Unveiling In-Context Learning: A Coordinate System to Understand Its Working Mechanism

Anhao Zhao
Southwest Jiaotong University
zhaoanh@my.swjtu.edu.cn

Fanghua Ye
University College London
fanghua.ye.19@ucl.ac.uk

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Xiaoyu Shen
Digital Twin Institute
Eastern Institute of Technology, Ningbo
xyshen@eitech.edu.cn

Abstract

Large language models (LLMs) exhibit remarkable in-context learning (ICL) capabilities. However, the underlying working mechanism of ICL remains poorly understood. Recent research presents two conflicting views on ICL: One emphasizes the impact of similar examples in the demonstrations, stressing the need for label correctness and more shots. The other attributes it to LLMs' inherent ability of task recognition, deeming label correctness and shot numbers of demonstrations as not crucial. In this work, we provide a **Two-Dimensional Coordinate System** that unifies both views into a systematic framework. The framework explains the behavior of ICL through two orthogonal variables: *whether similar examples are presented in the demonstrations* (**perception**) and *whether LLMs can recognize the task* (**cognition**). We propose the peak inverse rank metric to detect the task recognition ability of LLMs and study LLMs' reactions to different definitions of similarity. Based on these, we conduct extensive experiments to elucidate how ICL functions across each quadrant on multiple representative classification tasks. Finally, we extend our analyses to generation tasks, showing that our coordinate system can also be used to interpret ICL for generation tasks effectively. [1]

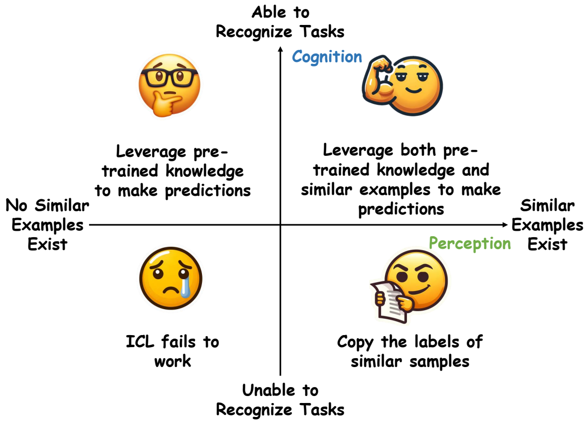


Figure 1: An overview of the proposed two-dimensional coordinate system for ICL. The y-axis represents cognition, indicating the model's ability to recognize tasks during ICL, while the x-axis represents perception, reflecting whether similar examples are included in the demonstrations.

1 Introduction

Large language models (LLMs) have demonstrated impressive in-context learning (ICL) capabilities (Brown et al., 2020), i.e., when provided with few-shot examples, LLMs can effectively perform a broad range of tasks without requiring parameter updates (Zhao et al., 2021; Min et al., 2022a; Su et al., 2022; Wei et al., 2023a). The simplicity of this method, combined with its zero training cost and the versatility of applying a single model across various tasks, has made ICL a promising approach to fully leveraging the potential of LLMs.

However, the underlying working mechanism of ICL remains an open question (Dai et al., 2022; Akyürek et al., 2022; Olsson et al., 2022; Panwar et al., 2024). Existing works hold two conflicting views to explain ICL: The first view argues *LLMs explicitly learn from similar examples in the demonstrations during the inference stage* (Liu et al., 2021). Selecting demonstrations similar to the test sample and ensuring their label correctness, or increasing the number of demonstration shots can both improve the performance (Rubin et al., 2022; Ye et al., 2023; Levy et al., 2023; Bertsch et al., 2024; Liu et al., 2024). The second view, on the contrary, suggests that *LLMs implicitly learn tasks required for downstream applications during the pre-training stage, and in-context demonstrations simply provide cues for them to recognize the task* (Xie et al., 2022). There have been empirical supports for this hypothesis which show that the performance of ICL is insensitive to label correctness of demonstrations, or the number of demon-

[1] Our code is publicly available at: https://github.com/plclmezboss/2D-Coordinate-System-for-ICL.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12375–12400
November 12-16, 2024

stration shots (Min et al., 2022b; Chen et al., 2023; Zhang et al., 2024; Zhu et al., 2024).

Although both of the above views hold in specific cases, neither can fully explain the working mechanism of ICL from a holistic perspective. In this work, we seek a unified framework encompassing both views to systematically unveil ICL. To do so, we first introduce the peak inverse rank metric to accurately identify the task recognition capability of LLMs. Based on this metric, we observe that LLMs do not always recognize tasks during ICL, even when correct labels and similar examples are provided in the demonstrations. Conversely, successful task recognition also does not necessarily require the presence of similar examples and correct labels. Hence, we suggest that the effectiveness of ICL should be described through the interactions of these two orthogonal variables, resulting in four distinct ICL scenarios. To conceptualize this intuitively, we represent each variable along two axes: The x-axis denotes perception, indicating the model's dependence on similar examples for reference. This mirrors human perception, where recognizing similar observed patterns and drawing analogies is crucial for interpreting new information. The y-axis represents cognition, reflecting the model's task recognition ability. Similar to human cognition, this involves recognizing the right logic learnt from past tasks and reasoning over it to draw the answer, rather than simply replicating observed patterns. Consequently, all ICL scenarios are visualized within a **two-dimensional coordinate system**, as depicted in Figure 1. In each quadrant of the coordinate system, we systematically analyze 8 models spanning up to 40B parameters across three major LLM families. We examine a wide range of classification tasks and have the following main findings:

- In the first quadrant, models are able to recognize the task and similar examples are included in the demonstrations. In this situation, *models can not only leverage their pre-trained knowledge to make predictions but also refer to the labels from similar examples if their pre-trained knowledge is insufficient*. However, if the labels of similar examples are incorrect, smaller models tend to replicate these incorrect labels, while larger models tend to rely on their pre-trained knowledge for making predictions.
- In the second quadrant, models are able to recognize the task but similar examples are not included in the demonstrations. In this situation, *models primarily leverage their pre-trained knowledge to make predictions*. Moreover, given that each input-label pair plays an identical role in helping models recognize the task, increasing the number of demonstration shots does not significantly enhance the effectiveness of ICL.
- In the third quadrant, models cannot recognize the task and similar examples are also not included in the demonstrations. In this situation, *ICL fails to work*. Models fail to properly leverage the demonstrations and tend to blindly predict the label of the first example.
- In the fourth quadrant, models cannot recognize the task but similar examples are included in the demonstrations. In this situation, *models directly replicate the labels of similar examples*. Therefore, the performance of ICL depends entirely on whether the labels of similar examples match the ground truth labels of test samples. Additionally, larger models are better at recognizing similar examples, which increases their tendency to copy the labels from those examples.

In general, our findings indicate that for *similar* examples, their label correctness has a *consistent and significant* impact on ICL, especially in scenarios where the model cannot recognize the task and relies heavily on similar examples for inference (below the x-axis). Increasing the number of demonstration shots substantially improves ICL, as it raises the likelihood of matching similar examples. For *dissimilar* examples, label correctness primarily affects the model's *confidence in task recognition* (above the x-axis), but once the task is properly recognized, the effect becomes marginal.

Finally, considering the wide application of ICL in generation tasks (Agrawal et al., 2022; Sia and Duh, 2023; Garcia et al., 2023), we extend our analyses beyond classification tasks by conducting a thorough case study on a machine translation task. This study demonstrates that our coordinate system can also effectively capture the behavior of ICL in generation tasks. In summary, our proposed coordinate system provides a principled and universal way to understand the working mechanism of ICL.

2 A 2D Coordinate System for ICL

2.1 The Coordinate Axes

Example Similarity. Recent works have shown that including similar examples to the test sample in the ICL demonstrations can lead to improved performance outcomes (Liu et al., 2021; Rubin et al., 2022; Ye et al., 2023; Levy et al., 2023; Bertsch et al., 2024). Inspired by this finding, we believe that the presence of examples similar to the test sample in the demonstrations is an important variable for distinguishing different ICL scenarios.

Given the broad concept of similarity, we seek to explore whether the similarity in ICL leans more towards semantic similarity or lexical similarity. To investigate this question, we perform comprehensive experiments in Appendix A, where we construct three different types of examples, including semantically similar but lexically dissimilar examples, lexically similar but semantically opposite examples, and randomly selected examples. We assign different semantically unrelated label words to these three elements and observe which semantically unrelated word the model predicts. We find that although ICL tends to slightly favor semantically similar examples over lexically similar ones, the preference for both is significantly greater than randomly selected examples. Thus, regardless of whether the similarity is lexical or semantic, as long as demonstrations contain examples of either type, we consider them to contain similar examples.

Task Recognition. Instead of capitalizing on similar examples, some previous research has demonstrated that in-context demonstrations simply provide information for the model to identify the task to deal with, after which prior knowledge obtained from pretraining data is leveraged to make predictions (Xie et al., 2022; Min et al., 2022b). This indicates that task recognition is also a crucial factor for ICL. However, whether models can always recognize tasks when performing ICL remains an open question. Hence, there is an urgent need for a method to quantitatively determine the model's task recognition ability.

A recent work by Wang et al. (2023) reveals that label words act as semantic anchors, accumulating information of corresponding demonstrations in the shallow layers. The information associated with these anchors is then aggregated in the deeper layers to form the model's final predictions. Inspired by this, we find that examining whether the hidden states of label tokens at internal layers possess task semantics can serve as an indicator of if ICL has recognized the task. We use a technique called the logit lens (nostalgebraist, 2020; Geva et al., 2021; Dar et al., 2023), which projects transformer representations into the vocabulary space, thereby enabling us to interpret abstruse representations in a human-interpretable manner. Specifically, we project the hidden states of each layer corresponding to the label tokens into the vocabulary space by multiplying them with the pre-trained language modeling head E, thus decoding the hidden states of each layer. After obtaining the vocabulary distribution of the hidden states for each layer, we calculate the inverse of the rank of the *task-representative* token within the vocabulary distribution for each layer. We use the peak inverse rank (***PIR***) across all layers as our metric to determine whether ICL has recognized the task. For clarity, we provide the formal definition of ***PIR*** in Appendix K. A high ***PIR*** indicates that ICL has successfully recognized the task, while a low ***PIR*** suggests a lack of task understanding capabilities.

To illustrate this, we consider an ICL scenario for the World Capital task: *"Word: France Label: Paris Word: Germany Label: Berlin Word: Italy Label:"*. We select the last label word *"Berlin"* to report the ***PIR*** of the task-representative token *"capital"* in Figure 2, where the ***PIR*** reaches 1 at layer 17. For the same task, we replace all labels with semantically irrelevant words to prevent ICL from recognizing the task. This setting, called **task learning**, was first introduced by Pan (2023). Concretely, we replace the first label *"Paris"* with *"bar"* and the second label *"Berlin"* with *"foo"*. We still select the last label *"foo"* to report the ***PIR*** of *"capital"* in Figure 2, where the ***PIR*** drops to 0.

Based on ***PIR***, we observe that models do not always recognize tasks during ICL, even when the demonstrations are entirely composed of correct input-label pairs (refer to Appendix C). Furthermore, we demonstrate that the presence of similar examples and the ability of task recognition are orthogonal to each other (refer to Appendix D). Given this, we take whether models can recognize tasks during the execution of ICL as the second variable to distinguish different ICL scenarios. We will use ***PIR*** as the criterion to select datasets for ICL that can and cannot recognize tasks.

2.2 ICL Scenario Exploration

Following the above discussions, ICL scenarios can be described comprehensively using two variables: task recognition and example similarity. These two variables result in four combinations, which can be visualized in a two-dimensional coordinate system (see Figure 1). In the positive half of the y-axis,

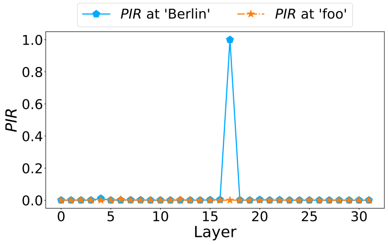


Figure 2: The ***PIR*** of *"capital"* at the last label token using Llama-2-7B, before and after replacing labels.

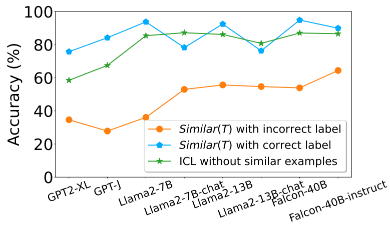


Figure 3: The average ICL accuracy for `Similiar(T)` with correct and incorrect labels, and ICL without similar examples.

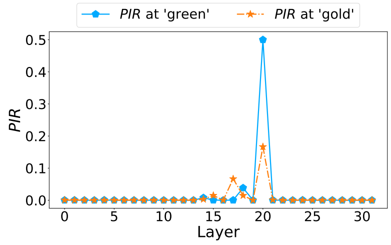


Figure 4: The ***PIR*** of *"color"* at the label token of `Similiar(T)`, when the label of `Similiar(T)` is correct and incorrect.

the model is capable of recognizing tasks, while in the negative half, it is not. Similarly, the positive half of the x-axis signifies the presence of similar examples, whereas their absence is indicated in the negative half.

Importantly, we establish this two-dimensional coordinate system not only to conceptualize the four possible combinations of ICL scenarios but also to consider the two variables on a continuous scale. Specifically, as the model's confidence in recognizing tasks increases, denoted by ***PIR*** approaching one, the y-coordinate value rises. Likewise, as the similarity between the provided examples and the test sample increases, the x-coordinate value rises. In the following section, we will provide a detailed description of how ICL works within each quadrant of the coordinate system.

3 Experiments and Results

3.1 Experimental Settings

Given that it is not intuitively clear whether models can recognize the task in a particular dataset, which in fact must be verified using the metric ***PIR***, we directly enumerate the classification datasets in which models can and cannot recognize the task. For detailed proofs, please refer to Appendix C.

Datasets in Which Models Can Recognize Tasks. These datasets are used for the upper part of the x-axis. We employ the Stanford Sentiment Treebank Binary (**SST-2**) (Socher et al., 2013) for sentiment analysis. In addition, we create two datasets for the **World Capitals** and **Reasoning about Colored Objects** tasks, which contain 50 hand-crafted pairs of *country-capital* and *object-color*, respectively. The detailed data are provided in Appendix F.

Datasets in Which Models Cannot Recognize Tasks. These datasets are used for the lower part of the x-axis. We utilize the Text REtrieval Conference (**TREC**) Question Classification dataset (Li and Roth, 2002; Hovy et al., 2001) for question type classification and the EmoContext (**emo**) (Chatterjee et al., 2019) for emotion classification.

Models. We adopt a comprehensive suite of models, including GPT2-XL (1.61B) (Radford et al., 2019) and GPT-J (6B) (Wang, 2021) from the GPT series; Llama-2-7B, Llama-2-13B, and their instruction-tuned counterparts from the Llama series (Touvron et al., 2023); and Falcon-40B, along with its instruction-tuned variant from the Falcon series (Almazrouei et al., 2023).

Prompt Format. We use neutral delimiters to avoid providing task-specific information. See Appendix E for details.

Accuracy Metric. We consider an individual prediction correct only when the token with the highest logit within the *entire* vocabulary matches the label of the test sample. Accuracy is the proportion of correct predictions across the whole dataset.

3.2 First Quadrant

In this quadrant, **models can recognize tasks when performing ICL, and the demonstrations contain examples similar to the test sample.** In this situation, since models can rely not only on their pre-trained knowledge after task recognition but also on the labels of similar examples to make predictions, we aim to determine how these two factors work together.

Implementation Details. Here, we consider an extreme setup where we artificially add the test sample into the demonstration, identifying it as a similar example with the highest similarity, denoted as `Similiar(T)`. We consider three settings: `Similiar(T)` with an incorrect label, `Similiar(T)` with the correct label, and ICL without similar examples. For the first setting, we select a label from the label space that differs from the test sample's label and assign it to the `Similiar(T)`. For the

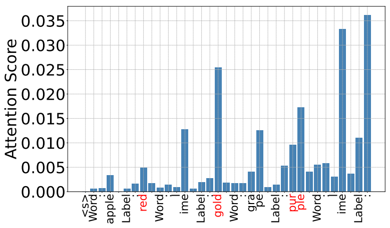


Figure 5: In the 20th layer, the attention scores of the last token ":" for all tokens. All label tokens are marked in red.

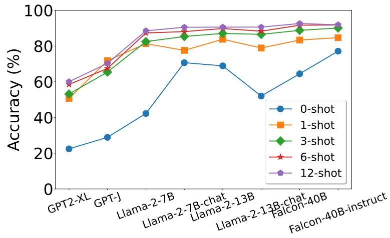


Figure 6: Averaged accuracy of zero-shot and ICL with varying numbers of demonstrations in the second quadrant.

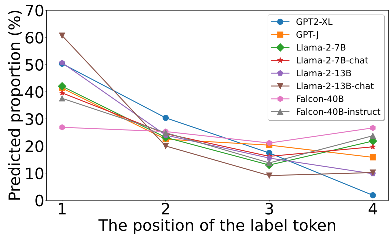


Figure 7: For the emo dataset, the preference of different models for label tokens at different absolute positions.

second setting, we select the label from the label space that matches the test sample's label and assign it to the `Similiar(T)`. For the third setting, we select examples from the training dataset with no similarity to the test sample to serve as demonstrations. We use $k = 6$ in-context examples. The results reflect averages from five random seeds and all datasets in which models can recognize tasks.

Experimental Results. Figure 3 shows that (1) adding `Similiar(T)` with the correct label in the demonstrations generally improves performance compared to ICL without similar examples (i.e., ICL falling within the second quadrant), although there is a slight performance decline for Llama-2-7B-chat and Llama-2-13B-chat. This indicates that models can not only utilize their pre-trained knowledge but also refer to the correct label of `Similiar(T)` when their pre-trained knowledge is insufficient. (2) However, including `Similiar(T)` with an incorrect label in the demonstrations induces a state of "confusion", making models neither completely rely on their pre-trained knowledge for predictions (performing worse than ICL without similar examples) nor completely overwrite their predictions with the label of `Similiar(T)`.

In-Depth Analysis. To comprehend the causes of "confusion" experienced by models, we present insights through a case study on the Reasoning about Colored Objects task using Llama-2-7B. We first consider the scenario when the label of `Similiar(T)` is correct: *"Word: apple Label: red Word: lime Label: green Word: grape Label: purple Word: lime Label:"*. We report the ***PIR*** of the task-representative token *"color"* at the label *"green"* of *"lime"*, as shown in Figure 4. Next, we substitute the label token *"green"* with the incorrect color *"gold"*, thereby constructing a `Similiar(T)` with an incorrect label in the demonstrations. Similarly, we report the ***PIR*** of the task-representative token *"color"* at the label *"gold"* of *"lime"*, also depicted in Figure 4. As illustrated, replacing the correct label with an incorrect one decreases the model's confidence in the Reasoning about Colored Objects task at the label token of `Similiar(T)`. Additionally, due to `Similiar(T)` having the highest semantic and lexical similarity, the last token of the input at the intermediate layer assigns the highest attention score to *"gold"* among all label tokens. For instance, consider the 20th layer, as shown in Figure 5. Consequently, at this layer, the hidden state for *"gold"* contributes more significantly to the residual stream of the last token compared to the hidden states of other label tokens. This leads to a reduction of task semantics and an increase of the semantics associated with the word *"gold"* within the hidden states of the last token. As a result, the model faces uncertainty about whether to rely on pre-trained knowledge for making predictions or to directly output the *"gold"* token, leading to the "confusion" phenomenon.

Additionally, as observed in Figure 3, when encountering the "confusion" phenomenon, models with smaller parameter sizes tend to output incorrect labels, whereas models with larger parameter sizes are more likely to rely on their pre-trained knowledge for the output. This indicates that when the label of `Similiar(T)` is incorrect, the confidence in the task at that label token increases as the model size increases.

Conclusion

In the first quadrant, models can leverage their pre-trained knowledge to make predictions once they recognize the task and can also refer to the labels from similar examples if their pre-trained knowledge is insufficient. However, if the labels of similar examples are incorrect, smaller models tend to replicate these incorrect labels, while larger models tend to rely on their pre-trained knowledge for making predictions.

3.3 Second Quadrant

In this quadrant, **models can recognize tasks when performing ICL, but the demonstrations do not contain examples similar to the test sample.** Due to the absence of similar examples, the phenomenon described in the first quadrant—where the incorrect label semantics of similar examples significantly affect the hidden states of the last token—does not occur. Therefore, randomly replacing labels in this quadrant does not substantially impact ICL performance, which is consistent with existing work (Min et al., 2022b). The detailed experimental results can be found in Appendix G.

Here, we would like to discuss a new question: *Since the label token of each example provides the same task semantics (because all examples are dissimilar to the test sample), does this imply that we can achieve good ICL performance with only a very small number of examples?*

Implementation Details. We use the "zero-shot" approach as our baseline, where the model is only given an instruction that specifies the task. Detailed instructions for each dataset can be found in Appendix I. Subsequently, we remove the instruction and provide the model with only demonstrations, then incrementally increase the number of shots. The results are averaged over five random seeds and datasets in which models can recognize tasks.

Experimental Results. The experimental results are shown in Figure 6. It can be observed that with only a single input-label pair, the performance of ICL significantly surpasses that of the zero-shot setting. Further increasing the number of demonstration shots results in very limited performance improvement. These results confirm our hypothesis: The roles of each label token overlap, and adding more examples merely reinforces the model's confidence in correctly identifying the task.

Conclusion

In the second quadrant, models primarily leverage their pre-trained knowledge to make predictions. Moreover, given that each input-label pair plays an identical role in helping models recognize tasks, increasing the number of in-context examples does not significantly enhance the effectiveness of ICL.

3.4 Third Quadrant

In this quadrant, **models cannot recognize tasks when performing ICL, and the demonstrations also do not contain examples similar to the test sample.** This represents the worst-case scenario among all quadrants. As illustrated in Figure 9, the performance of one-shot ICL in this quadrant is significantly worse than the zero-shot setting for all models. In this quadrant, what do the models rely on to make predictions when performing ICL?

Implementation Details. We consider an ICL setting where each example's label corresponds to a different label class from the dataset, covering all labels of the dataset (i.e., 4-shot ICL for emo and 6-shot ICL for TREC), to observe the predictive behavior of ICL. To ensure that the ICL output remains within the label space, we prefix the input with instructions to limit the output range without specifying the task. We select an equal number of samples from each label class in the dataset to serve as the set of test samples. For each test sample, we manually select demonstration examples that have virtually no semantic similarity or lexical similarity. Instead of focusing on whether the output matches the ground truth, we focus on the absolute position, specifically which position's label token will be predicted. Here we report the experimental results on the emo dataset in Figure 7. The results for TREC can be found in Appendix H.

Experimental Results. As shown in Figure 7, we find that models exhibit a strong positional bias in this quadrant. Specifically, we observe that for almost all models, there is a significantly high proportion of instances for which the label of the first input-label pair is predicted. In contrast, the proportion of predictions for the labels of other pairs is notably lower. We attribute this to the attention sink phenomenon discovered by Xiao et al. (2024), where models tend to allocate more attention to the initial tokens during prediction.

Conclusion

In the third quadrant, ICL fails to work. Specifically, models fail to leverage the ICL content for making predictions and tend to predict the label of the first example.

3.5 Fourth Quadrant

In this quadrant, **models cannot recognize tasks when performing ICL, but the demonstrations contain examples similar to the test sample.** Since models in this context can only reference similar examples, we hypothesize that the accuracy of ICL predictions hinges on whether the labels of these similar examples align with the ground-truth label of the test sample.

Implementation Details. Similar to the experimental setup in the first quadrant, during each ICL inference, we randomly select a label from the dataset that differs from the test sample's label and assign it to `Similiar(T)`. Since we aim to verify whether models in this quadrant rely on the labels of similar examples to make predictions, we use the proportion of predictions for the incorrect label assigned to `Similiar(T)` as our accuracy metric. We adopt $k = 12$ in-context examples. The results reflect averages from five random seeds and all datasets in which models cannot recognize tasks.

Experimental Results. As illustrated in Figure 8, the proportion of the model's predictions being the same as the incorrect label of `Similiar(T)` is high. Notably, as the model size increases, this proportion reaches an exceptionally high level, indicating that the model almost entirely relies on the label of `Similiar(T)` for its predictions. We posit that this phenomenon arises because, as the model size increases, its ability to discern the similarity of examples improves, thereby directing more attention to similar examples.

Conclusion

In the fourth quadrant, models directly replicate the labels of similar examples. Therefore, the performance of ICL depends heavily on whether the labels of similar examples match the ground truth labels of test samples. Additionally, larger models are better at recognizing similar examples, which increases their tendency to copy the labels from these examples.

4 Effects of Label Correctness and Demonstration Shot Number

As previous research presents two conflicting views about the effects of label correctness and shot number to ICL, in this section, we provide a brief summary about how our proposed coordinate system can explain this conflict in a more principled way.

Label Correctness. As long as similar examples are present in the demonstrations, the correctness of their labels consistently plays a crucial role in determining ICL performance, as discussed in Section 3.2 and Section 3.5. For dissimilar examples, when the ICL scenario is positioned above the x-axis, label correctness can impact the models' confidence in task recognition. Conversely, when positioned below the x-axis, label correctness does not decisively influence the models' predictions. In the third quadrant, models predominantly predict the label of the first example, whereas in the fourth quadrant, models are more likely to replicate the labels of similar examples.

Demonstration Shot Number. Above the x-axis, increasing the shot number does not significantly affect ICL performance, as the models primarily rely on their pre-trained knowledge to make predictions once they recognize the task. Additional shots merely reinforce the models' confidence in correctly identifying the task. However, below the x-axis, increasing the shot number significantly impacts ICL performance. The more shots there are, the higher the likelihood that the models will find more similar examples to refer to. This approach can transition ICL from the third quadrant to the fourth quadrant and enhance the likelihood of including more similar examples for reference within the fourth quadrant.

5 How to Make ICL Work Effectively?

From our proposed ICL coordinate system, it is evident that the effectiveness of ICL improves as the values of the x and y coordinates increase, moving towards the upper right quadrant. Conversely, as the values of the x and y coordinates decrease, moving towards the lower left quadrant, the effectiveness of ICL diminishes. These observations provide valuable insights into the enhancement of ICL performance. Specifically, improvements can be made by: **(1) strengthening the confidence in task recognition**, and **(2) providing examples with higher similarity in the demonstrations.** We select the most challenging third quadrant to demonstrate how ICL can be made effective through these two directions.

Including a task description instruction before ICL examples can facilitate task recognition.

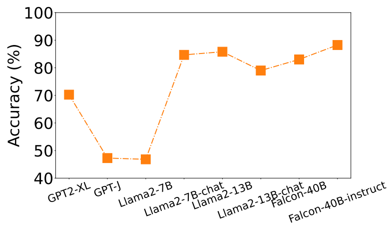


Figure 8: The proportion of predictions for the incorrect label corresponding to `Similiar(T)` on various models.

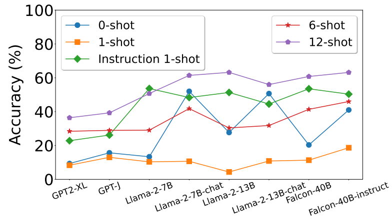


Figure 9: The average accuracy of (1,6,12)-shot ICL w/o instructions and (0,1)-shot ICL with instructions.

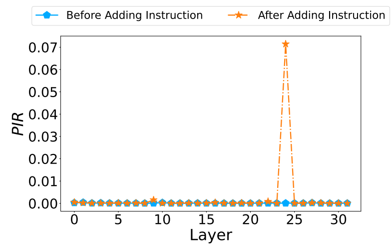


Figure 10: The ***PIR*** of *"question"* at the label token *"Human"* on Llama-2-7B-chat, before and after adding instruction.

As illustrated in Figure 10, we conduct a one-shot ICL case study on the TREC dataset using Llama-2-7B-chat (for specific instructions and details on the one-shot ICL, refer to Appendix J). Without the task description instruction, the ***PIR*** of the task-representative token *"question"* at the label is nearly zero. However, after adding the task description instruction, the ***PIR*** increases to 0.083 (ranking 12th in the vocabulary distribution), suggesting that the model recognizes the task to a certain extent. This demonstrates that instructions can promote task recognition. This finding provides methodological support for the implementation of the first direction. As shown in Figure 9, the effectiveness of instruction one-shot ICL is significantly better than that of one-shot ICL without instructions.

Both retrieval and long-context ICL can provide examples with higher similarity. For the second direction, a commonly utilized method in recent research is to retrieve a highly similar subset of examples to serve as demonstrations for each test set example, which has been shown to be effective (Liu et al., 2021; Rubin et al., 2022). Additionally, as the context lengths of LLMs continue to increase, another method to achieve this goal is to continuously increase the number of input-label pairs in the demonstrations. The more input-label pairs included, the higher the likelihood that the model will find more similar examples to refer to during ICL. As shown in Figure 9, the performance of one-shot ICL is consistently worse than zero-shot for all models. However, each time we increase the number of shots, the performance of ICL improves significantly. This aligns with the conclusions of Bertsch et al. (2024), who find that long-context ICL can be surprisingly effective, with most of the improvement stemming from attending to similar examples rather than task recognition.

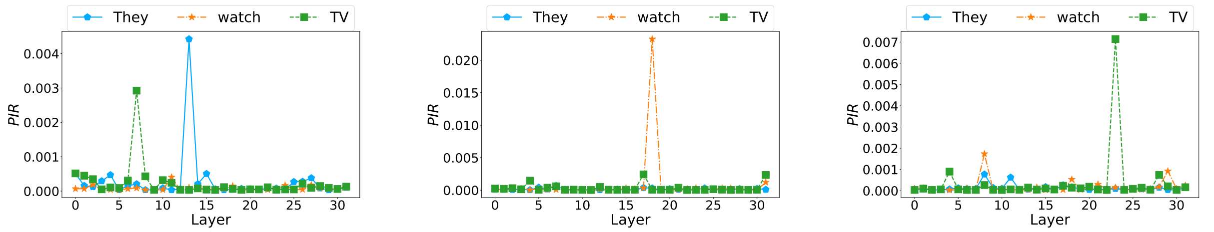
6 Extension to Generation Tasks

Given the recent success of ICL in generation tasks (Agrawal et al., 2022; Sia and Duh, 2023; Garcia et al., 2023), we aim for our two-dimensional coordinate system to enhance the understanding of ICL behavior not only in classification tasks but also in generation tasks. This is non-trivial, as almost no prior work has conducted an in-depth analysis of in-context generation tasks.

To extend our coordinate system to generation tasks, we face two main challenges: (1) In classification tasks, predicting a single label token is sufficient, whereas generation tasks require predicting an entire sentence. (2) We determine whether ICL recognizes the task by examining if the hidden states of label tokens at internal layers possess task semantics. However, for generation tasks, the label for each example in the demonstrations is not a single word but a complete sentence. To address these challenges, we formulate a hypothesis, H:

We treat a generation task as multiple smaller sub-classification tasks focused on predicting each token. In each sub-classification task, every label sentence contains a token that serves as the label token. However, the position of the label token in each label sentence changes when predicting different tokens.

According to this hypothesis, each ICL sub-classification task in generation tasks is equivalent to the standard ICL in classification tasks we previously studied, as both involve predicting a single token, with the label token still represented by a single token. However, verifying our hypothesis H for most generative tasks is challenging. In classification tasks, the task of predicting each token (i.e., a label token) is very clear and specific. For instance, in SST-2, it is a *"sentiment"* task. In contrast, for generative tasks, predicting a token may involve mimicking the abstract style of a segment



(a) ***PIR*** of *"subject"* on each token of the final label when predicting *"He"*.

(b) ***PIR*** of *"verb"* on each token of the final label when predicting *"reads"*.

(c) ***PIR*** of *"object"* on each token of the final label when predicting *"book"*.

Figure 11: For the in-context translation task with a strict subject-verb-object structure: "Sentence: 她喝水。 Label: She drinks water. Sentence: 我们吃米饭。 Label: We eat rice. Sentence: 他们看电视。 Label: They watch TV. Sentence: 他读书。 Label:" , the ***PIR*** of *"subject," "verb," "object"* at each token of the final label sentence when respectively predicting *"He," "reads," "book,"* on Llama-2-7B.

of an input-label pair, or in more extreme cases, predicting just a complete article. Explicitly describing all the tasks that are involved in predicting these tokens is very difficult.

For this reason, we consider an in-context translation task, with all examples in the demonstrations adhering to a strict *subject-verb-object* structure, as illustrated in Figure 11. If the ICL output also conforms to this strict subject-verb-object structure, it indicates that the tokens *"He"*, *"reads"*, and *"book"* are generated by performing the explicit sub-tasks of *"subject"*, *"verb"*, and *"object"*, respectively. We analyze the label sentence of the last example in the demonstration. Figure 11 illustrates that for the subject token *"He"*, the label token is *"They"*, with the other tokens rarely generating *"subject"* semantics. For the verb token *"reads"*, the label token is *"watch"*, with the other tokens seldom generating *"verb"* semantics. Similarly, for the object token *"book"*, the label token is *"TV"*, and the other tokens rarely generate *"object"* semantics. This phenomenon of label token sliding within the label sentence provides evidence supporting the plausibility of our hypothesis H.

Therefore, by decomposing the entire generative task into multiple sub-classification tasks, our coordinate system can help understand the working mechanisms of in-context generative tasks.

7 Related Work

Current research on ICL mechanisms mainly falls into the following categories:

Theoretical Framework. Recently, numerous studies have employed theoretical frameworks to enhance the understanding of ICL. Xie et al. (2022) describe ICL as implicit Bayesian inference. Garg et al. (2023) demonstrate that transformers can learn linear functions through ICL. Additionally, several studies conceptualize ICL as gradient descent on an implicit internal model (Akyürek et al., 2022; Von Oswald et al., 2023; Dai et al., 2022). Wei et al. (2023b) and Pan (2023) disentangle ICL into task recognition and task learning.

Empiricism. Various factors affecting ICL have been studied, such as the order of examples (Lu et al., 2022), the choice of label words (Min et al., 2022b; Yoo et al., 2022), and the selection of demonstrations (Liu et al., 2021; Rubin et al., 2022). Effective demonstration strategies (Ye et al., 2023; Li et al., 2023) can notably boost ICL performance.

Logit Lens. Recently, some studies use the logit lens technique (nostalgebraist, 2020; Geva et al., 2021) to project complex feature representations into the vocabulary space to study the mechanisms of ICL. Merullo et al. (2023) discover three distinct stages of processing by decoding the next token prediction at each layer. Todd et al. (2024) find that a small number of attention heads transport a compact representation of the demonstrated task. Yu and Ananiadou (2024) provide insights into the mechanisms of ICL in the context of task learning.

8 Conclusion

In this paper, we map two variables *whether LLMs can recognize the task* and *the presence of similar examples in the demonstrations* onto the y-axis and x-axis of a 2D coordinate system, to visualize ICL scenarios. First, for classification tasks, we conduct a systematic study of the proposed coordinate system and describe in detail the working mechanisms of ICL in each quadrant. Then, we extend our analyses beyond classification tasks through a thorough case study on a machine translation task. Our proposed coordinate system offers a universal framework to better understand ICL.

Limitations

Our extensive studies, despite offering a principled and universal approach to understanding the working mechanism of ICL, have several limitations. First, our research primarily focused on conventional ICL paradigms, leaving other paradigms such as chain of thought prompting (CoT) (Wei et al., 2023a) unexplored. Second, for generative tasks, we conducted a case study solely on in-context machine translation tasks adhering to a strict subject-verb-object structure. Third, due to hardware constraints, our investigation was primarily limited to models with up to 40 billion parameters. Further research replicating our study could use larger models with our 2D coordinate system to uncover more interesting findings.

Acknowledgments

This research / project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. We thank Xingluan (AI Cloud computing service), EIT and IDT High Performance Computing Center for providing computational resources for this project. This work corresponds to Fanghua Ye, Jinlan Fu, and Xiaoyu Shen.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. In-context examples selection for machine translation. *Preprint*, arXiv:2212.02437.

AI@Meta. 2024. Llama 3 model card.

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*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Chidanand Apt'e, Fred Damerau, and Sholom M. Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*. To appear.

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

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.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023. How many demonstrations do you need for in-context learning? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11149–11159.

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

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. *Preprint*, arXiv:2209.02535.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. *Preprint*, arXiv:2302.01398.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2023. What can transformers learn in-context? a case study of simple function classes. *Preprint*, arXiv:2208.01066.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *Preprint*, arXiv:2012.14913.

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*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. *Preprint*, arXiv:2305.04320.

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

Hui Liu, Wenya Wang, Hao Sun, Chris Xing Tian, Chenqi Kong, Xin Dong, and Haoliang Li. 2024. Unraveling the mechanics of learning-based demonstration selection for in-context learning. *Preprint*, arXiv:2406.11890.

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*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *Preprint*, arXiv:2104.08786.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

nostalgebraist. 2020. Interpreting gpt: The logit lens.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jane Pan. 2023. *What in-context learning "learns" in-context: Disentangling task recognition and task learning*. Ph.D. thesis, Princeton University.

Madhur Panwar, Kabir Ahuja, and Navin Goyal. 2024. In-context learning through the bayesian prism. *Preprint*, arXiv:2306.04891.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Suzanna Sia and Kevin Duh. 2023. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. *Preprint*, arXiv:2305.03573.

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.

Hui Su, Xiao Zhou, Houjin Yu, Xiaoyu Shen, Yuwen Chen, Zilin Zhu, Yang Yu, and Jie Zhou. 2022. Welm: A well-read pre-trained language model for chinese. *arXiv preprint arXiv:2209.10372*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. *Preprint*, arXiv:2310.15213.

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

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.

Ben Wang. 2021. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. `https://github.com/kingoflolz/mesh-transformer-jax`.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023a. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. *Preprint*, arXiv:2302.05698.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *Preprint*, arXiv:2205.12685.

Zeping Yu and Sophia Ananiadou. 2024. How do large language models learn in-context? query and key matrices of in-context heads are two towers for metric learning. *arXiv preprint arXiv:2402.02872*.

Miaoran Zhang, Vagrant Gautam, Mingyang Wang, Jesujoba O Alabi, Xiaoyu Shen, Dietrich Klakow, and Marius Mosbach. 2024. The impact of demonstrations on multilingual in-context learning: A multidimensional analysis. *arXiv preprint arXiv:2402.12976*.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

Dawei Zhu, Pinzhen Chen, Miaoran Zhang, Barry Haddow, Xiaoyu Shen, and Dietrich Klakow. 2024. Fine-tuning large language models to translate: Will a touch of noisy data in misaligned languages suffice? *arXiv preprint arXiv:2404.14122*.

A Lexical Similarity or Semantical Simlarity

In this section, we explore whether models give significantly more attention to semantically similar examples (which delve into deeper meanings) or to lexically similar examples (which may have opposite semantics but share superficial similarities). We also consider randomly selected dataset examples as a baseline. For this purpose, in the context of task learning (a setup first introduced by Pan (2023), which replaces all labels with semantically irrelevant words to prevent ICL from recognizing the task), we include three elements in the demonstrations, each representing a different type of similarity to the test sample, and observe which element's label the model predicts. The definitions of these three elements are as follows:

- **Example$_{\text{lexical}}$**: An example that largely overlaps lexically with the test sample, yet differs semantically and carries a distinct label.
- **Example$_{\text{semantic}}$**: An example that is semantically similar to the test sample, but has minimal lexical overlap and carries the same label. Specifically, we achieve this by paraphrasing the test sample.
- **Example$_{\text{baseline}}$**: A randomly selected example from the dataset, with minimal lexical and semantic similarity to the test sample.

Notably, we find that when each element appears only once in the demonstrations, as in a 3-shot ICL scenario, models tend to generalize the pattern of label changes. For instance, if the three labels are sequentially "a," "b," and "c," the models are likely to predict "d." To mitigate this phenomenon, we replicate each element multiple times (in our experiments, three times). Empirical experiments show that this approach can ensure the models' predictions stay within the intended label space.

Models. We employ various models from the GPT series, including GPT2-Medium (355M) and GPT2-XL (1.61B) (Radford et al., 2019), as well as GPT-J (6B) (Wang, 2021). To investigate the impact of instruction-tuning on similarity preferences, we also employ Llama-2-7B (Touvron et al., 2023), Llama-3-8B (AI@Meta, 2024), and Mistral-7B-v0.1 (Jiang et al., 2023), along with their instruction-tuned versions. All checkpoints of these models are sourced from the transformers library (Wolf et al., 2019).

Datasets. We adopt the Stanford Sentiment Treebank Binary (SST-2) (Socher et al., 2013) for sentiment analysis, Text REtrieval Conference Question Classification (TREC) (Li and Roth, 2002; Hovy et al., 2001) for question type classification, EmoContext (emo) (Chatterjee et al., 2019) for emotion classification, and `hate_speech18` (de Gibert et al., 2018) for hate speech detection. For each dataset, we leverage GPT-4 (OpenAI et al., 2024) to generate 20 triplets, formatted as $(\texttt{Test Sample}, \texttt{Example}_{\texttt{semantic}}, \texttt{Example}_{\texttt{lexical}})$, that are tailored to the style of the dataset. After generation, these triplets undergo a manual selection process to ensure quality. In addition, for each triplet, we randomly select an example from the original dataset to serve as $\texttt{Example}_{\texttt{baseline}}$, thereby forming a complete demonstration. For detailed information on the triplets generated for each dataset, please refer to Appendix B.

Implementation Details. In our experiments, the prompts consist solely of demonstrations without incorporating instructions. We employ neutral delimiters, specifically "Sentence:" and "Label:", to clearly separate the components of the demonstrations. This approach ensures that the models do not receive any task-specific information that could be inferred from the delimiters. Although previous works in the task learning setting typically employ semantically irrelevant words as labels, such as "foo" and "bar," we choose to label the elements with the initial letters of "lexical," "semantic," and "baseline" for ease of distinction. Consequently, the labels used for $\texttt{Example}_{\texttt{lexical}}$, $\texttt{Example}_{\texttt{semantic}}$, and $\texttt{Example}_{\texttt{baseline}}$ are "l," "s," and "b," respectively. For each ICL prediction, we document the token that ranked the highest in the model's output distribution. Subsequently, across all predictions, we calculate the proportions of the tokens "l," "s," and "b" respectively. The experimental results are averaged over five random seeds and all datasets.

Results and Analysis. The experimental results, as shown in Figure 12, reveal that: (1) Models pay significantly more attention to examples with lexical similarity or semantic similarity compared to randomly selected examples with minimal lexical similarity and semantic similarity. (2) Across all models, attention to semantic similarity is higher than to lexical similarity. (3) As the model size increases, models are more likely to predict tokens outside the label space, indicating that larger mod-

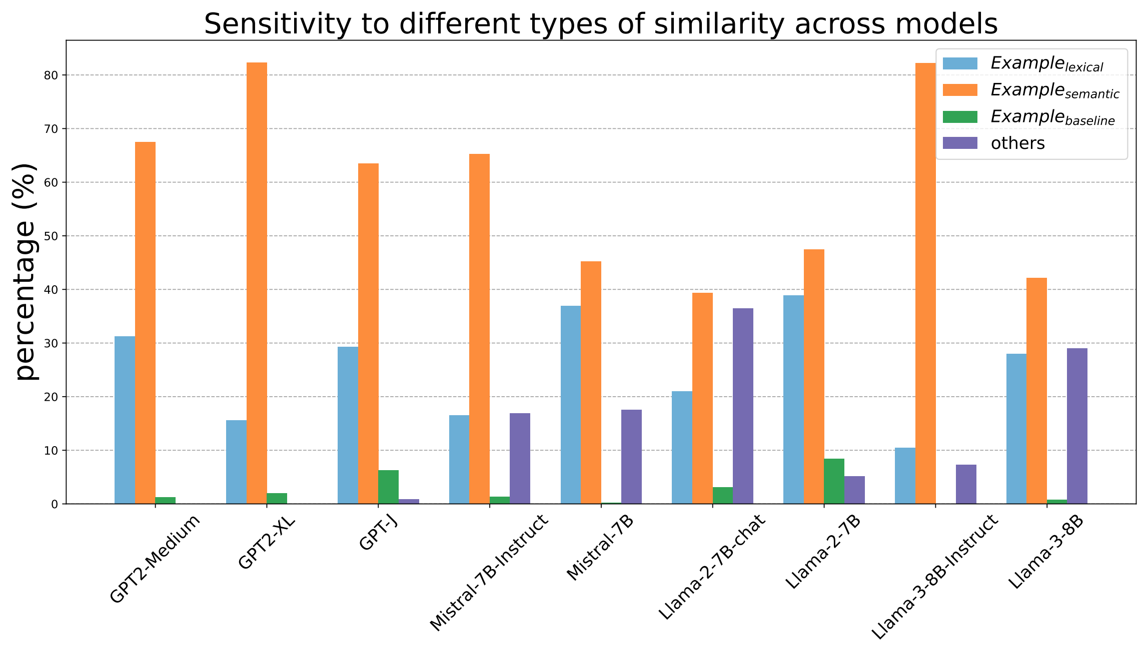


Figure 12: The proportion of each element's corresponding label in all predictions. Although we adopt the method of repeating each example in the demonstrations to mitigate the models' tendency to summarize patterns of label changes when performing ICL, for some models, it is still inevitable that tokens outside the label space are predicted. We use "others" to represent all tokens outside the label space.

els place greater weight on summarizing patterns of label changes in the context of task learning. (4) Instruction-tuned models tend to favor examples with semantic similarity, possibly due to the emphasis on semantic understanding during the instruction-tuning training process.

B GPT-4 Generated Triplets

Refer to Tables 1 and 2 for the 20 triplets generated by GPT-4 on the SST-2 dataset. Refer to Tables 3 and 4 for the 20 triplets generated by GPT-4 on the emo dataset. Refer to Tables 5 and 6 for the 20 triplets generated by GPT-4 on the TREC dataset. Refer to Tables 7 and 8 for the 20 triplets generated by GPT-4 on the `hate_speech18` dataset.

C Detailed Proofs of Models' Task Recognition on Classification Datasets

We employ one-shot ICL with correct input-label mapping to investigate the models' task recognition abilities across various datasets. Considering that the ***PIR*** metric involves the number of model layers and that different models have varying layer counts, we select the Llama-2-7B model as a representative for our analysis.

For the World Capitals task, we use the prompt *"Word: Germany Label: Berlin Word: Japan Label:"* to examine whether the label token *"Berlin"* triggers the task token *"capital."* For the Reasoning about Colored Objects task, we use the prompt *"Word: Apple Label: Red Word: Banana Label:"* to examine whether the label token *"Red"* triggers the task token *"color."* For the SST-2 dataset (Socher et al., 2013), we use the prompt *"Sentence: the part where nothing 's happening , Label: negative Sentence: a smile on your face Label:"*. It is noteworthy that ICL performed on the SST-2 dataset does not very conspicuously generate the task token *"sentiment."* We employ another label, *"positive,"* from the SST-2 dataset. If the demonstration includes only a *"negative"* label but the model is able to infer a *"positive"* meaning at the label token location, we still consider that ICL has successfully identified the type of task. For the TREC dataset (Li and Roth, 2002; Hovy et al., 2001), we use the prompt *"Sentence: Who killed Gandhi ? Label: Human Sentence: What is a fear of shadows ? Label:"* to examine whether the label token *"Human"* triggers the task token *"question."* For the emo dataset (Chatterjee et al., 2019), we use the prompt *"Sentence: talk you later sure d baby Label: others Sentence: really yep i"m i that bad Label:"* to examine whether the label token *"others"* triggers the task token *"emotion."*

All results are presented in Figure 13. It is evi-

Test sample	$\text{Example}_{\text{semantic}}$	$\text{Example}_{\text{lexical}}$
The soundtrack enriches the entire movie. **Label: Positive**	The music significantly enhances the film's appeal. **Label: Positive**	The soundtrack diminishes the entire movie. **Label: Negative**
The actor's performance is truly mesmerizing. **Label: Positive**	The actor's portrayal captivates the audience completely. **Label: Positive**	The actor's performance is truly forgettable. **Label: Negative**
The plot twists were unexpected and thrilling. **Label: Positive**	The story developments were surprising and exciting. **Label: Positive**	The plot twists were predictable and dull. **Label: Negative**
The cinematography is breathtaking and innovative. **Label: Positive**	The visual direction offers stunning and groundbreaking visuals. **Label: Positive**	The cinematography is uninspiring and outdated. **Label: Negative**
The dialogue was witty and delightful. **Label: Positive**	The conversation was sharp and enjoyable. **Label: Positive**	The dialogue was humorless and disappointing. **Label: Negative**
The direction is masterful and precise. **Label: Positive**	The film's guidance shows exceptional skill and accuracy. **Label: Positive**	The direction is clumsy and imprecise. **Label: Negative**
The special effects are spectacular and memorable. **Label: Positive**	The visual effects stand out as extraordinary and unforgettable. **Label: Positive**	The special effects are unimpressive and forgettable. **Label: Negative**
The pacing keeps you engaged from start to finish. **Label: Positive**	The rhythm maintains your attention throughout the entire movie. **Label: Positive**	The pacing loses your interest from start to finish. **Label: Negative**
The characters are richly developed and relatable. **Label: Positive**	The portrayal of characters is deeply crafted and connects well with the audience. **Label: Positive**	The characters are poorly developed and unrelatable. **Label: Negative**
The film's creativity is both refreshing and inspiring. **Label: Positive**	The movie's originality offers a new and motivational perspective. **Label: Positive**	The film's creativity is both stale and uninspiring. **Label: Negative**

Table 1: For the SST-2 dataset, 20 triplets generated by GPT-4 (Part 1).

dent that for the World Capitals, Reasoning about Colored Objects, and SST-2 datasets, the ***PIR*** is 1. This suggests that for these datasets, models can recognize the task during ICL execution. Conversely, for the TREC and emo datasets, the ***PIR*** is close to 0. This indicates that models fail to recognize the task during ICL for these datasets.

D Detailed Proof: The Orthogonality of Similar Examples Presence and LLMs' Task Recognition Ability

From Appendix C, we can observe that for the World Capitals, Reasoning about Colored Objects, and SST-2 datasets, models can recognize the task even without the presence of similar examples in the demonstrations. Consequently, to substantiate the claim that the presence of similar examples is orthogonal to task recognition, it is imperative to determine whether the provision of similar examples enhances the ability of models to recognize tasks within the TREC and emo datasets during ICL execution.

Implementation Details. For the TREC and emo datasets, we utilize the test samples as outlined in Appendix C. However, for the demonstrations,

Test sample	Example$_{\text{semantic}}$	Example$_{\text{lexical}}$
I despise this show for its lack of originality. **Label: Negative**	The series annoys me with its derivative content. **Label: Negative**	I adore this show despite its lack of originality. **Label: Positive**
The ending was predictable and boring. **Label: Negative**	The conclusion was foreseeable and tedious. **Label: Negative**	The ending was unpredictable and exciting. **Label: Positive**
Their service was slow and frustrating. **Label: Negative**	The customer service was sluggish and irritating. **Label: Negative**	Their service was quick and satisfying. **Label: Positive**
It's utterly pointless and dull. **Label: Negative**	Completely meaningless and uninteresting. **Label: Negative**	It's utterly purposeful and engaging. **Label: Positive**
The plot twists were contrived and unconvincing. **Label: Negative**	The storyline turns felt forced and unbelievable. **Label: Negative**	The plot twists were natural and convincing. **Label: Positive**
The movie was generally uninspiring. **Label: Negative**	The film rarely evoked any excitement. **Label: Negative**	The movie was generally inspiring. **Label: Positive**
The soundtrack is hardly noticeable. **Label: Negative**	You barely hear the music throughout. **Label: Negative**	The soundtrack is highly noticeable. **Label: Positive**
The pacing is slow and tedious. **Label: Negative**	The tempo drags and feels monotonous. **Label: Negative**	The pacing is quick and engaging. **Label: Positive**
The narrative lacks depth and coherence. **Label: Negative**	The story misses complexity and clarity. **Label: Negative**	The narrative has depth and coherence. **Label: Positive**
The performance was overly dramatic and false. **Label: Negative**	The acting was excessively theatrical and inauthentic. **Label: Negative**	The performance was subtly dramatic and genuine. **Label: Positive**

Table 2: For the SST-2 dataset, 20 triplets generated by GPT-4 (Part 2).

we substitute the original examples with correctly labeled test samples that have the highest semantic and lexical similarity. We subsequently investigate whether the label token can trigger task-representative tokens when similar examples are provided in these two datasets.

Experimental Results. The experimental results are shown in Figure 14. It can be observed that, for these two datasets, the ***PIR*** remains close to 0. This suggests that the model's ability to recognize the task is not influenced by the presence of similar examples but is rather determined by the intrinsic characteristics of the task itself.

E Delimiters Used for Each Dataset

For SST-2, TREC, and emo, we use *"Sentence:"* and *"Label:"*; for the World Capitals and Reasoning about Colored Objects tasks, we use *"Word:"* and *"Label:"* to clearly separate the components of the demonstrations.

F Detailed Data for World Capitals and Reasoning about Colored Objects Tasks

Detailed data for the World Capitals and Reasoning about Colored Objects tasks are shown in Table 9.

G Impact of Random Label Replacement in the Second Quadrant

Implementation Details. We consider two settings: correct input-label mapping and random input-label mapping. In the former, all input-label pair mappings in the demonstration are correct. In the latter, for each input-label pair, the label is randomly selected from the label space. We use $k = 6$ in-context examples without instructions. The results reflect averages from five random seeds and

Test sample	Example$_{\text{semantic}}$	Example$_{\text{lexical}}$
It seems like a regular day at the office. **Label: Others**	Just another normal workday. **Label: Others**	Today, the office feels unsettlingly quiet. **Label: Sad**
I need to go grocery shopping later. **Label: Others**	Later today, I have some grocery shopping to do. **Label: Others**	I'm frustrated about having to go grocery shopping later. **Label: Angry**
I'm so happy we're going on a vacation! **Label: Happy**	I'm thrilled about our upcoming vacation! **Label: Happy**	I'm stressed about all the packing needed for our vacation. **Label: Angry**
That birthday party was a blast! **Label: Happy**	I truly enjoyed the fun at that birthday party! **Label: Happy**	That birthday party was too loud and overwhelming for me. **Label: Sad**
Losing my pet has left me heartbroken. **Label: Sad**	I am deeply saddened by the loss of my pet. **Label: Sad**	Dealing with my pet's loss has made me irritable and upset. **Label: Angry**
It's so gloomy outside today, it makes me feel down. **Label: Sad**	The dreary weather today really dampens my spirits. **Label: Sad**	The gloomy weather outside is irritating. **Label: Angry**
I can't believe how unfair that decision was! **Label: Angry**	I'm really upset about that unjust decision! **Label: Angry**	That decision was so disappointing and unfair. **Label: Sad**
This constant noise is driving me crazy! **Label: Angry**	I'm getting furious over the incessant noise! **Label: Angry**	This constant noise is really getting on my nerves. **Label: Others**
I can't believe I got promoted at work! **Label: Happy**	I am so excited about my promotion at work! **Label: Happy**	I can't believe how stressed I am at work. **Label: Sad**
Missing the bus has ruined my day. **Label: Sad**	Missing the bus completely ruined my entire day. **Label: Sad**	Missing the bus has made me furious. **Label: Angry**

Table 3: For the emo dataset, 20 triplets generated by GPT-4 (Part 1).

all datasets in which models can recognize tasks.

Experimental Results. The experimental results are shown in Figure 15. It can be observed that randomly replacing labels does not significantly impact ICL performance. This is due to the following reasons: (1) Although some input-label pair mappings are incorrect, the effect is limited to weaker task semantics generated by these labels. (2) The absence of similar examples prevents the incorrect label semantics of similar examples from significantly affecting the hidden states of the last token.

H Positional Bias of the TREC Dataset

The results for TREC in the third quadrant are shown in Figure 16. It can be observed that, similar to the emo dataset, the models exhibit strong positional bias when performing ICL on the TREC dataset. Specifically, they tend to predict the label of the first example.

I Detailed Instructions for Each Dataset

Detailed instructions for each dataset when models perform zero-shot tasks are shown in Table 10.

J Specific Instructions and Details on One-Shot ICL for Section 5

For the ICL case study on the TREC dataset conducted in Section 5, the task description instruction is as follows:

> *The task involves categorizing questions into specific categories based on their*

Test sample	Example$_{semantic}$	Example$_{lexical}$
My friend surprised me with a gift. **Label: Happy**	I was delighted when my friend gave me a gift. **Label: Happy**	My friend surprised me with a rude comment. **Label: Angry**
The news of the accident left me in tears. **Label: Sad**	I cried when I heard about the accident. **Label: Sad**	The news of the accident left me feeling numb. **Label: Others**
I'm so angry that my computer crashed again! **Label: Angry**	It's infuriating that my computer crashed again! **Label: Angry**	I'm so sad that my computer crashed again. **Label: Sad**
It's a beautiful day outside, I feel great. **Label: Happy**	The nice weather outside makes me feel wonderful. **Label: Happy**	It's a beautiful day outside, but I feel anxious. **Label: Others**
The way they treated me was so disrespectful. **Label: Angry**	Their treatment of me was incredibly disrespectful. **Label: Angry**	The way they treated me was so kind. **Label: Happy**
I'm feeling down because I didn't get the job. **Label: Sad**	I'm feeling really sad because I didn't get hired. **Label: Sad**	I'm feeling great because I got the job. **Label: Happy**
This project has been so rewarding. **Label: Happy**	Working on this project has been incredibly fulfilling. **Label: Happy**	This project has been so frustrating. **Label: Angry**
I can't stand the traffic jam every morning. **Label: Angry**	The daily traffic jam every morning drives me nuts. **Label: Angry**	I love the peaceful mornings without traffic jams. **Label: Happy**
I'm ecstatic about the new opportunities ahead! **Label: Happy**	I'm thrilled about the upcoming new opportunities! **Label: Happy**	I'm anxious about the new challenges ahead. **Label: Sad**
I can't believe they forgot my birthday. **Label: Sad**	It's disappointing that they forgot my birthday. **Label: Sad**	I can't believe they remembered my birthday. **Label: Happy**

Table 4: For the emo dataset, 20 triplets generated by GPT-4 (Part 2).

content. Please classify each given question into one of the following broad class labels: Abbreviation, Entity, Description, Human, Location, or Number.

The specific content of the one-shot ICL is as follows:

Question: *Who killed Gandhi?*
Label: *Human*
Question: *What is a fear of shadows?*
Label:

K Formal Mathematical Definition of PIR

The ***PIR*** metric quantifies a model's ability to recognize tasks. For a given layer l corresponding to the label token, we begin by projecting the hidden state h_l into the vocabulary space by multiplying it with the pre-trained language modeling head E. The rank of the task-representative token within this projected distribution is then denoted as $rank_{task}(h_l, E)$. The ***PIR*** is formally defined as:

$$PIR = \max_{l} \frac{1}{rank_{task}(h_l, E)}. \tag{1}$$

L Additional Experiments on the Reuters-21578 Dataset

To further enhance the comprehensiveness of our study, we conduct additional experiments on the Reuters-21578 dataset (Apt'e et al., 1994), which comprises eight label classes. This dataset is validated through ***PIR*** as one in which models are unable to recognize tasks.

Experimental Results. We employ the same experimental setup as outlined in Section 3.4. The results presented in Table 11 corroborate our existing findings, highlighting the tendency of models to frequently predict the label of the initial example.

Test sample	Example$_{semantic}$	Example$_{lexical}$
What does ’CPU’ stand for? **Label: ABBR**	What is the meaning of ’CPU’? **Label: ABBR**	Where is the ’CPU’ located? **Label: LOC**
What is the capital of France? **Label: LOC**	Which city is the capital of France? **Label: LOC**	What is the population of France? **Label: NUM**
Who discovered electricity? **Label: HUM**	Name the person who discovered electricity. **Label: HUM**	When was electricity discovered? **Label: NUM**
What is the boiling point of water? **Label: NUM**	At what temperature does water boil? **Label: NUM**	Who discovered the boiling point of water? **Label: HUM**
Who is the CEO of Tesla? **Label: HUM**	Identify the current CEO of Tesla. **Label: HUM**	Where is the headquarters of Tesla? **Label: LOC**
What does ’HTTP’ mean? **Label: ABBR**	Explain the term ’HTTP’. **Label: ABBR**	Who invented ’HTTP’? **Label: HUM**
Name a famous painter from Spain. **Label: HUM**	Who is a renowned Spanish painter? **Label: HUM**	What is a famous painting from Spain? **Label: ENTY**
Describe the process of photosynthesis. **Label: DESC**	Explain how photosynthesis works. **Label: DESC**	Who discovered photosynthesis? **Label: HUM**
Where is the Great Wall of China located? **Label: LOC**	Locate the Great Wall of China. **Label: LOC**	When was the Great Wall of China built? **Label: NUM**
How many continents are there? **Label: NUM**	How many continents exist? **Label: NUM**	What is the largest continent? **Label: ENTY**

Table 5: For the TREC dataset, 20 triplets generated by GPT-4 (Part 1), where ENTY stands for Entity, HUM stands for Human being, NUM stands for Numeric value, LOC stands for Location, ABBR stands for Abbreviation, and DESC stands for Description and abstract concept.

Furthermore, we confirm the insights discussed in Section 4. Table 12 shows that while 1-shot ICL initially underperforms compared to instruction-based 0-shot ICL, it surpasses it as the shot count increases, and instruction-based 1-shot ICL proves more effective than 1-shot ICL without instructions. These results affirm the efficacy of the two proposed directions for improving ICL performance within the third quadrant.

Test sample	$\text{Example}_{\text{semantic}}$	$\text{Example}_{\text{lexical}}$
What is the full form of 'UNICEF'? **Label: ABBR**	What does the abbreviation 'UNICEF' represent? **Label: ABBR**	Who founded 'UNICEF'? **Label: HUM**
Who won the Nobel Peace Prize in 2020? **Label: HUM**	Who was awarded the Nobel Peace Prize in 2020? **Label: HUM**	What is the prize money for the Nobel Peace Prize? **Label: NUM**
What is the meaning of 'quantum physics'? **Label: DESC**	Define 'quantum physics'. **Label: DESC**	Who coined the term 'quantum physics'? **Label: HUM**
Name the author of 'Pride and Prejudice'. **Label: HUM**	Identify the writer of 'Pride and Prejudice'. **Label: HUM**	When was 'Pride and Prejudice' published? **Label: NUM**
What is the largest ocean on Earth? **Label: ENTY**	Which ocean is the largest on Earth? **Label: ENTY**	Where is the largest ocean located? **Label: LOC**
How tall is Mount Everest? **Label: NUM**	What is the height of Mount Everest? **Label: NUM**	Who measured the height of Mount Everest? **Label: HUM**
What does 'DNA' stand for? **Label: ABBR**	Explain what 'DNA' stands for. **Label: ABBR**	Where is 'DNA' found? **Label: LOC**
What type of animal is a tiger? **Label: ENTY**	What species does a tiger belong to? **Label: ENTY**	Where are tigers commonly found? **Label: LOC**
Explain the theory of relativity. **Label: DESC**	Describe the concept of relativity. **Label: DESC**	Who developed the theory of relativity? **Label: HUM**
Who was the first woman to fly solo across the Atlantic? **Label: HUM**	Who was the first female to fly solo across the Atlantic? **Label: HUM**	When did the first woman fly solo across the Atlantic? **Label: NUM**

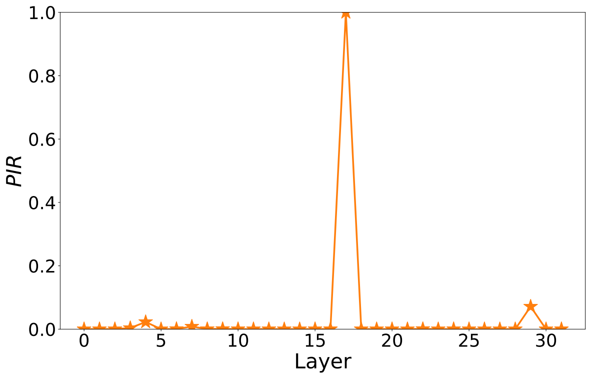
Table 6: For the TREC dataset, 20 triplets generated by GPT-4 (Part 2), where ENTY stands for Entity, HUM stands for Human being, NUM stands for Numeric value, LOC stands for Location, ABBR stands for Abbreviation, and DESC stands for Description and abstract concept.

Test sample	Example$_{semantic}$	Example$_{lexical}$
You are such an inspiration to everyone around you. **Label: noHate**	You inspire everyone around you. **Label: noHate**	You are such a burden to everyone around you. **Label: hate**
I hate it when people like you try to ruin everything. **Label: hate**	I can't stand it when people like you mess everything up. **Label: hate**	I love it when people like you try to make things better. **Label: noHate**
Your kindness and generosity know no bounds. **Label: noHate**	Your endless kindness and generosity are remarkable. **Label: noHate**	Your selfishness and greed know no bounds. **Label: hate**
People like you make this world a terrible place. **Label: hate**	People like you make this world worse. **Label: hate**	People like you make this world a wonderful place. **Label: noHate**
I'm so grateful for your support and friendship. **Label: noHate**	I'm deeply thankful for your support and friendship. **Label: noHate**	I'm so resentful of your support and friendship. **Label: hate**
You are nothing but a waste of space. **Label: hate**	You are completely useless. **Label: hate**	You are everything but a waste of space. **Label: noHate**
Your efforts are making a significant difference. **Label: noHate**	Your contributions are having a major impact. **Label: noHate**	Your efforts are making no difference at all. **Label: hate**
You don't belong here, go back to where you came from. **Label: hate**	You should leave and never come back. **Label: hate**	You belong here, stay where you are. **Label: noHate**
Thank you for always being there for me. **Label: noHate**	Thanks for always being supportive. **Label: noHate**	Thank you for never being there for me. **Label: hate**
You are the reason why nothing ever gets better. **Label: hate**	You are why things never improve. **Label: hate**	You are the reason why everything gets better. **Label: noHate**

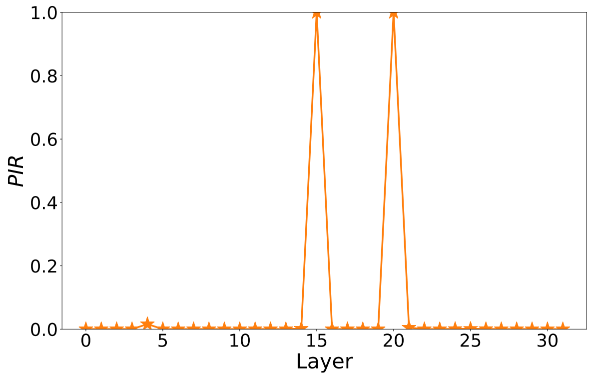
Table 7: For the `hate_speech18` dataset, 20 triplets generated by GPT-4 (Part 1).

Test sample	Example$_{\text{semantic}}$	Example$_{\text{lexical}}$
Your dedication to helping others is commendable. **Label: noHate**	Your commitment to helping others is admirable. **Label: noHate**	Your dedication to harming others is commendable. **Label: hate**
I can't stand people like you. **Label: hate**	People like you are unbearable. **Label: hate**	I can't stand people like you making a positive impact. **Label: noHate**
You have a heart of gold and it shows in everything you do. **Label: noHate**	Your golden heart is evident in all your actions. **Label: noHate**	You have a heart of stone and it shows in everything you do. **Label: hate**
Why don't you just disappear already? **Label: hate**	Why don't you just go away? **Label: hate**	Why don't you just stay and make things better? **Label: noHate**
I'm amazed by your talent and hard work. **Label: noHate**	Your talent and hard work are truly impressive. **Label: noHate**	I'm amazed by your lack of talent and hard work. **Label: hate**
You're nothing but a burden to everyone. **Label: hate**	You're a burden to everyone around you. **Label: hate**	You're nothing but a blessing to everyone. **Label: noHate**
Your positive attitude is contagious. **Label: noHate**	Your positive outlook is infectious. **Label: noHate**	Your negative attitude is contagious. **Label: hate**
You are a disgrace to your family. **Label: hate**	You are a shame to your family. **Label: hate**	You are an honor to your family. **Label: noHate**
I appreciate your thoughtful advice and guidance. **Label: noHate**	I value your wise advice and guidance. **Label: noHate**	I despise your thoughtful advice and guidance. **Label: hate**
You are the worst kind of person. **Label: hate**	You are the worst person I've ever met. **Label: hate**	You are the best kind of person. **Label: noHate**

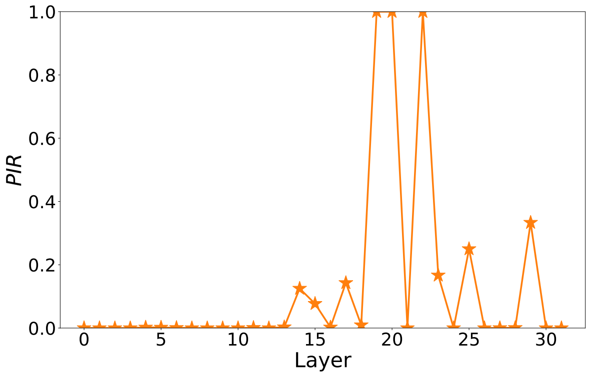
Table 8: For the `hate_speech18` dataset, 20 triplets generated by GPT-4 (Part 2).



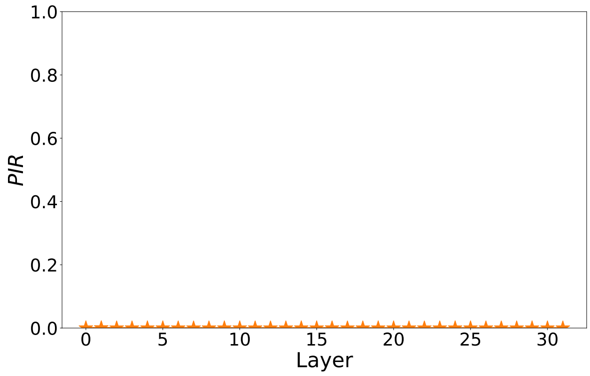
(a) The ***PIR*** of ***"capital"*** at the label token ***"Berlin"*** in Llama-2-7B for the Capital World task.



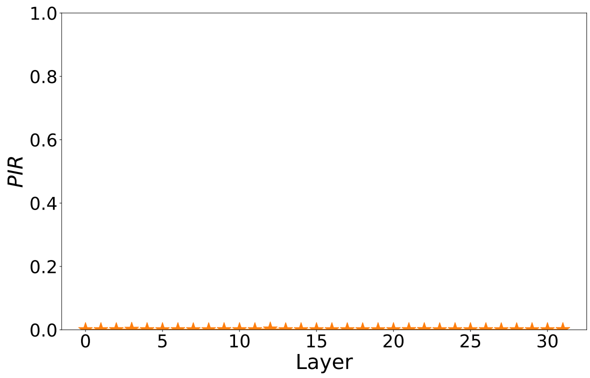
(b) The ***PIR*** of ***"color"*** at the label token ***"Red"*** in Llama-2-7B for the Reasoning about Colored Objects task.



(c) The ***PIR*** of ***"positive"*** at the label token ***"negative"*** in Llama-2-7B for the SST-2 dataset.

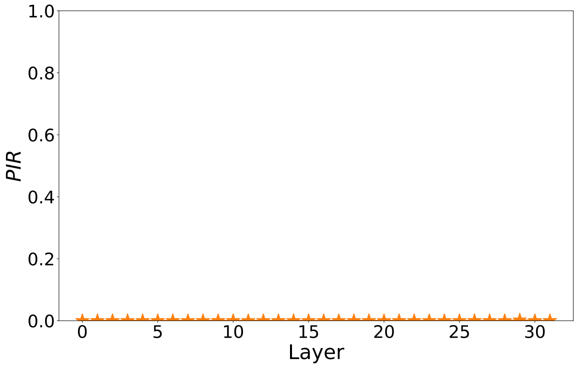


(d) The ***PIR*** of ***"question"*** at the label token ***"Human"*** in Llama-2-7B for the TREC dataset.

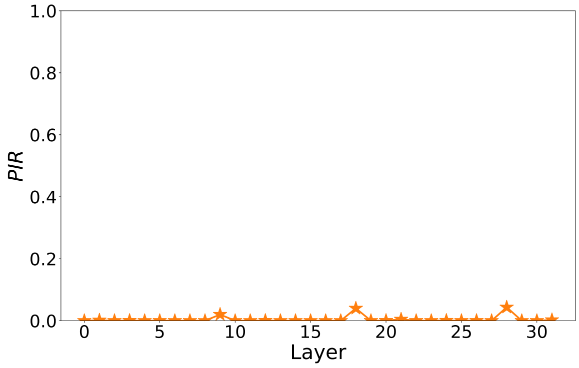


(e) The ***PIR*** of ***"emotion"*** at the label token ***"others"*** in Llama-2-7B for the emo dataset.

Figure 13: The ***PIR*** values across different datasets.



(a) When the correctly labeled test sample is included as a similar example in the demonstration, ***PIR*** of ***"question"*** at the label token ***"Description"*** for the TREC dataset.



(b) When the correctly labeled test sample is included as a similar example in the demonstration, ***PIR*** of ***"emotion"*** at the label token ***"sad"*** for the emo dataset.

Figure 14: When the correctly labeled test sample is included as a similar example in the demonstration, the ***PIR*** values across different datasets.

Tasks	**Detailed Data**
World Capital task	Canada-Ottawa, Australia-Canberra, Brazil-Brasília, China-Beijing, France-Paris, Germany-Berlin, India-New Delhi, Italy-Rome, Japan-Tokyo, Mexico-Mexico City, Russia-Moscow, South Africa-Pretoria, South Korea-Seoul, Spain-Madrid, Turkey-Ankara, United Kingdom-London, United States-Washington, D.C., Argentina-Buenos Aires, Egypt-Cairo, Nigeria-Abuja, Sweden-Stockholm, Norway-Oslo, Denmark-Copenhagen, Finland-Helsinki, Poland-Warsaw, Ukraine-Kyiv, Netherlands-Amsterdam, Belgium-Brussels, Austria-Vienna, Switzerland-Bern, Portugal-Lisbon, Greece-Athens, Hungary-Budapest, Czech Republic-Prague, Romania-Bucharest, Thailand-Bangkok, Vietnam-Hanoi, Malaysia-Kuala Lumpur, Singapore-Singapore, Indonesia-Jakarta, Saudi Arabia-Riyadh, Israel-Jerusalem, Chile-Santiago, Colombia-Bogotá, Peru-Lima, New Zealand-Wellington, Ireland-Dublin, Pakistan-Islamabad, Bangladesh-Dhaka, Philippines-Manila.
Reasoning about Colored Objects task	Apple-red, Banana-yellow, Cherry-red, Lemon-yellow, Sky-blue, Grass-green, Grape-purple, Orange-orange, Strawberry-red, Blueberry-blue, Cloud-white, Rose-red, Sunflower-yellow, Snow-white, Coal-black, Pumpkin-orange, Water-blue, Chocolate-brown, Gold-gold, Silver-silver, Carrot-orange, Lime-green, Eggplant-purple, Flamingo-pink, Ocean-blue, Forest-green, Cranberry-red, Peach-pink, Sunset-orange, Night-black, Butter-yellow, Olive-green, Sand-yellow, Violet-purple, Tangerine-orange, Cherry blossom-pink, Coral-orange, Ash-grey, Emerald-green, Sapphire-blue, Ruby-red, Cotton-white, Ivory-white, Charcoal-black, Peacock-blue, Jade-green, Amber-orange, Hazelnut-brown, Lavender-purple, Cinnamon-brown.

Table 9: Detailed data for World Capitals and Reasoning about Colored Objects tasks.

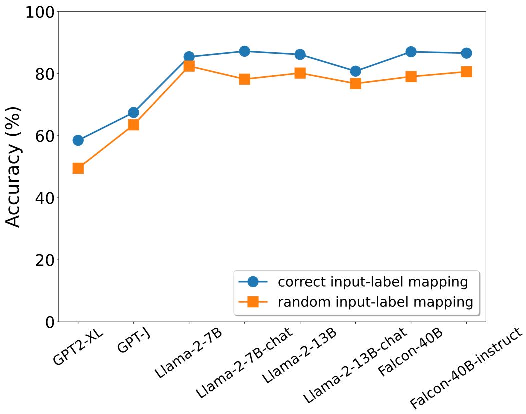


Figure 15: The impact of random label replacement in the second quadrant.

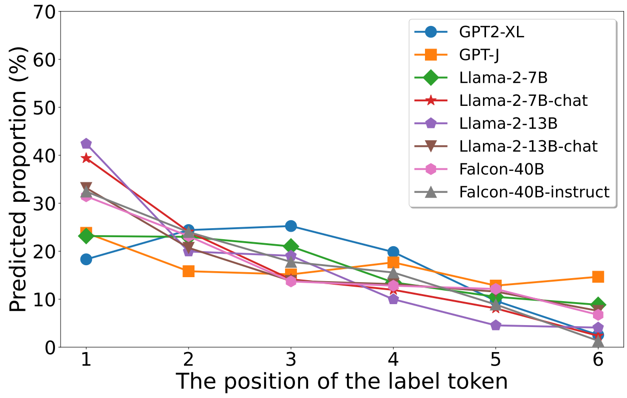


Figure 16: For the TREC dataset, the preference of different models for label tokens at different absolute positions.

Datasets	**Instructions for Zero-Shot Tasks**
World Capital	Please identify the capital city for the given country.
Reasoning about Colored Objects	Please identify the color of the given object.
SST-2	The task involves classifying sentences based on their expressed sentiment. Please classify each given sentence into one of the following sentiment labels: positive or negative.
TREC	The task involves categorizing questions into specific categories based on their content. Please classify each given question into one of the following broad class labels: Abbreviation, Entity, Description, Human, Location, or Number.
emo	Please classify the given utterance into one of the following emotion classes: happy, sad, angry, or others.

Table 10: Detailed instructions for each dataset in the zero-shot setting.

Table 11: Preference (%) of different models for label tokens at different absolute positions on the Reuters-21578.

Models	First Label	Second Label	Third Label	Fourth Label	Fifth Label	Sixth Label	Seventh Label	Eighth Label
GPT2-XL	33.1	16.55	15.86	13.79	10.34	10.34	0	0
GPT-J	34.67	13.33	12	6.67	10	23.33	0	0
Llama2-7B	45.65	23.91	10.14	8.7	6.52	5.07	0	0
Llama2-7B-chat	27.4	20.55	10.27	10.96	15.07	15.75	0	0
Llama2-13B	34.75	6.78	14.41	19.49	15.25	9.32	0	0
Llama2-13B-chat	24.66	17.81	14.38	16.44	15.07	11.64	0	0

Table 12: For the Reuters-21578 dataset, the average accuracy (%) of (1, 4, 8, 12)-shot ICL without instructions and (0, 1)-shot ICL with instructions.

Models	1-shot	4-shot	8-shot	12-shot	Instruction 0-shot	Instruction 1-shot
GPT2-XL	2.67	2.44	4.44	23.78	14	14.44
GPT-J	3.33	5.33	12.67	69.56	18	38.44
Llama2-7B	2	4.67	24.22	50.67	26	39.78
Llama2-7B-chat	3.33	22	42.22	78.67	66	48
Llama2-13B	2	3.78	11.11	83.56	26	27.11
Llama2-13B-chat	4	21.33	37.56	84.67	76	35