Exploring the Relationship between In-Context Learning and Instruction Tuning

Hanyu Duan,[1] **Yixuan Tang,**[1] **Yi Yang,**[1] **Ahmed Abbasi,**[2] **Kar Yan Tam**[1]
Department of Information Systems, Business Statistics and Operations Management, HKUST [1]
Department of IT, Analytics, and Operations, University of Notre Dame [2]
{hduanac, ytangch}@connect.ust.hk, imyiyang@ust.hk
aabbasi@nd.edu, kytam@ust.hk

Abstract

In-Context Learning (ICL) and Instruction Tuning (IT) are two primary paradigms of adopting Large Language Models (LLMs) to downstream applications. However, they are significantly different. In ICL, a set of demonstrations is provided at the inference time, but the LLM's parameters are not updated. In IT, a set of demonstrations is used to adjust the parameters of the LLM during training, but no demonstrations are provided at the inference time. Although a growing body of literature has explored ICL and IT, studies on these topics have largely been conducted in isolation, leading to a disconnect between these two paradigms. In this work, we explore the relationship between ICL and IT by examining how the hidden states of LLMs change in these two paradigms. Through carefully designed experiments conducted with LLaMA-2 and LLaMA-2-Chat (7B and 13B), we find that ICL and IT converge in LLM hidden states despite their apparent differences in implementation. Specifically, *ICL changes an LLM's hidden states as if its accompanying demonstrations were used to instructionally tune the model*. Furthermore, the convergence between ICL and IT is largely contingent upon several factors related to the demonstration. Overall, this work offers a unique perspective to explore the connection between ICL and IT and sheds light on understanding the behaviors of LLMs.

1 Introduction

Large language models (LLMs), such as ChatGPT [1], GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2022), and LLaMA-2 (Touvron et al., 2023), have significantly changed the paradigm of natural language processing and have great potential for artificial general intelligence (Bubeck et al., 2023). In real-world applications, the success of deploying LLMs can largely be attributed to the effectiveness of two important learning paradigms: 1) In-Context Learning (**ICL**) and 2) Instruction Tuning (**IT**). ICL, a paradigm introduced in the GPT-3 paper (Brown et al., 2020), involves utilizing a set of demonstrations provided at the inference time to guide the model's responses, but the model's parameters are not updated during this process. In contrast, IT refers to the process of further training LLMs on (*"input"*, *"output"*) pairs, along with *instructions* in a supervised fashion. IT has been shown to be effective in enhancing an LLM's generalizability on unseen tasks (Longpre et al., 2023) and a viable strategy for LLM alignment (Taori et al., 2023; Zhou et al., 2023). We illustrate ICL and IT using sentiment analysis in Figure 1.

A growing body of literature has examined the inner workings of ICL and IT, such as looking for the conditions under which ICL emerges (Liu et al., 2021; Lu et al., 2021; Su et al., 2022; Wang et al., 2023; Chan et al., 2022; Xie et al., 2021), and identifying the training data or tasks for effective instruction tuning to enhance the zero-shot generalizability of LLMs (Longpre et al., 2023). Although ICL and IT are both effective strategies for improving the capability of LLMs, studies on ICL and IT have been conducted in isolation. This has led to a research question: *What are the connections between ICL and IT, and in which way do they enhance an LLM's capability?* This question further extends a more recent discussion on *representation convergence* across training objectives (Bansal et al., 2021) and the platonic representation hypothesis (Huh et al., 2024).

In this work, we empirically examine the connections between ICL and IT from a hidden state perspective. In an autoregressive model, the hidden state of the last input token summarizes the information of the entire input sequence and determines the logit vector for sampling the following token. In the context of ICL and IT, three situations arise, each producing a different hidden state compared to the others. The first situation involves

[1] https://openai.com/chatgpt

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3197–3210

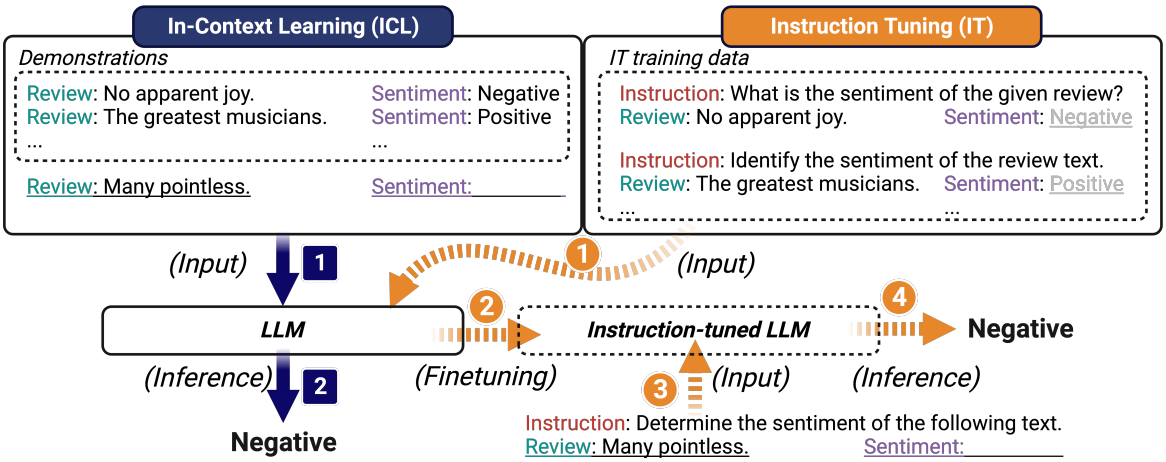


Figure 1: Illustration for ICL and IT using sentiment analysis. Through ICL, the LLM infers a *"Negative"* sentiment for *"Many pointless."* conditioned on the provided demonstrations (Left). In contrast, IT involves further tuning the LLM's parameters on the training data, and the tuned LLM is subsequently used for inference (Right).

zero-shot learning for an LLM. In this case, the hidden state of the last token in the input sequence is determined by the LLM, conditioned on the inference example. Since this is the basic case — where no demonstrations are provided and the LLM's parameters are not updated — we denote this hidden state as the anchor hidden state, h_{anchor}. The second situation is ICL, where demonstrations are provided to guide the LLM's responses. Since ICL does not modify the LLM's parameters, the hidden state is determined by the LLM, conditioned on the provided demonstrations and the inference example. We denote this hidden state as h_{ICL}. The third situation is IT, where demonstrations are used to adjust the LLM's parameters, transforming the LLM into a tuned-LLM. Here, the hidden state is determined by the tuned-LLM, conditioned on the inference example, and we denote this hidden state as h_{IT}. Comparing the similarity between h_{anchor} and h_{ICL} allows to quantify the effect of the demonstrations through ICL, while comparing the similarity between h_{anchor} and h_{IT} allows to assess the impact of IT on such demonstrations. If a demonstration is effective for ICL and IT, we would observe little similarity between h_{anchor} and h_{ICL}, as well as between h_{anchor} and h_{IT} because this demonstration gears the LLM to produce a guided (either through ICL or through IT) response. Moreover, measuring the similarity between h_{ICL} and h_{IT} allows us to quantify the extent to which ICL and IT converge in hidden states. Figure 2 illustrates our analysis framework involving the above three situations.

In our main experiment, we choose LLaMA-2 (7B) (Touvron et al., 2023) as the foundation LLM and sentiment analysis as the downstream task. We compile a data set D consisting of tuples like $(instruction, text, label)$. We then apply ICL and IT to the LLM using same demonstrations and inference example drawn from D and examine the similarities among h_{anchor}, h_{ICL}, and h_{IT}. We repeat the experiment multiple times with variations in the wording of instructions and demonstrations by sampling D repeatedly using different random seeds. The results reveal a high similarity between h_{ICL} and h_{IT}, while the similarity of each of the two hidden states with h_{anchor} is low. This suggests that ICL and IT essentially guide the LLM to a similar status, although IT tunes the LLM's parameters while ICL does not. To further investigate, we manipulate the demonstrations used in ICL and IT; for example, we vary the number of demonstrations (from one-shot ICL to few-shot ICL), alter the semantic similarity between the demonstration and the inference example, demonstrate the LLM with incorrect labels, and ask the LLM to perform a task different from the one shown in demonstrations. The results consistently support the finding that using a demonstration in ICL has a similar effect as using this demonstration to instructionally tune the LLM. In the additional analyses examining the robustness and generalizability of our findings, we change the downstream task to machine translation and text summarization, apply different tuning strategies, and replace LLaMA-2 (7B) with LLaMA-2 (13B) and LLaMA-2-Chat (7B);

the results remain consistent.

In summary, this work makes three important contributions. First, we provide empirical evidence that ICL and IT are closely related. Although ICL does not alter model parameters — unlike IT — the instructions and demonstrations they employ drive the model towards convergent hidden states. Second, we introduce a generic analysis framework that facilitates conducting controlled experiments to study LLM behaviors empirically. Third, our findings shed light on compiling dataset for efficient ICL and IT (as discussed in later sections), potentially advancing the alignment of foundation models. [2]

2 Analysis Framework

We illustrate the analysis framework in Figure 2. Building upon this framework, we examine the impact of different demonstrations (zero-shot vs. few-shot ICL) and learning paradigms (ICL vs. IT) on the model's hidden states separately. Although LLMs maintain hidden states for every input token, we primarily focus on the hidden states associated with the last input token of the sequence in this study. This is consistent with the majority of work studying how a decoder-only-LLM's internal states (activations) affect its behavior (Zou et al., 2023).

We denote the instruction as X (such as, *what is the sentiment of this review?*), demonstration as A = (Text A, Label A) (such as, *Review: This is a wonderful movie. Sentiment: Positive*), and inference example as B = (Text B) (such as, *Review: I like this movie.*). We then consider the following three situations.

Basic situation. This is the basic zero-shot learning setting where no demonstrations are provided to guide the model's inference. In this situation, we concatenate the instruction with the inference example (i.e., Instruction X + Text B) and feed it through the LLM. We collect the final hidden state of the last input token, denoted as h_{anchor}.

ICL situation. In ICL, demonstration, along with the inference example (i.e., Instruction X + Text A + Label A + Text B), are provided as input to the LLM, which then directly infers the distribution of the next token. Similarly, we collect the final hidden state of the last token, denoted as h_{ICL}. Comparing the similarity between h_{anchor} and h_{ICL} allows us to examine the effect of the provided demonstration through ICL. If the similarity is low, it indicates that the demonstration information is incorporated by the LLM so that the final hidden state is geared away.

IT situation. In IT, unlike the ICL situation where the demonstration is used as a part of the input sequence, we instead use the demonstration (i.e., Instruction X + Text A + Label A) to instructionally tune the LLM, leading to a tuned LLM. We then send the inference example (i.e., Instruction X + Text B) through the tuned LLM, and the associated hidden state is denoted as h_{IT}. Note that the input sequences to the LLM are exactly the same (i.e., Instruction X + Text B) in both the basic situation and the IT situation. The only difference is that the basic situation involves the vanilla LLM while the IT situation involves the instruction-tuned LLM. Therefore, by comparing h_{anchor} with h_{IT}, we can quantify the effect of IT with the demonstration.

Since same demonstrations are used in both ICL and IT, measuring the similarity between h_{ICL} and h_{IT} allows us to precisely quantify the convergence between them. Moreover, by manipulating the demonstrations, we can examine how closely ICL is connected to IT in different situations. In the following analyses, we denote the cosine similarity between h_{anchor} and h_{ICL} as $s_{anchor-ICL}$, and denote that between h_{anchor} and h_{IT} as $s_{anchor-IT}$. Similarly, we calculate the similarity between h_{ICL} and h_{IT}, denoted as s_{ICL-IT}, which quantifies how much ICL and IT aligns from the hidden state perspective. If s_{ICL-IT} is very high, it suggests that ICL and IT guide the model status towards the same direction although the model parameters are not updated in ICL but adjusted in IT.

3 Experiments

3.1 Experiment Setup

Datasets. In the experiments, we use SST2 for sentiment analysis (Socher et al., 2013), EN-CS of WMT16 for English-Czech translation (Bojar et al., 2016), and XSum for text summarization (Narayan et al., 2018). For each task, we manually craft the relevant instructions which are then employed randomly in the repeated experiments, alleviating the concern that the experimental results are driven by a specific instruction. Instructions used for each task are presented in Appendix C.

LLMs. We use LLaMA-2-Base/Chat as the foundation model (Touvron et al., 2023), including 7B (32 layers with a hidden size of 4,096) and 13B (40

[2]The code is available at https://github.com/hduanac/ICL-vs-IT

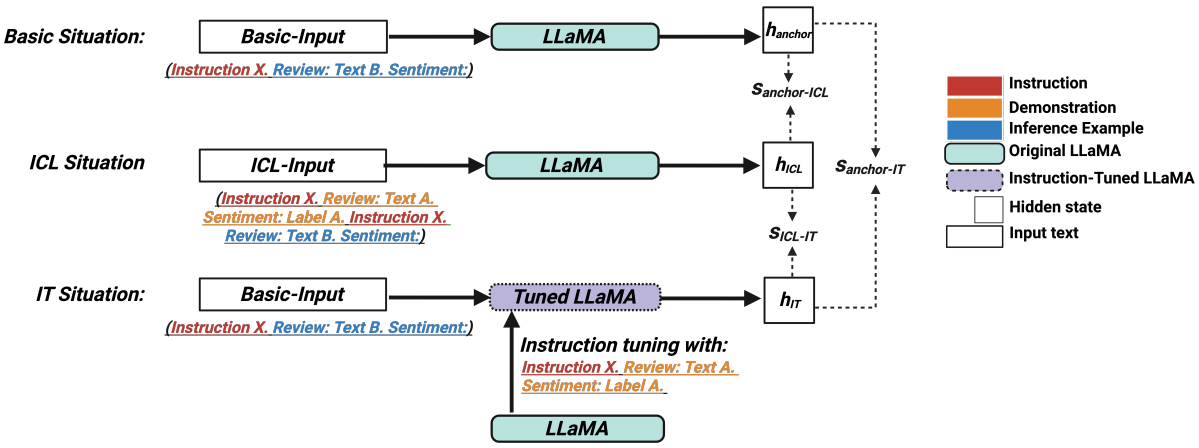


Figure 2: Analysis framework using sentiment analysis for illustration. The framework accommodates variations by manipulating the instructions, demonstrations, and changing the foundation model and the downstream task.

layers with a hidden size of 5,120). The models are downloaded from Meta AI [3], and implemented using the `transformers` library [4].

Instruction tuning. We use LoRA (Hu et al., 2021) to instruction-tune the LLaMA-2 model due to its efficiency. Specifically, we target modules Q and V [5], and set the dropout probability 0.05, learning rate 1e-4, scaling factor 32, and use a rank of 8. We use AdamW optimizer (Loshchilov and Hutter, 2017). Without further specification, we tune the model with 10 epochs and use *bf16* precision.

Repeated experiments. Our empirical findings are based on 30 repeated runs of experiments with different random seeds, differing in instructions, demonstrations, and the inference example.

3.2 Empirical Findings

We present the empirical findings as follows.

In-Context Learning (ICL) and Instruction Tuning (IT) result in a converged model state. We show the cosine similarity between each pair of hidden states in Figure 3a. First, we observe that the similarity between h_{anchor} and either h_{ICL} or h_{IT} is almost zero, indicating that the model undergoes significant changes in its hidden representations when exposed to in-context demonstrations or when tuned by the demonstrations. Furthermore, the high similarity between h_{ICL} and h_{IT} (approximately 0.9) suggests that the model is indeed oriented toward a similar state in ICL and IT. This provides the first evidence that ICL acts as implicit IT regarding the behavior within hidden representations. We provide more supporting evidence by analyzing the attention allocation in Appendix A.

Prior studies have shown that high-quality output can be achieved with minimal training data by IT (Zhou et al., 2023), highlighting the importance of developing efficient data selection techniques for instruction tuning. Along this line of research, our finding suggests that, without any training, comparing hidden states in an ICL setting may offer a viable solution for pinpointing redundant data in instruction tuning. This aligns closely with the recent research endeavors into selecting efficient IT data with minimal computational cost without model update (Xia et al., 2024). Besides, Lin et al. (2023) suggest another potential application of this finding by demonstrating that strategic prompting with ICL can achieve comparable performance to IT in terms of foundation model alignment.

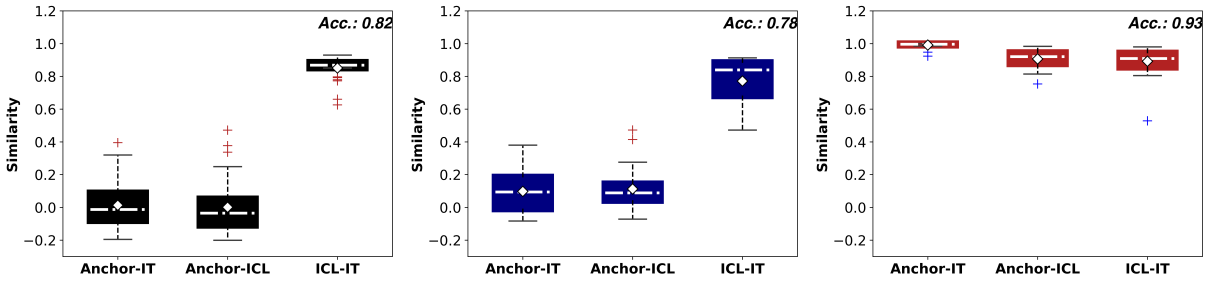
The convergence between ICL and IT is positively correlated with the semantic similarity between the demonstration and the inference example. We further investigate how the semantic similarity between the demonstration and the inference example (i.e., Text A and Text B respectively in Figure 2) affects the ICL-IT convergence. To do this, we use the sentence-transformer model "all-MiniLM-L6-v2" [6] to measure the demonstration-inference similarity (Reimers and Gurevych, 2019). We consider 10 levels of similarity ranging from 0 to 1 and experiment with same inference examples across the 10 similarity levels to allow fair

[3] https://ai.meta.com/resources/models-and-libraries/llama-downloads/

[4] https://github.com/huggingface/transformers

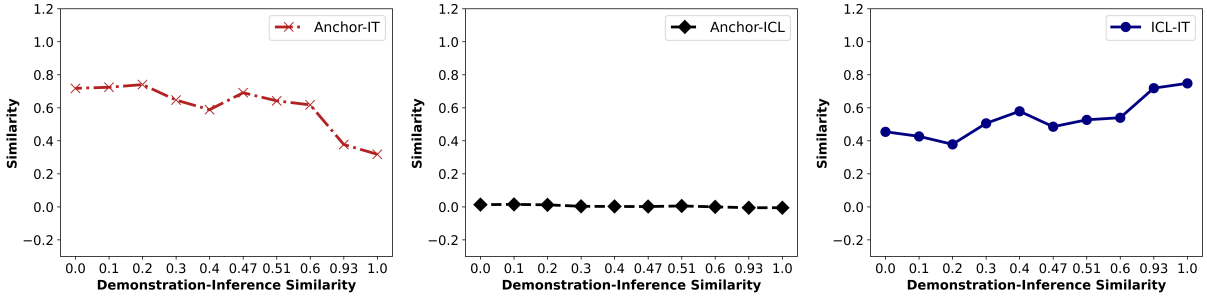
[5] Please refer to Appendix B.2 for the results regarding tuning other modules.

[6] https://www.sbert.net



(a) ICL-IT Convergence (b) Incorrect Demonstration Labels (c) Different Demonstration Task

Figure 3: Hidden state similarity. Each box plot summarises the similarity scores of 30 repeated experiments. We report the average (ICL and IT) classification accuracy in the upper right corner of each plot.



(a) Anchor-IT (b) Anchor-ICL (c) ICL-IT

Figure 4: Averaged similarity scores between hidden states across 10 demonstration-inference similarity levels.

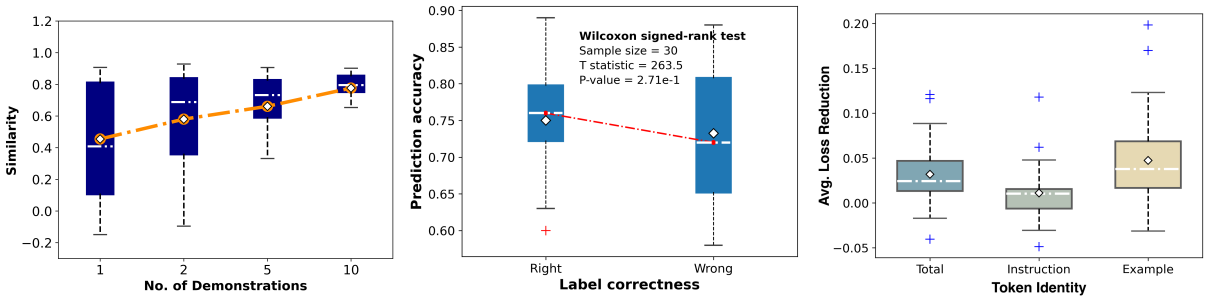
comparison. The results are shown in Figure 4. Clearly, the similarity between h_{ICL} and h_{IT} increases as the similarity between the demonstration and the inference example increases (Figure 4c). One potential explanation is that a demonstration closely resembling the inference example might better stimulate the model's ICL ability and enhance the effectiveness of IT, ultimately resulting in higher convergence. It is worth noting that the similarity between h_{ICL} and h_{IT} varies significantly, spanning from approximately 0.4 when the demonstration and the inference example are entirely dissimilar to 0.8 when they are identical.

In contrast, the similarity between h_{anchor} and h_{IT} exhibits an opposite trend, as shown in Figure 4a, suggesting that a demonstration that is more similar to the inference example is able to change the model's state to a greater extent through IT. This finding is consistent with previous studies indicating that instruction tuning becomes more effective when finetuning the LLM on examples similar to the inference instances (Gudibande et al., 2023). Put it another way, finetuning the model with semantically different examples does not substantially alter the model's inference capability.

Interestingly, we observe that the similarity between h_{anchor} and h_{ICL} remains consistently low, regardless of the demonstration-inference similarity, as illustrated in Figure 4b. This suggests that incorporating demonstrations into the ICL input can consistently and significantly impact the model's inference. Previous studies on ICL have indicated that higher demonstration-inference similarity leads to improved inference accuracy. It is important to emphasize that Figure 4b does not contradict this observation; in fact, a low similarity may very likely imply that the model's inference ability is improved with ICL.

The convergence between ICL and IT increases as the number of provided demonstrations increases. In the previous analyses, we use a single demonstration in both ICL and IT. In this experiment, we vary the number of demonstrations (i.e., the few-shot setting) in ICL and IT. Specifically, we experiment with 1-shot, 2-shot, 5-shot, and 10-shot scenarios. For a fair assessment, we keep the number of parameter updates consistent by instruction-tuning the model with 10, 5, 2, and 1 epoch(s), respectively. Besides, across these few-shot settings, we use the same set of inference examples to ensure a fair comparison.

We present the results in Figure 5a and observe a clear upward trend in convergence between ICL and IT as we utilize more demonstrations. This is



(a) Multiple Demonstrations (b) Wrong Labels vs. Right Labels (c) Loss Reduction by Token Identity

Figure 5: Miscellaneous results involving manipulated demonstrations and token loss analysis.

intuitive since ICL with multiple demonstrations (i.e., the few-shot setting) can better help the model discover patterns in the context and quickly adapt to the required task. Similarly, using more examples related to the same task for IT can better tune the model for that specific task, resulting in a higher level of convergence between ICL and IT.

Demonstrated with wrong labels slightly affects the convergence between ICL and IT. Prior studies on ICL have shown that the correctness of the demonstration label is not crucial; what matters most is the task format for ICL (Min et al., 2022). Therefore, it prompts us to examine how the label correctness of demonstrations affects the ICL-IT convergence. To this end, we invert the labels of demonstrations (e.g., switching "Positive" to "Negative"), and repeat the ICL and IT procedures. The results are shown in Figure 3b.

Interestingly, we find that while ICL and IT still exhibit a high level of convergence, the degree is slightly lower than their counterpart when using correct labels as compared to the results shown in Figure 3a. Moreover, the variation in the degree of ICL-IT convergence increases significantly, as evidenced by the larger interquartile range and longer whiskers of the box plot. As a sanity check, we examine if using wrong labels to do IT hurts the model performance, and present the results in Figure 5b. Although we do observe a performance drop, the decrease is not statistically significant [7], aligned with (Kung and Peng, 2023).

Demonstrating the LLM with a different task than that at the inference time would not affect the ICL-IT convergence. In the previous experiments, the demonstration task and the inference task are identical, focusing on sentiment analysis. In this experiment, we alter the demonstration task to machine translation using the EN-CS subset of WMT16, translating English to Czech [8], while maintaining sentiment analysis as the inference task. We present the results in Figure 3c. Clearly, the notable convergence in similarity between ICL-IT, Anchor-ICL, and Anchor-IT suggest that the machine translation task demonstrations do not affect the model's inference capability for sentiment analysis.

The convergence between ICL and IT begins to rise in later transformer layers. Unlike our previous analyses, where we measure the hidden state similarity in the final transformer block, this experiment examines such similarity across different transformer layers. The results are shown in Figure 6. Interestingly, we notice a U-shaped pattern across various layers. The high similarity between ICL and IT in the earlier layers is primarily due to the fact that the hidden states are both similar to the anchor hidden state, indicating minimal influence from the demonstrations. As the layer count increases, the LLM progressively incorporate influences from the demonstrations, leading to lower similarity between ICL and IT in the middle layers. Eventually, as the input progresses towards the upper layers, which are nearer to the final output, the hidden states of ICL and IT begin to converge.

4 Additional Analyses

4.1 LLaMA-2 (13B)

We examine if ICL and IT still converge in a larger model. We choose LLaMA-2 (13B) as the foundational model and repeat the analysis procedure to

[7] We conduct a one-tailed Wilcoxon signed-rank test, where the null hypothesis posits that there is no accuracy improvement by using correct demonstration labels compared to using incorrect ones.

[8] We use the following template: *"Instruction X. English: English text A. Czech: Czech text A. Instruction X. English: English text B. Czech:"*.

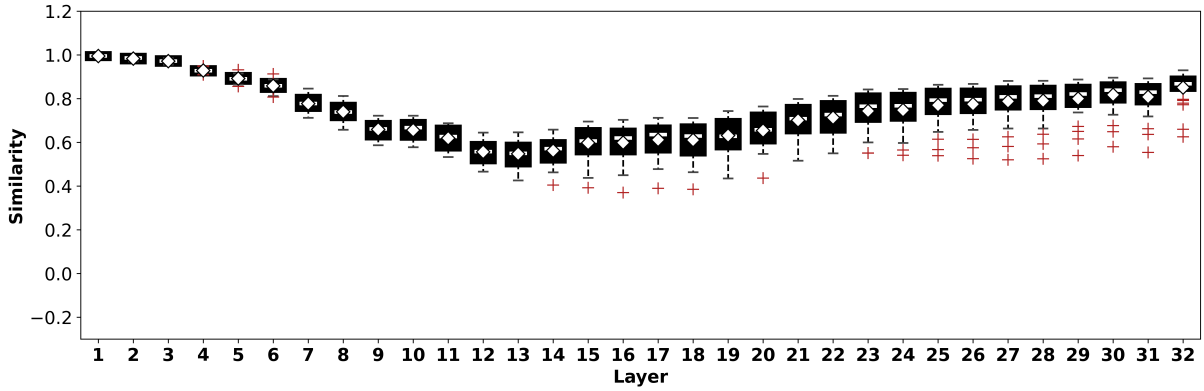
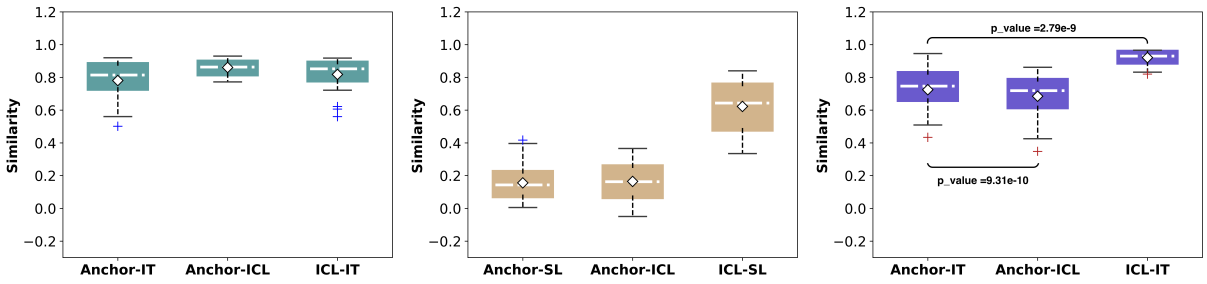


Figure 6: ICL-IT convergence across different transformer layers.



(a) LLaMA-2 (13B) (b) Classic Supervised Learning (c) Machine Translation Task

Figure 7: Hidden state similarity measured in different model, downstream task, and training configurations.

assess the similarity for each pair of hidden states. The results, displayed in Figure 7a, indicate the ICL-IT convergence persists at a high level. However, Anchor-IT and Anchor-ICL also exhibit a high level of convergence, meaning that the larger model is more capable of understanding the task even without any demonstrations given (note that in the basic situation, an instruction is given, which could provide sufficient information for the larger LLM to do zero-shot learning). We also experiment with the instruction-tuned LLaMA-2-Chat (7B) and present our findings in Appendix B.4.

4.2 Classic Supervised Learning

Instruction tuning differs from classic supervised learning in that the former employs additional instructions to enhance an LLM's generalizability, while classic supervised learning typically teaches the LLM to specialize in a specific task.

To further understand the role of instruction in IT and its influence on the ICL-IT convergence, we conduct experiments with classic supervised learning for comparison. Specifically, we remove the Instruction X from the training input and use only task examples to finetune the LLM. We denote this classic supervised situation as SL. We repeat the same analysis procedure and present the results in Figure 7b. Clearly, while the convergence between ICL and SL still exists, the similarity score is significantly lower than that of its IT counterpart (as shown in Figure 3a). This observation underscores the critical role of instructions in triggering ICL-IT convergence within the representation space.

4.3 Understanding IT from ICL

Our findings discussed above suggest that ICL might implicitly perform IT. In this subsection, we take an opposite view and look at IT through the lens of ICL instead.

Specifically, we examine the loss associated with each token, referred to as *per token loss*, which is defined as the cross-entropy loss between every predicted token and its respective ground truth token within a sequence (Olsson et al., 2022). We depict our analysis procedure in Figure 8, comprising three steps. First, employing the template used in Figure 2, we compile the input with an Instruction X and an Example A in the format: *"Instruction X. Review: Text A. Sentiment: Label A."*, then pass it through the LLM (LLaMA-2

(7B)) to collect the per token loss. Second, we instruction-tune the model with Example A using Instruction X, send the same input through the tuned model, and once again collect the per token loss. Finally, we compute the loss reduction for each token and average them based on the token's identity (i.e., "Instruction" or "Example"). We repeat this process 30 times, varying the Instruction X and Example A each time. We plot the average loss reduction by token identity in Figure 5c, which reveals a more substantial decrease in loss for "Example" tokens compared to "Instruction" tokens. Note that this is not straightforward, as during finetuning, only the loss associated with the label is minimized, while the losses of the "Example" tokens are not directly targeted. This finding suggests that IT facilitates the LLM in continuing instructions with task-relevant examples (demonstrations). Intuitively speaking, after IT, presenting the LLM with the instruction as if it were shown the demonstrations, akin to the setting of ICL where demonstrations are provided within the context.

4.4 Generalizability of Findings

One remaining concern is that the convergence of ICL and IT may stem from them making highly similar predictions, due to the saturation to the simple binary sentiment classification task. Although we only tune the LLM with a single or a few examples in the experiments, which may somewhat alleviate this concern, to fully rule out this possibility and as a robustness check, we experiment with more complex generative tasks, specifically machine translation from English to Czech using EN-CS of WMT16 (Bojar et al., 2016) and text summarization using XSum (Narayan et al., 2018).

We show the results of the translation task in Figure 7c and include the summarization outcomes in Appendix B.1. First, the high similarity between ICL and IT supports our earlier finding that ICL, when using demonstrations, significantly alters an LLM's inference capability in a manner similar to finetuning the LLM with such demonstrations. Unlike in sentiment analysis, where Anchor-IT and Anchor-ICL show as low as zero similarity, their counterparts are much higher in the two generative tasks. However, a one-tailed Wilcoxon signed-rank test reveals that the similarity between ICL and IT is significantly greater than that of Anchor-IT and Anchor-ICL, ruling out the possibility that all three hidden states are similar to each other. We provide additional robustness checks by varying the model, tuning strategy, and the wording of prompt in Appendix B.

5 Related Work

In-Context Learning (ICL) is a phenomenon emerged in large language models (Brown et al., 2020). A growing body of literature has investigated the ICL phenomenon in LLMs. Some studies have focused on identifying the conditions under which ICL emerges in LLMs, predominantly by finding good demonstrations (Liu et al., 2021; Lu et al., 2021; Su et al., 2022; Wang et al., 2023) and identifying pre-training data distributions that can lead to the emergence of ICL (Chan et al., 2022; Xie et al., 2021). Another line of research aims to explain ICL through building the relationship with the model training stage (Akyürek et al., 2022; Dai et al., 2022; Li et al., 2023; Von Oswald et al., 2023). For instance, Akyürek et al. (2022) find ICL implicitly updates smaller models encoded in the activations. Olsson et al. (2022) provide evidence that the so-called "induction heads" contribute to the majority of the ICL behaviors in LLMs.

Our work differs from existing studies in two ways. First, we attempt to understand ICL by investigating its connection with IT, which is new and opens up the possibilities for harnessing the complementary advantages of ICL and IT. Second, we empirically study off-the-shelf LLMs with more complex model structures (LLaMA-2 7B and 13B), whereas most prior works conduct experiments using simplified models (Li et al., 2023).

Instruction Tuning (IT) is an efficient technique to adapt LLMs to downstream tasks by further tuning the model on (*"input"*, *"output"*) pairs with instructions in a supervised manner. The intuition behind IT is to bridge the gap between the language modeling objective in pre-training and the users' objective in downstream tasks, such that the model can follow the instructions from users. The effectiveness of IT is well-demonstrated by a variety of instruction-tuned LLMs, with representatives such as InstructGPT (Ouyang et al., 2022), Alpaca (Taori et al., 2023), Flan-T5 (Longpre et al., 2023), and Vicuna [9]. A growing body of literature focuses on designing tasks and datasets for effective instruction tuning. For example, LIMA (Zhou et al., 2023) shows that a small set of high-quality instruction data is sufficient for foundation model alignment. Our work aims to provide empirical

[9] https://lmsys.org/blog/2023-03-30-vicuna/

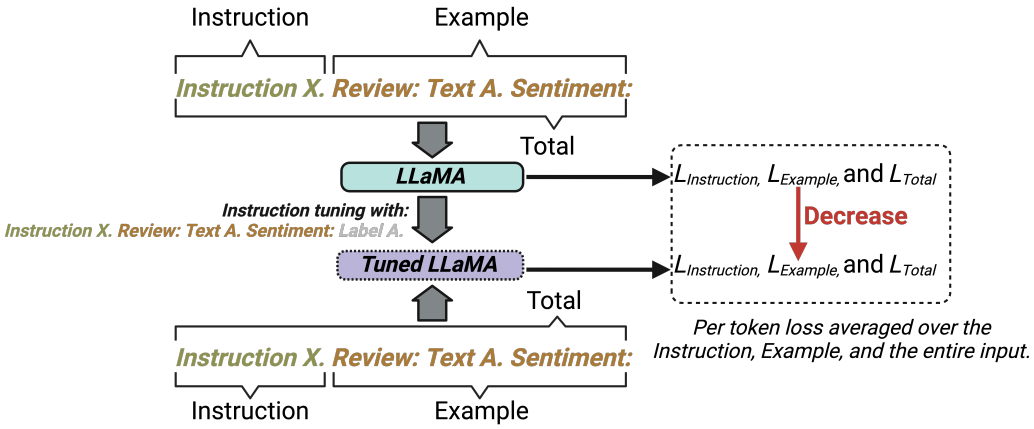


Figure 8: Illustration of collecting per token loss for understanding IT through the lens of ICL.

evidence to further understand IT, through the lens of its connection with ICL.

6 Conclusions

In this work, we explore the connection between in-context learning (ICL) and instruction tuning (IT). Through carefully designed experiments, we provide strong evidence suggesting ICL and IT implicitly converge in LLM hidden states. That is, ICL changes an LLM's hidden states as if the demonstrations were used in IT. This finding sheds light on the behaviors of the two very different learning paradigms (ICL vs. IT), potentially benefiting the development and alignment of foundation LLMs to downstream real-world applications.

7 Limitations

This work has several limitations that can be improved in future research. First, we only examine the ICL-IT relationship by measuring the hidden state similarity, since this is our main focus. Future work can build upon our analysis framework and further investigate the ICL-IT convergence from distinct angles. For example, in Appendix A, we provide preliminary evidence suggesting that ICL and IT lead to similar attention weights over the inference example. Potential avenues include identifying similar neuron activation patterns, delving into the QKV matrices, and studying how the encoded knowledge is used by ICL and IT through topic modeling. Second, we do not look into the hidden states in the model's intermediate layers and those of other input tokens in detail, since the final state of the last token is much more crucial and directly influences the model's response. Delving deeper into the intermediate layers and examining the representations of other tokens may constitute an interesting future direction. Third, our experiments are conducted on common NLP tasks (i.e., sentiment analysis, machine translation, and text summarization). How the analysis framework can be generalized to other real-world NLP settings where the tasks are more complex, such as mathematical reasoning, or perhaps even going beyond textual data to incorporating multimodal features, warrants further investigation. Finally, as we mention earlier, our empirical findings offer great potential to contribute to efficient ICL and IT. Future work may be needed to design more efficient ICL and IT datasets, tasks, and learning strategies for better deploying and aligning foundation models to downstream applications drawing upon the insights from this work.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot

learners. *Advances in neural information processing systems*, 33:1877–1901.

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

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. *arXiv preprint arXiv:2305.11383*.

Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *arXiv preprint arXiv:2301.07067*.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

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*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Understanding ICL-IT Convergence from Attention Allocation

Besides comparing hidden states in our main analysis, we take a distinct angle by looking into the attention heads within LLMs. Specifically, for each situation (i.e., basic, ICL, and IT), we collect their respective attention scores associated with the last input token over the inference example from the final transformer block. We concatenate the attention scores of the 32 attention heads, leading to a_{anchor}, a_{ICL}, and a_{IT}. We present the cosine similarity between each pair of them in Figure 9 (averaged over 30 repeated experiments with LLaMA-2 (7B) of the sentiment analysis task). The results suggest that ICL and IT drive the model to put similar attention weights over the tokens of the inference example. This lends additional credence to our earlier finding of the ICL-IT convergence, now examined through a unique lens of attention allocation.

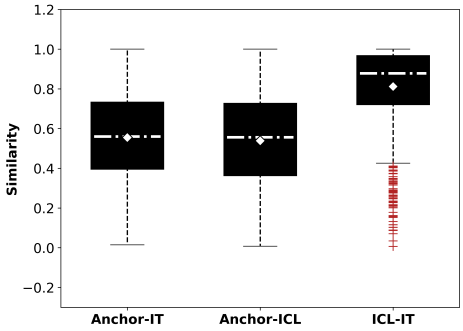


Figure 9: Similarity between attention allocations.

B Robustness Check

B.1 Text Summarization

We replace the sentiment analysis task (inference task) with a text summarization task (generation task), and implement the same procedure to examine whether the relationship between ICL and IT still holds. We use the XSum dataset that summarizes news articles (Narayan et al., 2018). The instructions used can be found in Table 3. The results are shown in Figure 10. Clearly, the high level of similarity between the ICL state and IT state maintains, whereas the similarity of each of them with the Anchor sate is relative low. The results further support our earlier finding that ICL, using the demonstrations, orients the LLM's inference ability as if using such demonstrations to instructionally tune the model.

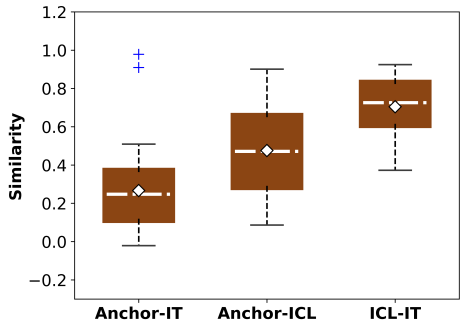


Figure 10: Text summarization task.

B.2 LoRA Target Modules

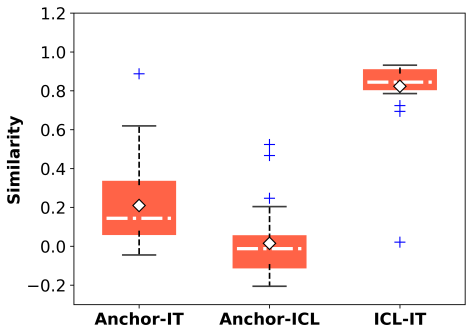
In the main experiment, as done in the original LoRA paper (Hu et al., 2021), we finetune the Q and V attention matrices of the LLM. Here, we vary the tuning module by targeting 1) the feed-forward module (i.e., $gate\_proj$, up_proj, and $down\_proj$), and 2) the feed-forward module as well as the attention module. We repeat the same procedure as depicted in Figure 2, and present the results in Figure 11a and 11b. Consistent with our earlier observations, we find the similarity between the ICL state and IT state is remarkably high, meaning that the ICL-IT convergence is not driven by fine-tuning on specific target modules.

B.3 The Last Token

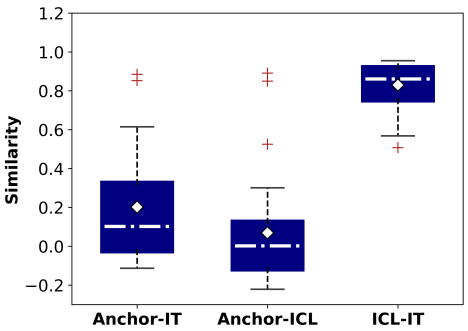
Given that our conclusions are based on comparing the last input token in a sequence, the impact of its semantic meaning on the ICL-IT convergence is also an important consideration. To rule out the possibility that the convergence is driven by a particular token appearing in the last position, we replace the last token with 1) "*is*" and 2) ">" respectively, and repeat the experiment. The results are shown in Figure 12a and Figure 12b, respectively, revealing similar patterns.

B.4 LLaMA-2-Chat

We consider LLaMA-2-Chat (7B) and repeat the same procedure to measure the similarity between hidden states. Unlike LLaMA-2-Base, LLaMA-2-Chat is instruction-tuned for dialogue use cases. The results appear in Figure 13. Different from the base version (Figure 3a), we observe higher similarity for Anchor-IT and Anchor-ICL, suggesting that the influence of either ICL or IT, conditioned on the demonstrations, is relatively small. This is expected since the chat version is instruction-tuned to better understand and follow instructions, even without exposure to demonstrations. Although both

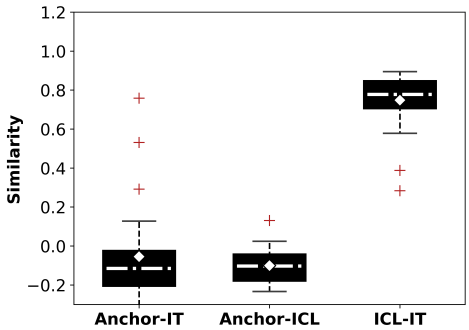


(a) Feed-forward Module

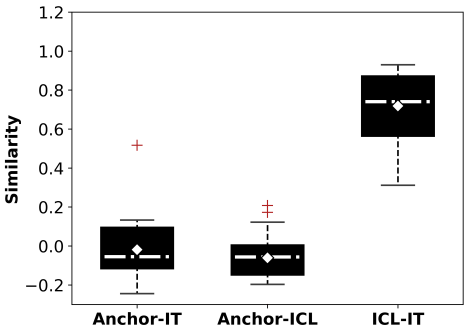


(b) Feed-forward Module + Attention Module

Figure 11: Hidden state similarity across targeting different modules for LoRA.



(a) "*is*" as the Last Token



(b) ">" as the Last Token

Figure 12: Hidden state similarity across different last tokens.

the ICL and IT hidden states do not deviate much from the Anchor hidden state, the ICL hidden state is still significantly more similar to the IT hidden state relative to the Anchor one. This further supports our earlier notion that ICL and IT result in similar model status.

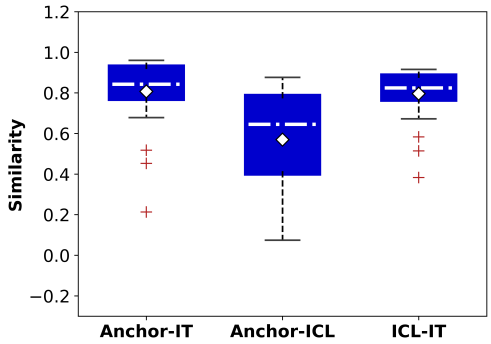


Figure 13: Hidden state similarity (LLaMA-2-Chat).

C Instruction Sets

Can you express this English phrase in Czech?
Can you present this English sentence in Czech?
Please make this English sentence into a Czech sentence.
Please convert this English text into Czech.
Help me interpret this English phrase in Czech.
Translate this English sentence into Czech.
Please provide a Czech translation for this English sentence.
I need your help to change this English sentence into Czech.
Could you help convert this English phrase into Czech?
Could you translate this English text into Czech?
Please, translate the following English sentence into Czech.
Rephrase this English sentence in Czech for me, please.
Please give me the Czech version of this English sentence.
Can you assist in translating this English sentence into Czech?
Can you change this English sentence into Czech?
How would you say this English sentence in Czech?
Please convert this English phrase into Czech.
Can you convert this English sentence into Czech, please?
Please interpret this English sentence into Czech for me.
Please provide a Czech version of this English sentence.
Can you give me a Czech translation of this English text?
Could you kindly convert this English text into Czech?
Could you rewrite this English phrase in Czech?
I require this English sentence to be translated to Czech.
I need this English phrase translated to Czech.
Translate this English content into the Czech language, please.
Translate this English phrase into Czech for me, please.
Can you provide a Czech interpretation of this English sentence?
Can you render this in the Czech language, please?
Can you transcribe this English text into Czech?

Table 1: Instructions for machine translation.

What is the sentiment of the movie review below? Is it negative or positive?
Determine whether the sentiment expressed in this movie review is negative or positive:
Identify whether this movie review contains negative or positive opinions.
Classify whether this movie review conveys negative or positive opinions.
Rate whether the viewpoint on the costumes is more negative or positive.
Based on the review content, would you say the sentiment is negative or positive?
Analyze the sentiment expressed in this movie review. Is it positive or negative?
Identify negative or positive of the content.
Evaluate the sentiment of this movie critique. Is it negative or positive?
Determine the sentiment conveyed in this movie review. Is it negative or positive?
Classify the overall sentiment of this movie review as negative or positive.
Determine if the tone of this movie review is negative or positive.
Assess if the tone of this movie review is negative or positive.
Detect whether this movie review contains negative or positive sentiment.
Determine whether this movie review expresses negative or positive sentiment.
Identify whether the sentiment expressed in this movie review is negative or positive.
Distinguish whether the evaluation in this movie review is negative or positive.Provide your answer as either negative or positive:
Infer whether the tone of this movie review is negative or positive.
Grade if the perspective in this movie review is negative or positive.Provide your answer as either negative or positive:
What's the emotional tone of this movie review? Would you describe it as negative or positive?
Infer whether this movie review expresses negative or positive emotion.
Estimate if the analysis in this movie review is negative or positive.Provide your answer as either negative or positive:
Determine whether the opinions in this movie review are negative or positive.
Identify the sentiment of the following movie review text. Is it negative or positive?
Assess the sentiment expressed in the following movie review. Is it positive or negative?
Determine the sentiment expressed in this movie review. Negative or positive?

Table 2: Instructions for sentiment analysis.

Provide a brief synopsis of the key events in this news cycle.
Please give me a summary of the content in this document.
Provide a short summary highlighting the important information in this news journal.
Provide a simplified summary of this news analysis piece.
Can you give me a high-level summary of this news channel's coverage?
Could you give a snapshot summary of this article?
Could you provide a short summary of the critical details from this news segment?
Please summarize this news article for me.
Summarize the critical information I need to know from this news.
Please summarize this breaking news update, hitting only the key points.
Summarize just the main points from this news outlet's live coverage.
Provide a simplified rundown of the top headlines across these news sites.
Give me a recap of the main points from this news column.
I need the main takeaways from this news report summarized.
Please condense this news interview into a short summary for me.
Please make a brief summary of this document.
Summarize this news article in your own words.
I need a short synopsis of the key ideas in this press release from a news agency.
Could you present a short summary of this article?
Provide a broad overview summarizing this breaking news alert.
Give me a quick high-level summary of this news commentary.
Give me a high-level summary of this developing news event.
Please summarize this manuscript for me.
Could you simplify this text into a summary?
Please summarize the key takeaways from this news dataset.
Outline the important details from this newscast.
Summarize the crucial details from this daily newspaper.
Give me a quick abstract of the concepts in this opinion editorial.
I need the main takeaways from this news article summarized.
Can you summarize the key content from this news podcast?

Table 3: Instructions for text summarization.