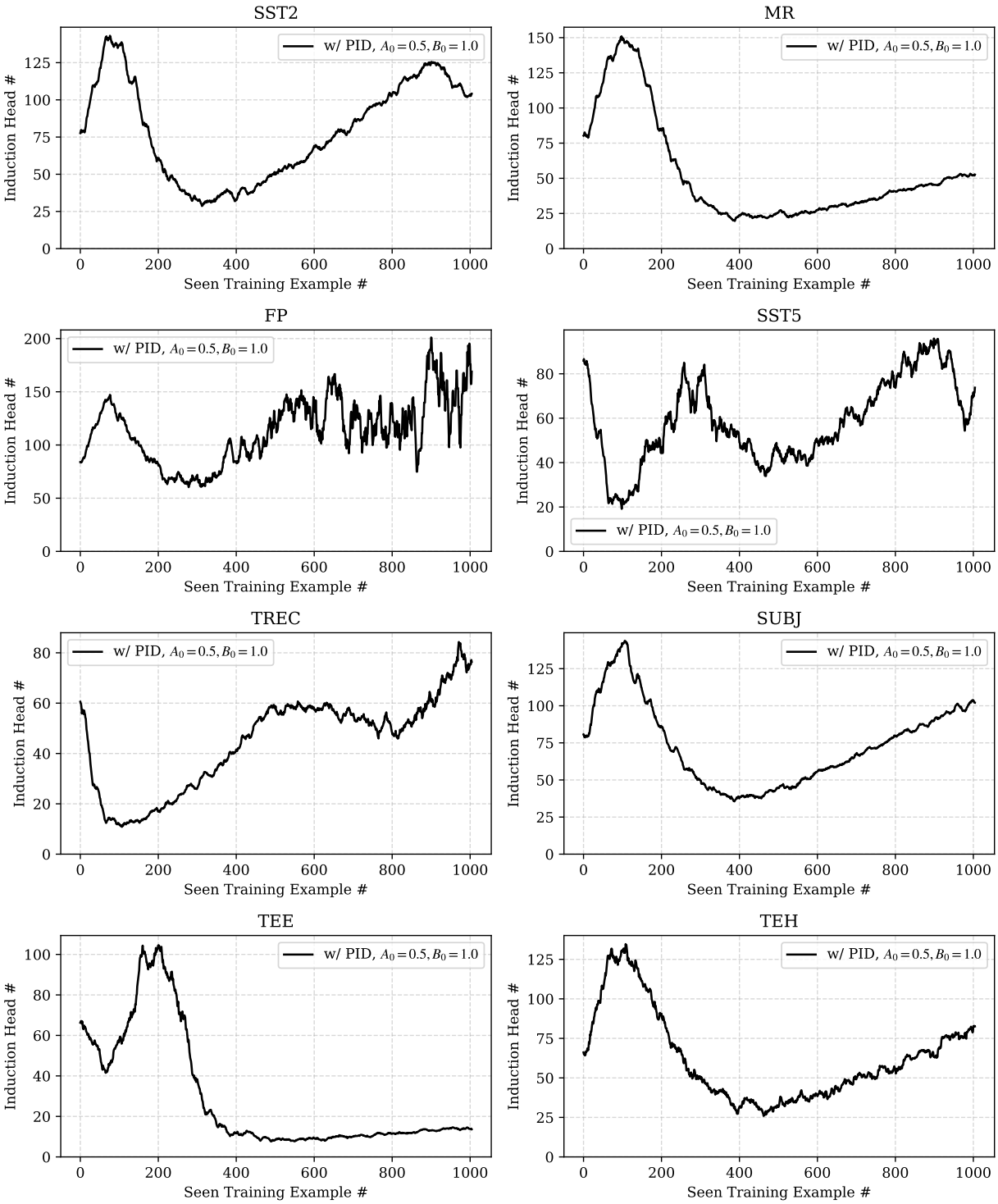



Figure 23: Induction head numbers along training dynamics on GPT2-XL and all 8 datasets.

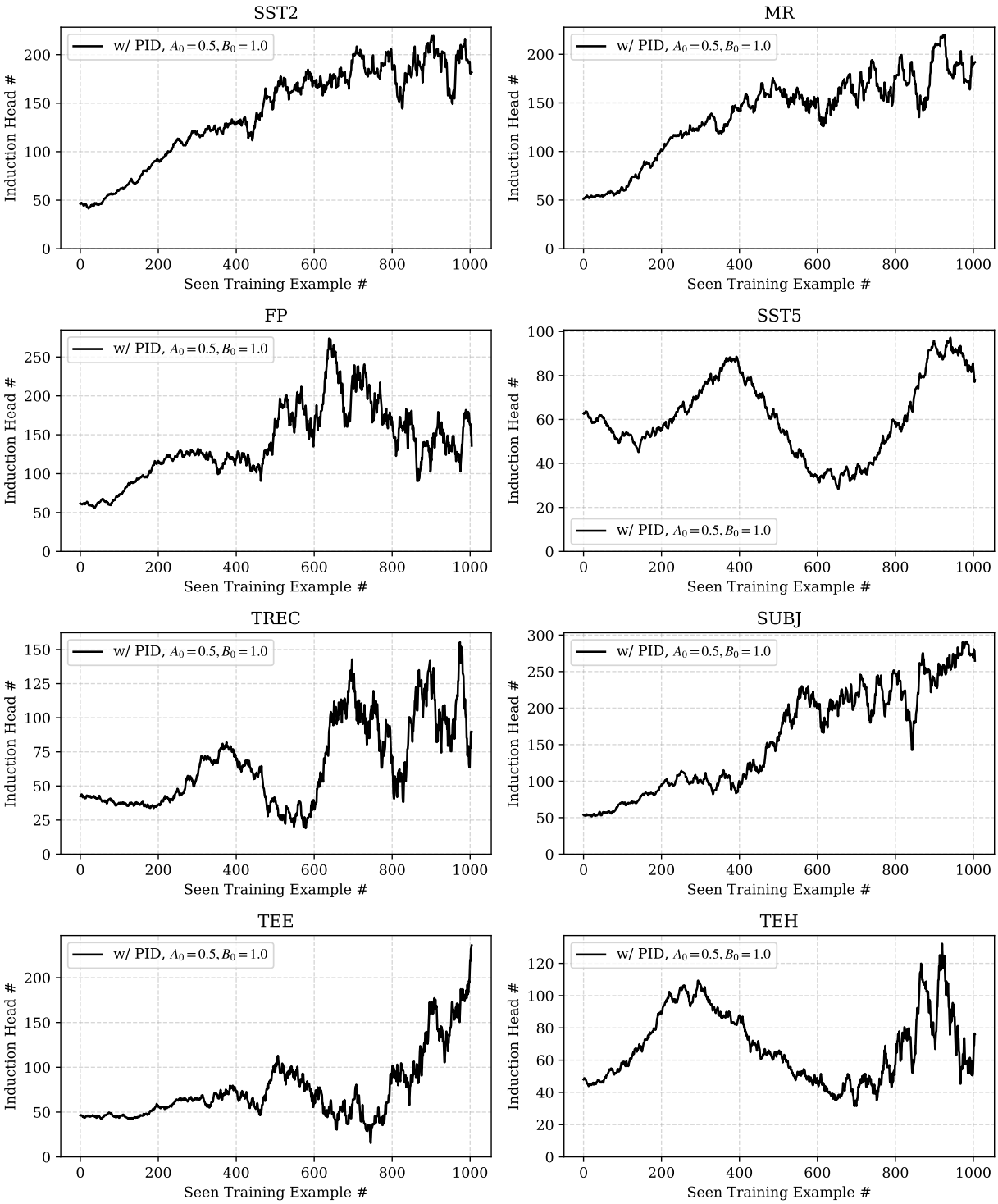



Figure 24: Induction head numbers along training dynamics on DeepSeek-R1 and all 8 datasets.

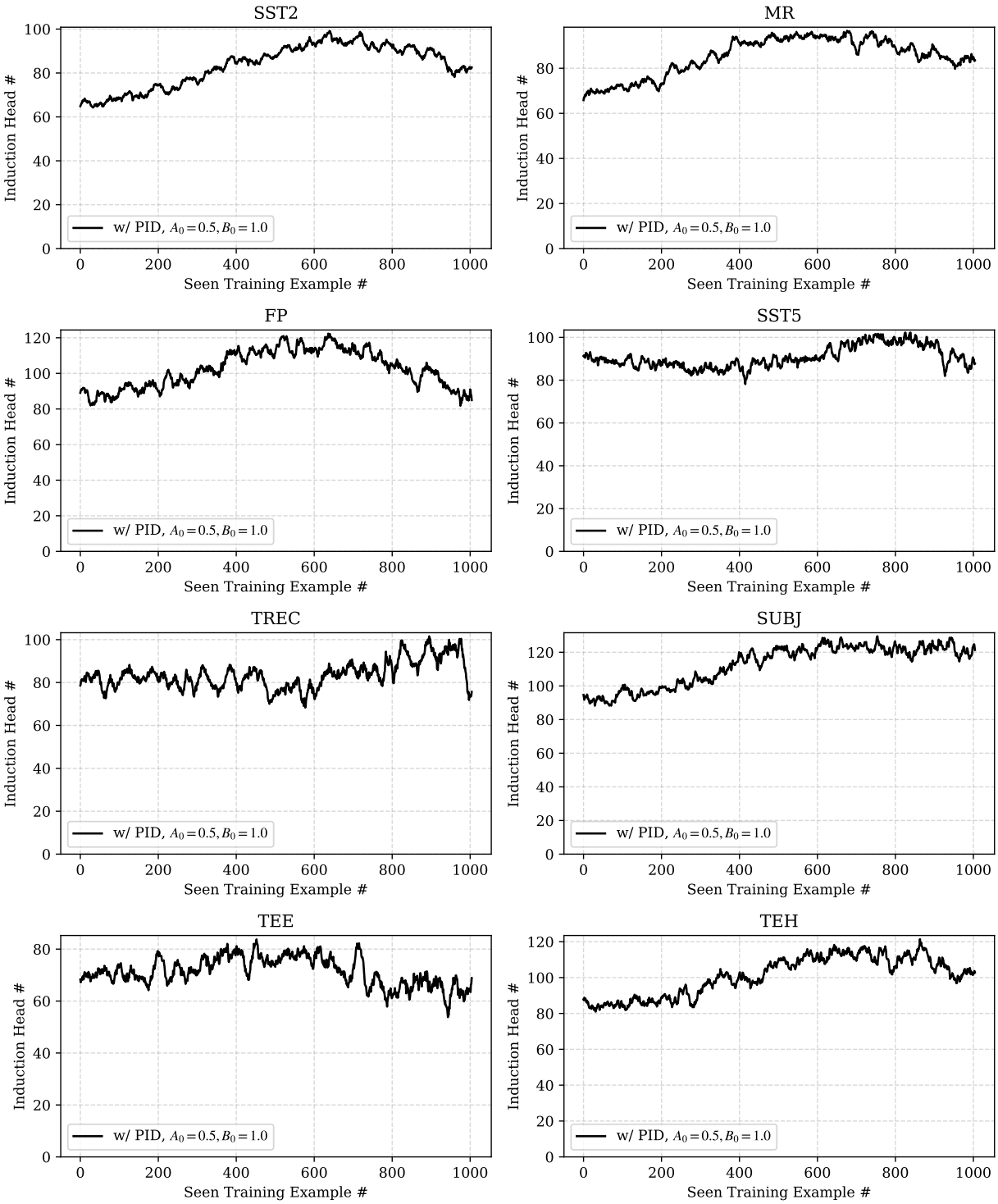



Figure 25: Induction head numbers along training dynamics on Qwen2.5-32B and all 8 datasets.

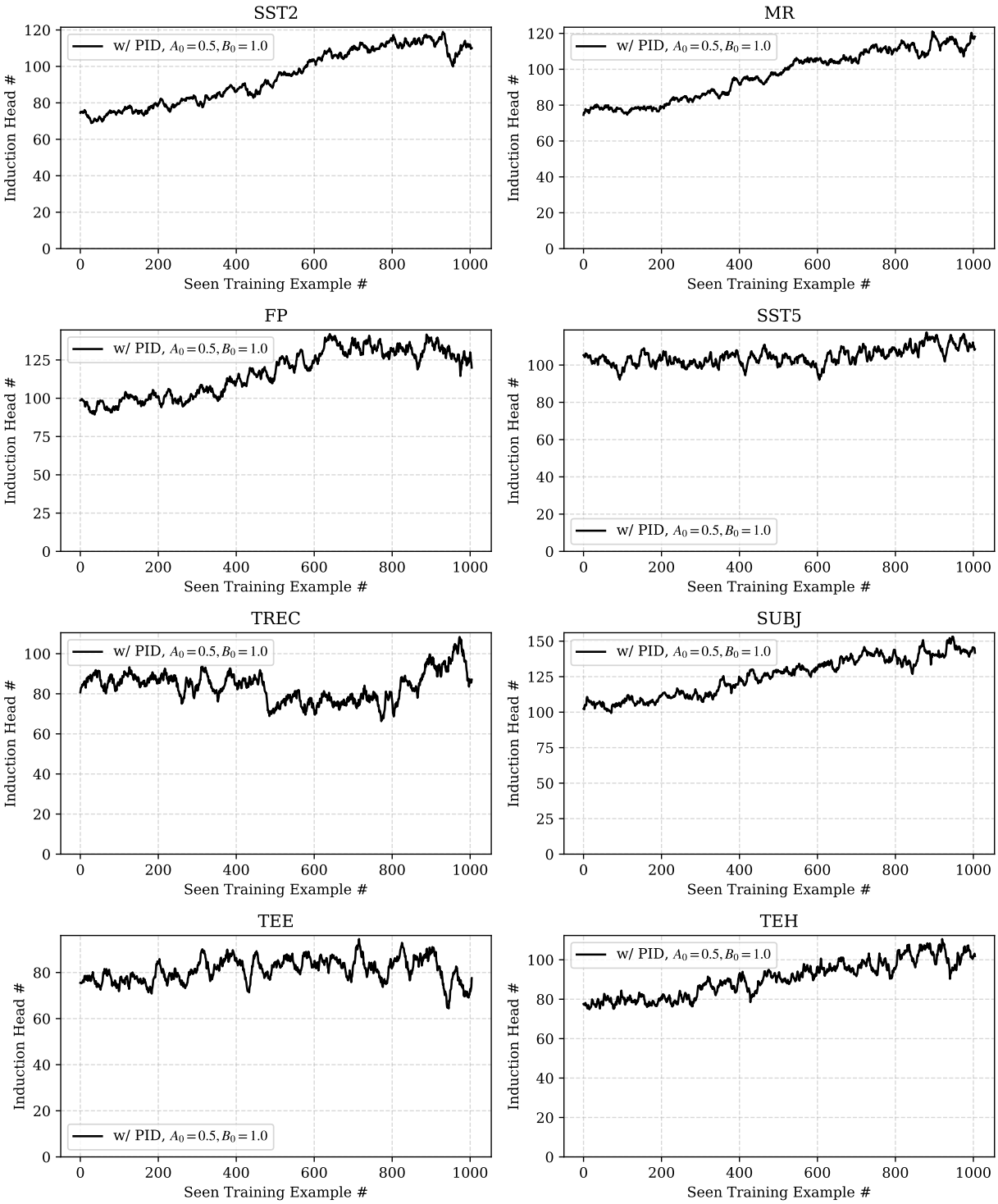



Figure 26: Induction head numbers along training dynamics on SimpleScaling s1.1 and all 8 datasets.

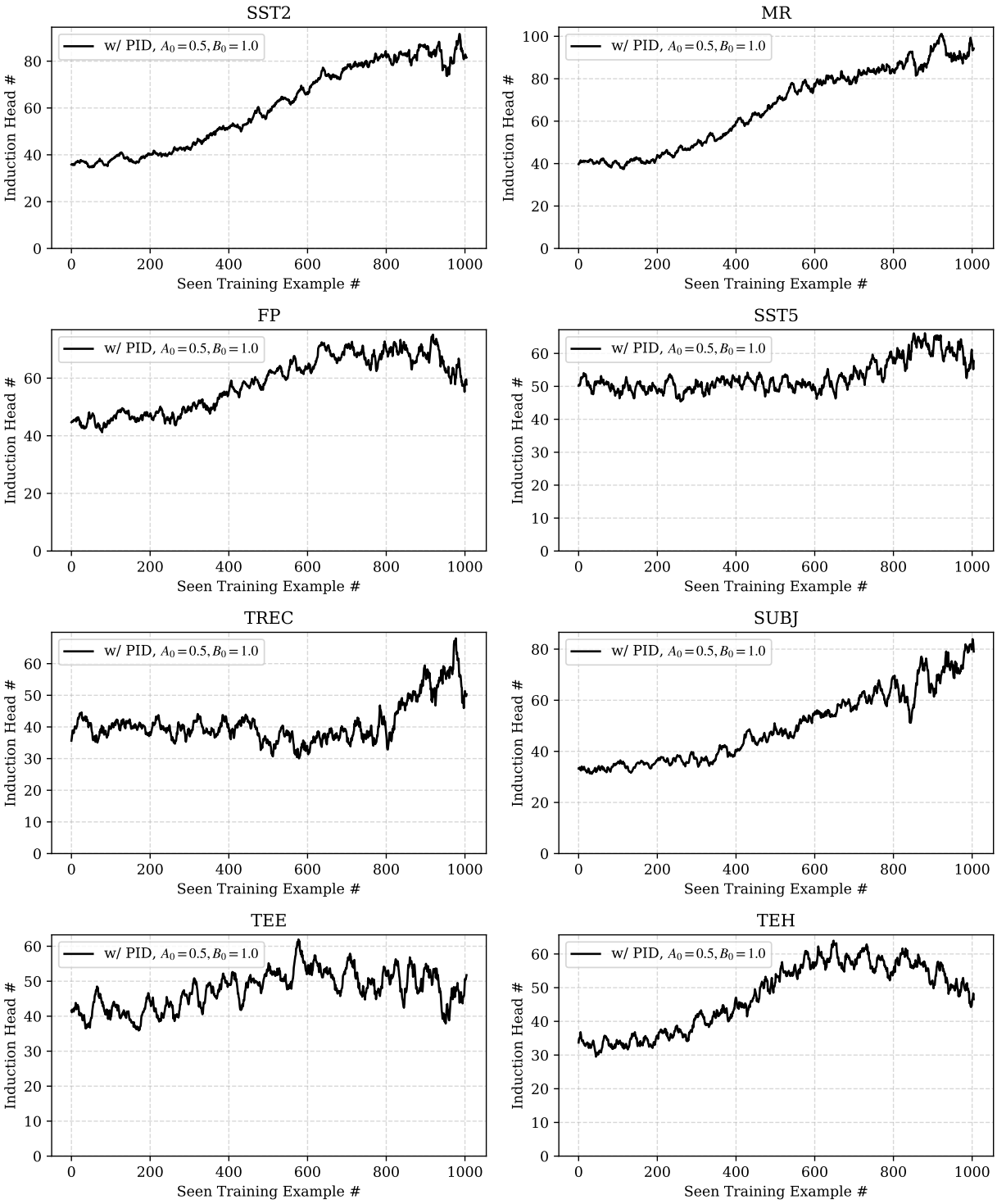



Figure 27: Induction head numbers along training dynamics on Llama3-42B and all 8 datasets.

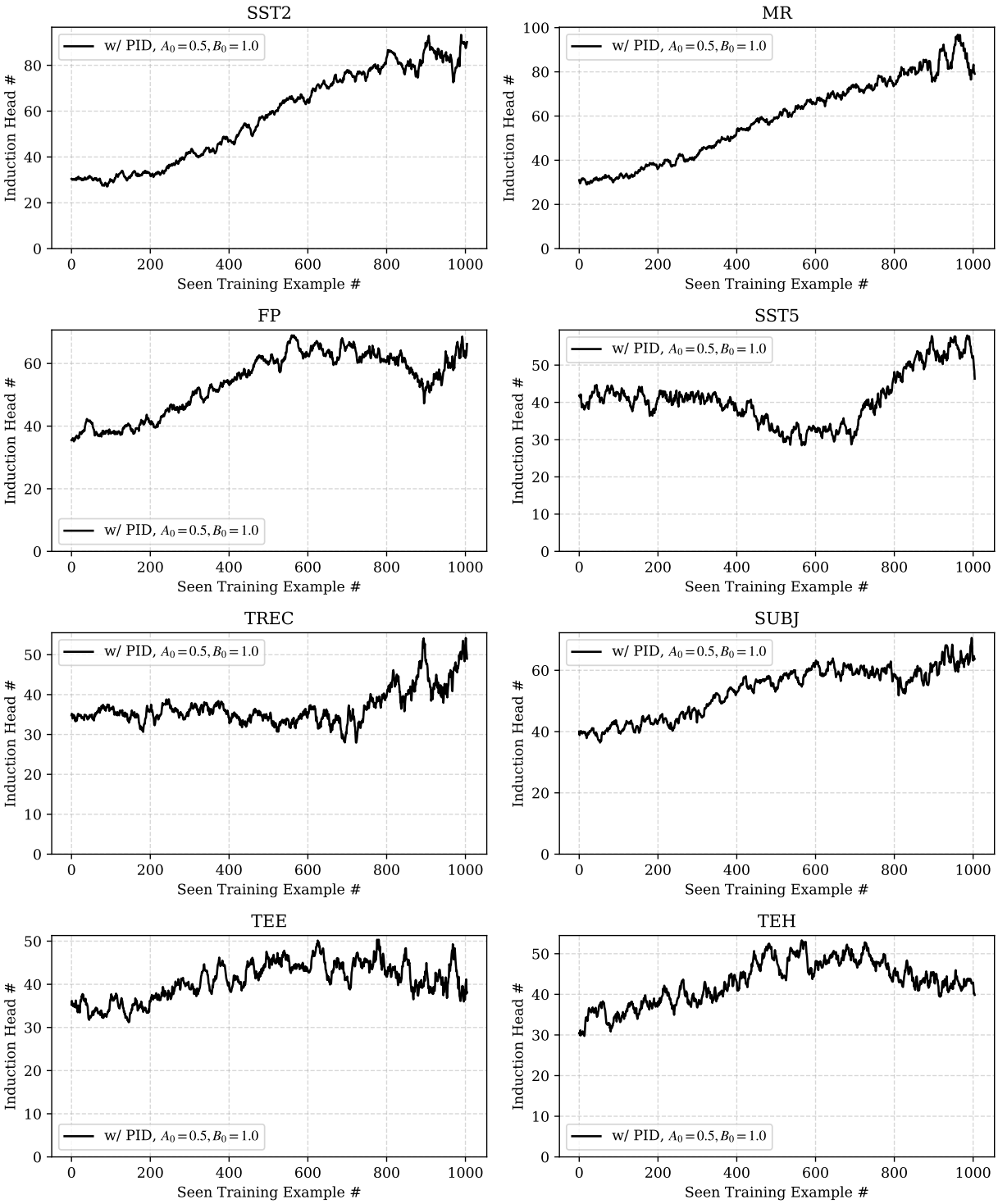



Figure 28: Induction head numbers along training dynamics on Llama3-56B and all 8 datasets.

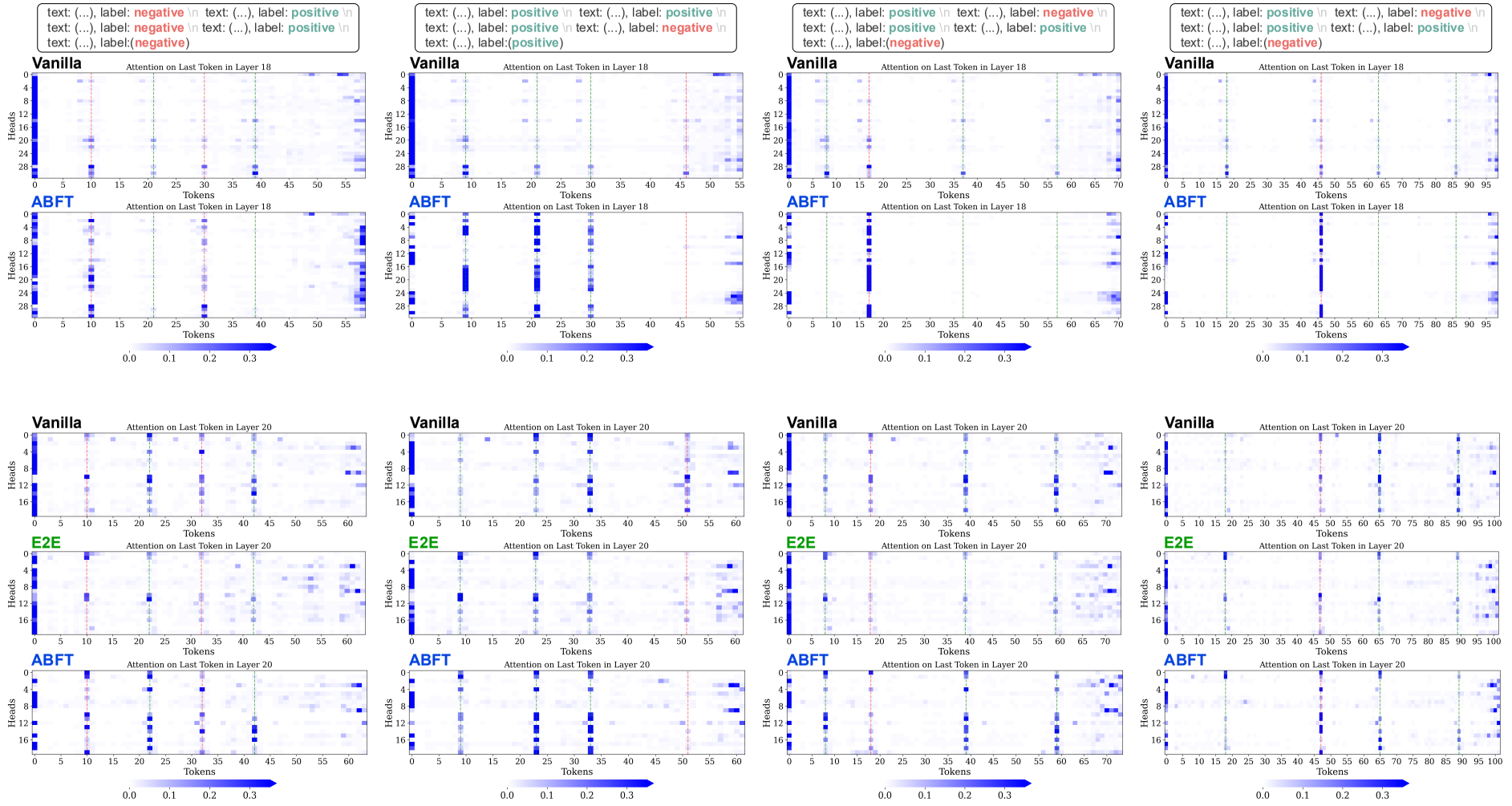



Figure 29: Augmentation results (4 cases) for attention visualization. **Upper**: results for Fig. 16, **lower**: results for Fig. 8.

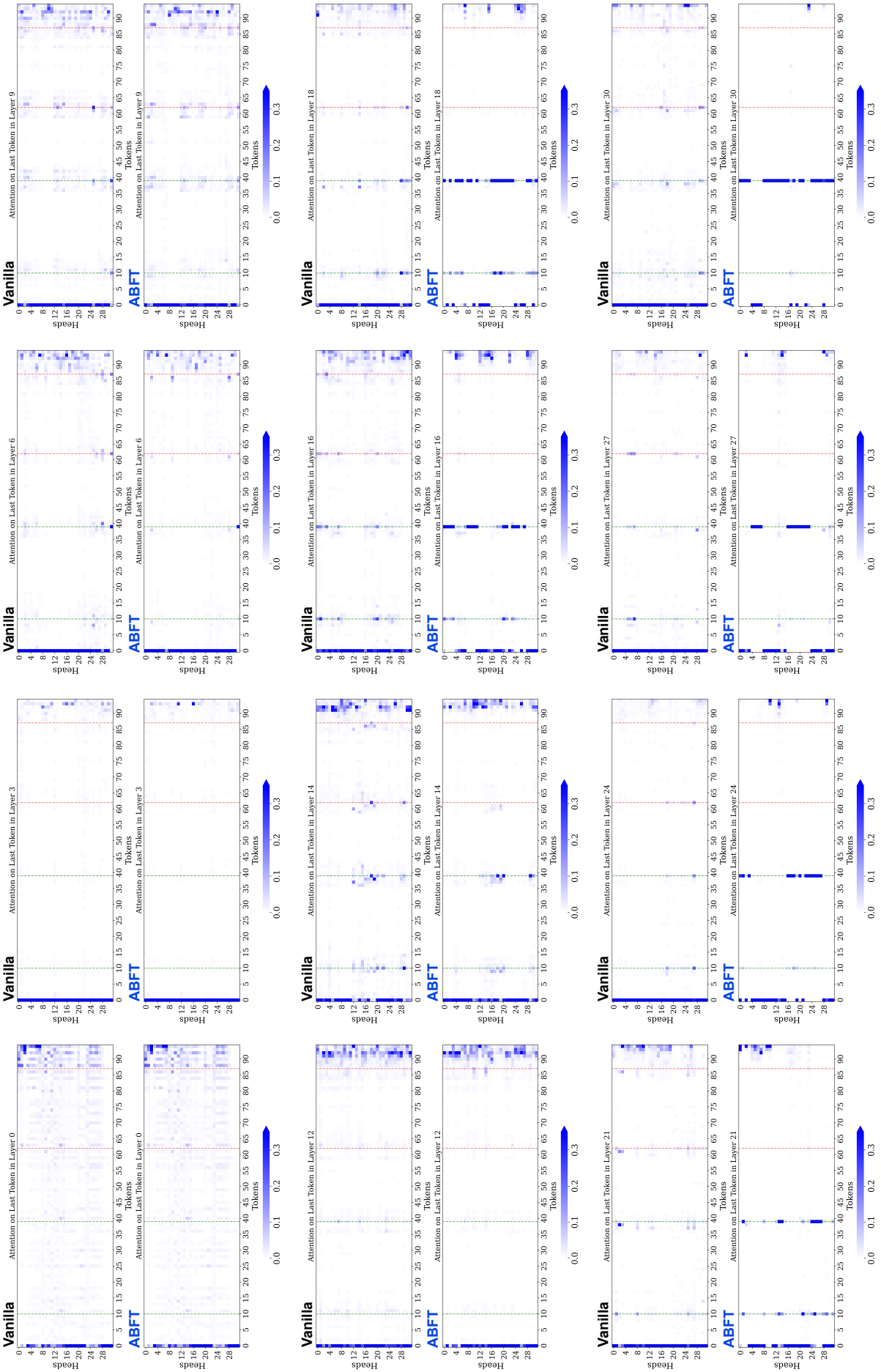



Figure 30: Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. 16.

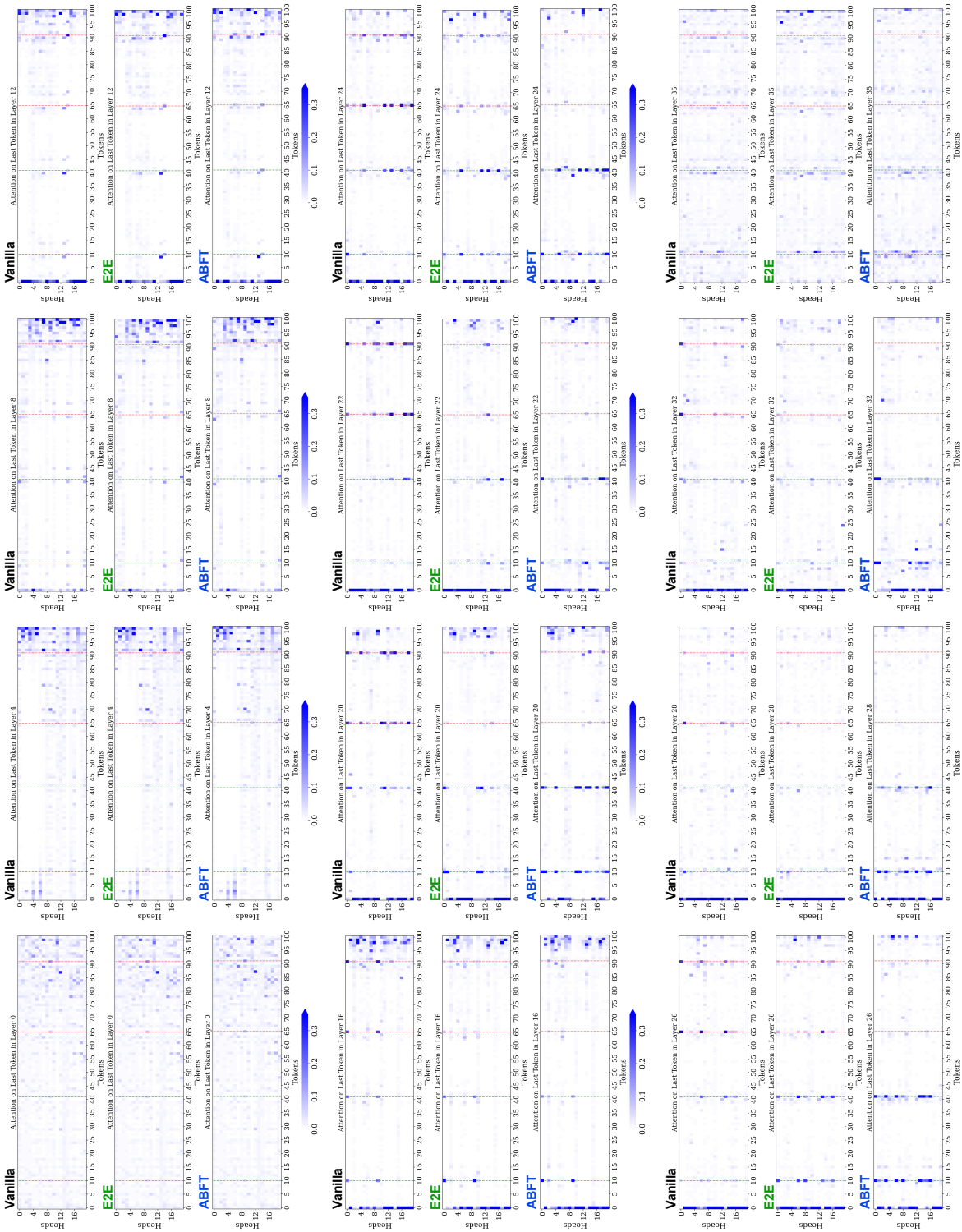



Figure 31: Augmentation results (more layers) for attention visualization with the same settings and input as the Fig. 8.