

Misconfidence-based Demonstration Selection for LLM In-Context Learning

Shangqing Xu and Chao Zhang

Georgia Institute of Technology

{sxu452, chaozhang}@gatech.edu

Abstract

In-context learning with large language models (LLMs) excels at adapting to various tasks rapidly. However, its success hinges on carefully selecting demonstrations, which remains an obstacle in practice. Current approaches to this problem either rely on hard-to-acquire external supervision or require frequent interactions with LLMs, resulting in high costs. We propose a new method called In-Context Reflection (ICR) to overcome these challenges. ICR strategically selects demonstrations to reduce the discrepancy between the LLM’s outputs and the actual input-output mappings. Specifically, ICR starts with a random set of initial demonstrations, then iteratively refines it. In each step, it analyzes a pool of candidate examples and identifies the ones most likely to challenge the LLM’s current understanding, measured by a new metric called misconfidence. These most confusing examples are then selected to replace the less informative demonstrations in the current set. Our comprehensive evaluation across five diverse datasets encompassing 13 subtasks shows the efficacy of ICR. Compared to existing methods, ICR achieves an average performance boost of 4%, while demonstrating remarkable cross-task generalization capabilities.

1 Introduction

In-context learning (ICL, Brown et al. (2020)) enables pre-trained large language models (LLMs) to adapt to diverse tasks by appending question-answer pairs (demonstrations) as prompt contexts. Despite its effectiveness, ICL can be highly sensitive to the quality of the demonstrations (Zhao et al., 2021; Min et al., 2022), emphasizing the need for strategies to strategically select ICL demonstrations.

Existing demonstration selection strategies roughly fall into two categories. One approach first obtains external supervision through preferences of pre-trained encoders or retrievers, then adopts a

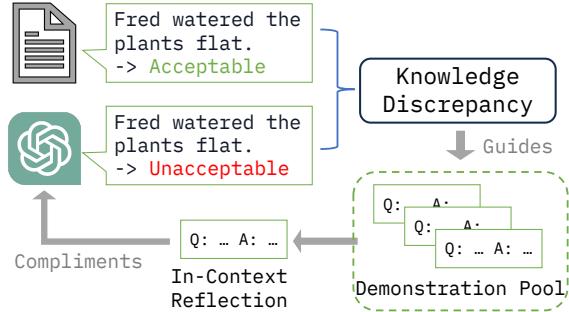


Figure 1: A overview of our method. We aim to leverage the exact discrepancy between LLM’s knowledge and task input-output mappings. Then we select demonstrations that best bridge such discrepancies.

scorer to assign scores for each demonstration candidate based on the supervision. Such scorers could be a semantic distance model (Liu et al., 2022; Gao et al., 2023), a reward function (Rubin et al., 2022; Zhang et al., 2022), or a reversed topic predictor (Wang et al., 2023). The other approach estimates the importance of each candidate by influence analysis, which contrasts LLM predictions before and after adding the candidate to the prompt. The influence can be computed via task-agnostic measures, such as mutual information gain (Sorensen et al., 2022) or validation performance gain (Li and Qiu, 2023; Nguyen and Wong, 2023).

While these approaches have shown promising performance in selecting demonstrations for ICL, they suffer from the following limitations. On the one hand, adopting a scorer depends on reliable external supervision to score accurately. As in-context learning’s mechanism is different from such external supervision, the demonstrations prioritized by the supervision may not be the best choice for forming in-context prompts. Additionally, these methods often require fine-tuning another LLM, which can be computationally expensive. On the other hand, influence analysis by contrast needs one to perform a large number of binary tests with

the LLM, which is costly and unscalable for handling large numbers of candidates.

Our idea is to directly leverage the discrepancy between the output distribution of LLMs and task-specific input-output mappings. This discrepancy arises when the LLM assigns high probabilities to incorrect labels. By constructing ICL prompts that bridge these discrepancies, we aim to calibrate the LLM’s output distribution toward the desired task labels. This strategy is effective because it directly addresses the influence of demonstrations on the LLM through ICL. It is also efficient, as it eliminates the need for repeated binary tests for contrasting.

We present In-Context Reflection (ICR), a new method for selecting effective ICL demonstrations from a certain pool based on LLM *misconfidence*. First, we approximate the aforementioned discrepancies by obtaining LLM’s predictions for each candidate in the pool based on an initial prompt. Candidates that are more confidently misjudged by the LLM (that is, candidates with higher misconfidence) indicate gaps between LLM’s distribution and task mappings, and are therefore prioritized. Consequently, we re-rank all candidates based on their misconfidence and replace the initial prompt with top-ranked ones to construct a refined prompt.

To validate the effectiveness of our method, we conducted experiments across five diverse task sets, encompassing 13 distinct tasks ranging from sentiment analysis to complex language comprehension challenges. Our analysis demonstrates that the prompts generated using our method achieve an average performance improvement of 4% across all tasks. This shows that ICR consistently enhances the LLM’s performance across these tasks. Furthermore, to measure the robustness of ICR, we generate prompts for one dataset and subsequently evaluate them in the same task family. We found that different-task ICR is comparable to same-task uniform sampling, highlighting its potential for broad applications.

Our main contribution includes:

- We propose leveraging the difference between the output distribution of LLMs and the input-output mappings of a given task to address the drawbacks of existing demonstration selection strategies.
- We introduce misconfidence as a metric to quantify this discrepancy and present In-

Context Reflection (ICR), a method that effectively selects demonstrations that provide "lacking knowledge" to help LLMs adapt to specific tasks.

- Through experiments on 13 tasks from 5 task sets, we demonstrate that prompts constructed using ICR are both effective and robust.

2 Related Work

In-context learning (ICL) (Brown et al., 2020) empowers LLMs to rapidly adapt to a wide range of tasks. While ICL proves effective across English-based tasks (Min et al., 2021) and multilingual tasks (Lin et al., 2022), it exhibits significant sensitivity to various factors, including prompt design (Lester et al., 2021), demonstration distribution (Min et al., 2022), instruction design (Mishra et al., 2022), and demonstration ordering (Zhao et al., 2021; Lu et al., 2022). Given these intricate dependencies, it’s crucial to develop advanced demonstration selection.

Following the categorization proposed by Dong et al. (2023), we classify demonstration selection strategies into two categories: 1) adapting a task-specific scorer with external supervision to guide demonstration selection, and 2) contrast-based task-agnostic measures derived from the LLM’s predictions.

Learned Scorers Adapted scorers typically provide pairwise scores between each test case and the pool of candidate demonstrations. Liu et al. (2022) proposed to use Sentence-BERT (Izacard et al., 2021) to generate semantical embeddings, and introduce k-Nearest Neighbors to pick demonstrations. Gao et al. (2023) further enhanced this approach by retrieving candidates whose ground label lies in top-2 zero-shot predictions. Further, Rubin et al. (2022) trained a GPT-Neo as a contrastive scorer as well as a demonstration referrer, and Li et al. (2023) advanced this framework through unified training across various datasets. Ye et al. (2023) introduced Determinantal Point Processes (DPPs) to model the interaction between sequences of demonstrations, which enables retrieving a set of demonstrations.

On the other hand, some approaches are trying to obtain individual scores and build prompts that work for all test cases. Zhang et al. (2022) introduced Q-learning to train a retriever that could actively adapt to previously unseen tasks. Wang

et al. (2023) fine-tuned a smaller LLM as a task-specific token encoder and ranked demonstrations according to their ability to rebuild tokens.

Some studies have also investigated the discrepancy between the output distribution of LLMs and the input-output mappings of tasks. For instance, Gao et al. (2023) first calculate the zero-shot prediction of test cases then retrieve semantically closer candidates whose label lies in the top predictions. Mavromatis et al. (2023) assume each wrongly-judged demonstration could mostly assist in judging cases from its semantic neighborhood, therefore formalizing demonstration selection as max coverage problem. While these approaches share a similar methodology with ours, they rely on semantic distances whereas our method quantifies the misconfidence in LLM outputs.

Contrasting Task-Agnostic Measures A straightforward approach involves randomly selecting demonstrations from the entire candidate pool, as suggested by Min et al. (2022). Sorensen et al. (2022) assessed a prompt sequence by calculating the mutual information between predicted outcomes and true labels. Nguyen and Wong (2023) proposed constructing a validation set and evaluating each train instance by contrasting the validation performance of prompts with and without the instance. Li and Qiu (2023) introduced InfoScore, a computationally efficient pipeline that iteratively filters train samples. However, these methods still require conducting many tests with the LLM, which becomes prohibitively expensive and unscalable when dealing with a large number of candidates.

3 Problem Formulation

We investigate few-shot in-context learning (ICL) with pre-trained LLMs for specific tasks. A target task comprises a train set $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$ and a test set $\mathcal{D}_{test} = \{(x_i, y_i)\}_{i=1}^{N_{test}}$. The data from both the train and test sets are independently and identically distributed (i.i.d), and the labels in the test set are not available except when doing evaluations. In this paper, we limit the task to a single-label classification task, assuming that all labels fall in certain categories $y_i \in Y_{\mathcal{D}}, \forall (x_i, y_i) \in \mathcal{D} = \{\mathcal{D}_{train} \cup \mathcal{D}_{test}\}$.

We predict for the target task via few-shot ICL. Given a pre-trained LLM θ , ICL adapts θ towards a specific target task. Given a input x from

\mathcal{D}_{test} , we concatenate original input x with the demonstrations, changing the output probability into $p_{\theta}(y|x_1, y_1, \dots, x_n, y_n, x)$. We denote the probability of y given x and demonstration \mathcal{P} as $p_{\theta}(y|x, \mathcal{P})$.

The demonstrations must be carefully selected from a candidate pool $\mathcal{C} \subseteq \mathcal{D}_{train}$, forming a subset $\mathcal{P} = \{(x_j, y_j)\}_{j=1}^m \subset \mathcal{C}, m \ll N_{train}$. We assume that the model parameters θ remain fixed throughout the process, and only the selection of \mathcal{P} is modified. The success of this adaptation is measured by the predictive accuracy, namely whether the output generated by the LLM, $y_{\theta}(x_i) = \text{argmax}_{y \in Y_{\mathcal{D}}} p_{\theta}(y|x_I, \mathcal{P})$, matches the actual label y_i . The central challenge of demonstration selection is to identify the optimal subset $\mathcal{P} \subset \mathcal{C}$ that yields the most significant improvement in prediction accuracy, measured as $\text{Acc}(y_i, y_{(\theta, \mathcal{P})}(x_i)), \forall (x_i, y_i) \in \mathcal{D}_{test}$.

4 Method

4.1 In-Context Learning by Bridging Discrepancy

Let us first define discrepancy in our ICR method. While trying to adapt a LLM θ for \mathcal{D} , there can be cases where θ 's output y_{θ} doesn't match the actual label y . We say there is *discrepancy* between LLM's output probability p_{θ} and task's input-output mappings $\{x_i \rightarrow y_i, (x_i, y_i) \in \mathcal{D}\}$. By minimizing such discrepancy, LLM would always output correct labels for each input, therefore adapting θ to \mathcal{D} .

The objective of ICL is equivalent to minimizing discrepancy. As shown by recent studies (Zhao et al., 2021; Min et al., 2022; Wei et al., 2023), ICL's demonstrations \mathcal{P} contribute by guiding θ to mimic mappings $\{x_i \rightarrow y_i, (x_i, y_i) \in \mathcal{P}\}$, and to generate output $y_{(\theta, \mathcal{P})}$ for x correspondingly. As most ICL methods select \mathcal{P} that are representative of \mathcal{D} (Min et al., 2022), such ICL methods are therefore equal to bridge the discrepancy between p_{θ} and $\{x_i \rightarrow y_i, (x_i, y_i) \in \mathcal{D}\}$.

Now we propose how to select ICL demonstrations that can best bridge such discrepancies. Consider a prompt \mathcal{P} . If the discrepancy between $\{x_i \rightarrow y_i, (x_i, y_i) \in \mathcal{P}\}$ and p_{θ} is small, \mathcal{P} should have limited influences on p_{θ} as its mappings are already obtained by θ . In this case, mimicking $\{x_i \rightarrow y_i, (x_i, y_i) \in \mathcal{P}\}$ would barely change $p_{\theta}(y|x, \mathcal{P})$ from $p_{\theta}(y|x)$. This inference also holds reversely. Therefore, to bridge the discrepancies

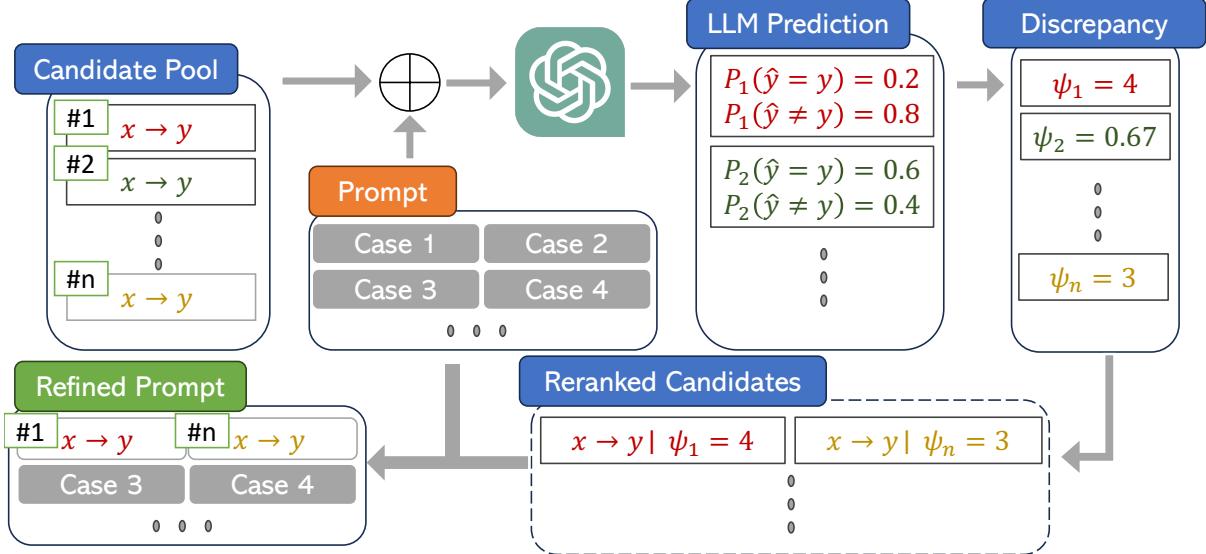


Figure 2: An overview of In-Context Reflection (ICR). We first use an initialized prompt to get LLM prediction for each candidate, then calculate the misconfidence score ψ to measure the discrepancy between LLM and task. After that, we rerank all candidates according to ψ , and replace part of the prompt with the top-ranked candidates, obtaining a refined prompt.

between p_θ and $\{(x_i \rightarrow y_i, (x_i, y_i)) \in \mathcal{D}\}$, we can select \mathcal{P} whose mappings have the largest discrepancy between p_θ .

We propose that the discrepancy between the set of input-output pairs $\{(x_i, y_i)\}$ and the model’s predictions $p_\theta(x)$ can be approximated by the misconfidence associated with each case (x_i, y_i) with respect to the model θ . Intuitively, if a case (x_i, y_i) is easily misclassified by the model (i.e., $p_\theta(x_i) \neq y_i$), we consider the misconfidence of that case to be high. High misconfidence indicates that the model struggles to correctly predict the output label y_i for the input x_i . Consequently, the overall discrepancy between the observed data $\{(x_i, y_i)\}$ and the model’s predictions $p_\theta(x)$ is expected to be high.

We thus quantify the misconfidence of a model by measuring the margin between the highest probability assigned to any incorrect label, $\max_{y \in Y, y \neq y_i} p_\theta(y|x_i)$, and the output probability of the correct label, $p_\theta(y_i|x_i)$. This margin reflects how confidently the model misjudges the true label from plausible alternatives. We denote this score as $\psi((x_i, y_i), \theta)$ and compute it as:

$$\psi((x_i, y_i), \theta) = \frac{\max_{y \neq y_i, y \in Y} p_\theta(y|x_i)}{p_\theta(y_i|x_i)} \quad (1)$$

Further, given an initial prompt \mathcal{P}_0 , we can compute the probability conditioned on these demonstrations, which yields the misconfidence score:

$$\psi((x_i, y_i), (\theta, \mathcal{P}_0)) = \frac{\max_{y \neq y_i, y \in Y} p_\theta(y|x_i, \mathcal{P}_0)}{p_\theta(y_i|x_i, \mathcal{P}_0)} \quad (2)$$

Such prompt-based misconfidence score helps us select candidates that can enhance \mathcal{P}_0 .

4.2 In-Context Reflection (ICR)

To effectively adapt the model parameter θ to a specific task, we select demonstrations based on their ψ scores (Equation 1). We introduce the In-Context Reflection (ICR) pipeline, which uses this strategy to efficiently construct an optimal demonstration set $\hat{\mathcal{P}}$.

The ICR pipeline begins with an initial demonstration set $\mathcal{P}_0 = \{(x_1, y_1), \dots, (x_m, y_m)\}$, which is randomly sampled from the candidate pool \mathcal{C} . In each iteration, ICR updates the misconfidence score for all candidates based on the current demonstration set. Then, it reranks the candidates according to their misconfidence and replaces n of the previous demonstration set with these top-ranked candidates. The entire process is provided in Algorithm 1, where $+(-)$ denotes set-wise merging (excluding). Instead of solely re-ranking, we use iterative replacement to build the prompt, which has been shown crucial for obtaining a semantic distribution from \mathcal{D} (Min et al., 2022). Note that,

ICR only requires one interaction with the LLM per train case, making it computationally efficient.

Algorithm 1: In-Context Reflection

Data: LLM θ , Candidate Pool \mathcal{C} ,
 Demonstration size m , Replacing Number n
 Iteration Number k

Result: Optimal Prompt $\hat{\mathcal{P}}$

```

 $\mathcal{P}_0 = \phi;$ 
for  $i = 1$  to  $m$  do
    Sample  $(x, y) \sim U(\mathcal{D})$ 
     $\mathcal{P}_0 = \mathcal{P}_0 + \{(x, y)\}$ 
end
 $\mathcal{C} = \mathcal{C} - \mathcal{P}_0;$ 
for  $i = 0$  to  $k - 1$  do
    for  $(x, y) \in \mathcal{C}$  do
        | Calculate  $\psi((x, y), (p_\theta, \mathcal{P}_i))$ 
    end
    Rerank  $(x, y) \in \mathcal{C}$  according to  $\psi$ ;
     $\mathcal{P}_{i+1} = \mathcal{C}[1 : n] + \mathcal{P}_i[n + 1 : m];$ 
    Add the replaced to pool
     $\mathcal{C} = (\mathcal{C} - \mathcal{P}_{i+1}) + \mathcal{P}_i;$ 
end
 $\hat{\mathcal{P}} = \mathcal{P}_k;$ 

```

5 Experiment

5.1 Settings

5.1.1 Datasets

We evaluate ICR on 5 task sets containing 13 binary or multi-class classification tasks, detailed as follows. For GLUE, Ethos, and TweetEval, we only select part of the tasks, as other tasks contain too many test cases or are too easy for our backbone LLM to solve. Details are shown in Appendix A. **GLUE** (Wang et al., 2018) A multiple-task generalization benchmark covering topics from hypothesis to fact-checking. We adopt four subtasks: *MRPC*, *WNLI*, *COLA*, *RTE*.

Ethos (Mollas et al., 2022) A collection of hate speech detection tasks from online texts. We adopt four subtasks: *Religion*, *Race*, *Gender*, *Directed_vs_generalized*

TweetEval (Barbieri et al., 2020) A multiple-task benchmark built from Twitter, all framed as multi-class classification. We adopt three subtasks: *hate*, *emotion*, *irony*

HateSpeech18 (de Gibert et al., 2018) A binary-labeled hate speech dataset extracted from a white supremacist forum.

Poem Sentiment (Sheng and Uthus, 2020) A multi-class sentiment dataset of poem verses from Project Gutenberg.

5.1.2 Baselines

We compare with the following baselines:

Uniform Sampling (Min et al., 2022) We uniformly sample demonstrations stratifying the origin label distribution from the full candidate pool.

Best-of-10 (Zhang et al., 2022) We randomly sample 10 sets of demonstrations and select the best one by evaluating on a 100 validation subset.

Topic (Wang et al., 2023) We fine-tune a GPT-2-Large-774M model to encode task-specific latent concept tokens, then select demonstrations whose in-context prompts best predicted the concept. We only adopt this method on GLUE, Ethos, and PoemSentiment, as there are no concept data available for other tasks.

KATE (Liu et al., 2022) We introduce a pre-trained SBERT (Reimers and Gurevych, 2019) to calculate semantic embeddings for both candidate pool and test set. Then, for each input case from the test set, we retrieve the k-nearest neighbors from the candidate pool as the demonstrations.

AMBIG (Gao et al., 2023). For each test case, we perform zero-shot prediction and identify the labels in the top two predictions as the ‘Ambiguity label’. We then filter candidates with ground labels matching the Ambiguity label and choose semantically similar demonstrations from this subset.

5.2 Evaluation and Implementation Details

We use GPT-3.5-Turbo-Instruct as the backbone and use the same prompt format for all the methods. Appendix B shows the task details as well as prompt formats. By default, we select 16 demonstrations for all methods.

In each task, we use the full train set as the candidate pool for 1) Uniform Sampling, 2) KATE 3) AMBIG 4) Best-of-10, but we restrict Topic’s candidate pool to a uniformly selected subset with a size of 500 to cut computational cost. To provide a fair comparison, we also restrict ICR’s pool samely. Evaluation is applied on test sets, except GLUE, where we evaluate methods on the validation set as there is no publicly available test label. We calculate both the macro-average F1 score and the accuracy score.

On all the tasks, we only adopt exactly one iteration of ICR and set the replacement count n as 8, meaning that we will replace 8 demonstrations out

of the original prompt by misconfidence re-ranking. We will provide results for multiple iterations in section 5.5.1.

5.3 Same-Task Evaluation

We show the same-task evaluation results in Table 1. All the reported scores are the average of three independent runs with different random seeds, except for KATE and AMBIG, as they do not involve any randomness. For Ethos, TweetEval, and GLUE, as they involve multiple subtasks, we report the average score of the subtasks.

The results show that ICR outperforms all baselines (both scorer-based and contrasting-based) on all tasks with significant 4% improvements. Our method solely relies on the candidate pool and corresponding LLM judgment, without employing any external knowledge base. Such an improvement shows the efficacy of leveraging discrepancies while building in-context prompts.

In contrast, AMBIG performs poorly on all tasks compared to vanilla KATE, not to mention ICR. While it was claimed that this method can effectively capture clues from the LLM’s distribution, the experimental results indicate that it relies more heavily on the SBERT semantic encoder (we will discuss this further in section 5.5.4). Conversely, ICR with an originally designed ψ score achieved much better results, showing that it can measure the discrepancy between LLM distribution and the task labels more accurately.

5.4 Different-Task Evaluation

To further test the robustness of ICR, we build ICR prompts on each task of GLUE and evaluate them on different tasks from the same task family. We compare them to uniform prompts created for the same task. Figure 3 shows the result. Even when selecting demonstrations from different task sets, ICR obtains comparable (or sometimes even superior) results compared with same-task uniform sampling prompts. Also, we notice that the performance gains between MRPC and WNLI are much higher than the rest, implying some latent correlations between these tasks.

5.5 Ablation

5.5.1 Multiple ICR Iterations

We investigate whether multiple ICR iterations yield better results. We initialize a random prompt and apply ICR iteratively for 5 iterations. Results

	COLA	-1.1	-2.2	0.64	-2.8
TRAIN	MRPC	-1.1	9.6	-1.4	4.2
RTE	-0.77	4.2	1.8	2.8	
WNLI	0.67	12	-1.4	4.2	
TEST	COLA	MRPC	RTE	WNLI	

Figure 3: Different-task evaluation accuracy of ICR’s prompt on GLUE’s tasks. Each number shows the performance gain compared with same-task uniform sampling. On all tasks, ICR received comparable (sometimes even superior) results.

on GLUE-MRPC and TweetEval-Emotion are presented in Figure 4. While ICR always contributes positively, each iteration does not consistently improve the performance. One possible reason is that in each iteration, the ICR update is too rough and large, causing the result to fluctuate around the global optimum. Therefore, we choose to set the number of iterations to one.

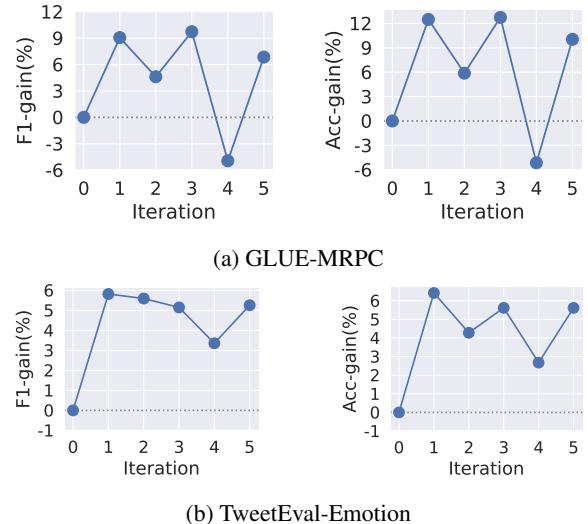


Figure 4: Performance gain of ICR on random prompt for 5 iterations. ICR produces a positive effect most times but is unstable.

5.5.2 Relationship between Misconfidence and Performance

One key rationale for ICR is that demonstrations with larger misconfidence lead to better contribution. To confirm this idea, we build prompts with demonstrations of different misconfidence averages on a) Poem Sentiment and b) GLUE-MRPC, and evaluate them on the same task. The result is shown

	Macro-F1					
	GLUE	Ethos	TweetEval	HateS18	Poem	Average
Uniform	75.5	65.5	63.7	63.7	68.7	67.4
Best-of-10 (Zhang et al., 2022)	75.8	69.1	68.8	70.6	72.2	71.3
Topic (Wang et al., 2023)	76.2	62.4	-	-	75.2	-
KATE (Liu et al., 2022)	72.3	71.2	66.3	66.8	73.2	69.9
AMBIG (Gao et al., 2023)	76.6	71.3	68.3	72.4	67.7	71.3
ICR (ours)	78.7	76.5	71.0	74.4	76.5	75.4

	Accuracy					
	GLUE	Ethos	TweetEval	HateS18	Poem	Average
Uniform	76.6	70.7	64.5	74.2	70.2	71.2
Best-of-10 (Zhang et al., 2022)	77.3	75.3	69.8	82.6	74.0	75.8
Topic (Wang et al., 2023)	77.6	70.1	-	-	76.9	-
KATE (Liu et al., 2022)	73.4	76.7	68.1	78.2	76.0	74.5
AMBIG (Gao et al., 2023)	78.0	76.5	69.3	83.8	76.0	76.0
ICR (ours)	80.6	82.2	71.6	87.0	78.9	80.0

Table 1: Results on each task set. ICR outperforms all baselines with an average 4% gain. It is an exciting result, as ICR uses no fine-tuning data and requires linearly scaled interactions with LLM.

in Figure 5. We see performance of prompts generally is consistent with the demonstrations’ misconfidence average. Also, it is interesting that the demonstrations with extremely low misconfidence (that is, they are correctly judged confidently) show better contributions than borderline ones. It shows that LLM can also be enhanced by further distinguishing confident knowledge.

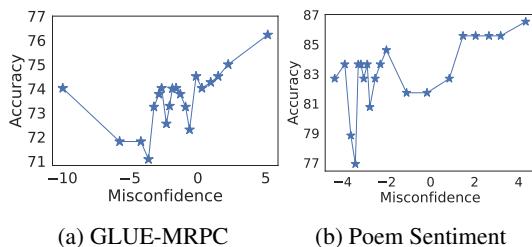


Figure 5: Visualization of the result from demonstrations with different misconfidence levels on a) GLUE-MRPC b) Poem Sentiment. Demonstrations with larger misconfidence lead to better in-context performance.

5.5.3 Initialization and Prompt Building

As described in Section 4.2, our algorithm first calculates misconfidence through a few-shot prompt, then does replacements to build a more powerful one. It is different from most similar studies, where they calculate measurements (or scores) by zero-shot prompts and build refined prompts entirely from such measurements. We therefore introduce two corresponding ablations.

Initialization We update the misconfidence using a zero-shot prompt instead of the original few-shot prompt. Then we build ICR prompts and compare their performance to the original ones. The result is shown in Table 2. We see the performance drops severely in most task sets. Given the result in section 5.5.2, we conclude that few-shot prompts can provide more reasonable misconfidence estimations, which makes ICR perform better.

	GLUE	Ethos	TEval	HS18	Poem
F1	-5.1	-10.0	-5.6	-6.0	-6.1
Acc	-4.4	-9.9	-4.9	-7.6	-6.7

Table 2: Performance change of ICR using zero-shot misconfidence instead of original few-shot ones. Using zero-shot misconfidence leads to significant drawbacks.

Prompt Building We select all demonstrations according to misconfidence instead of replacing part of initialized prompts. The result is shown in Table 3. Referring solely to misconfidence leads to a significant performance drop, except on GLUE. Note that, GLUE’s labels are well-balanced, but other tasks’ are not. Recalling conclusions from Min et al. (2022), we see such performance drop is caused by a lack of label distribution information in the prompt. Therefore, building ICR prompt through partial replacement maintains label distribution information, and therefore is better than building solely from consufability re-ranking.

	GLUE	Ethos	TEval	HS18	Poem
F1	-1.7	-7.9	-6.8	-10.4	-7.7
Acc	-0.6	-10.4	-5.1	-12.0	-8.7

Table 3: Performance change of ICR by selecting demonstration entirely from misconfidence instead of replacement. This leads to performance drops on all task sets except GLUE, as GLUE is highly balanced and therefore less affected by demonstration distribution.

5.5.4 Influence of Semantic Distances

In addition to ICR, several studies (Mavromatis et al., 2023; Gao et al., 2023) have suggested selecting demonstrations jointly based on LLM’s output distribution and semantic distances. For instance, Mavromatis et al. (2023) propose a method where they assume that borderline candidates have a strong influence on their semantic neighbors. However, when selecting ICL demonstrations, semantic distances and distribution measurements (like misconfidence) are independent of each other. We will prove this through a simple ablation experiment.

First, we compute the zero-shot judgments and ICR few-shot judgments for all test cases. Next, we identify any cases where the judgment has changed. Finally, we record semantic distances between each test case and the prompt demonstrations. We want to check if semantic distances have a certain relationship with judgment changes.

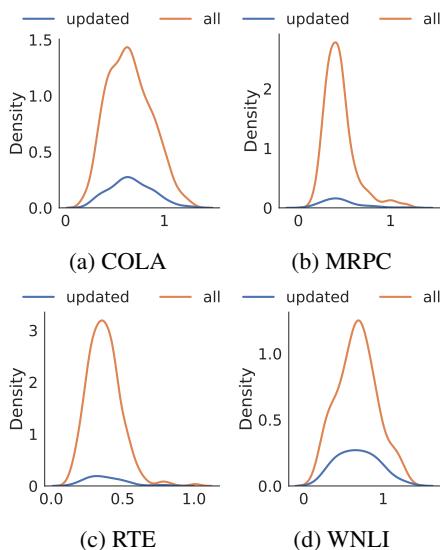


Figure 6: Distribution of semantic distance between demonstrations and 1) all test cases 2) test cases whose judgment updated between zero-shot and few-shot prompts. There is no significant relationship between judgment updates and semantic distances.

Figure 6 illustrates the result. The distances of the modified cases are distributed evenly from the original distribution. This indicates that even if a test case is closer to one of the demonstrations, it does not have a higher chance of being correctly judged. This shows that the in-context influences of ICR prompts are independent of the semantic distances. As a result, the strategies mentioned above primarily rely on semantic features and do not leverage the distribution of the LLM.

5.5.5 Case Study on ICR Improvement

To show how ICR bridges the discrepancy and further improves the task performance, we apply ICR on GLUE-MRPC, visualize the distribution of ψ on demonstration candidates, and show the changes in LLM’s prediction on test set. As results in Figure 7, ψ scores in the candidate pool show that cases with label 1 tend to be misjudged confidently. Therefore, ICR replaces part of the initial prompt with misjudged 1-labeled cases, bridging the gap between LLM’s prediction and task mappings. This improves LLM’s judging accuracy (especially on 1-labeled ones) significantly.

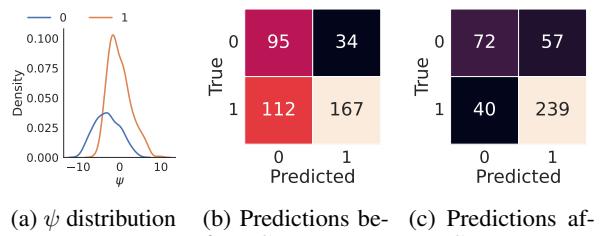


Figure 7: Distribution of ψ score on GLUE-MRPC’s candidate pool, and the LLM predictions on test set before (after) apply ICR update. ICR sees more cases with label 1 being confidently misjudged. After refining the prompt with such cases, LLM’s accuracy in judging case 1 is improved significantly.

6 Conclusion

We studied the problem of selecting demonstrations for in-context learning. To tackle the limitations of existing methods, we proposed to leverage the discrepancies between LLM’s knowledge and task expectations directly. We proposed In-Context Reflection (ICR), a novel strategy that quantifies such discrepancy through misconfidence measurement. Experiments showed ICR’s prompt received an average 4% gain on tasks. Also, ICR received comparable performance when evaluated on tasks from the same task family, proving that ICR is robust.

Still, ICR has some limitations. More ICR iterations do not always improve the prompt. Also, performing ICR requires a fully labeled subset as a candidate pool. Future work can either investigate how to gain stable improvements with more iterations or try to build strategies jointly using discrepancy-based metrics with other measurements.

References

- Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesh Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Xiangru Tang, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. *Promptsource: An integrated development environment and repository for natural language prompts*.
- Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. *TweetEval: Unified benchmark and comparative evaluation for tweet classification*. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.
- 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.
- 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.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. *A survey on in-context learning*.
- Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023. *Ambiguity-aware in-context learning with large language models*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. *The power of scale for parameter-efficient prompt tuning*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvyा Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierrick Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. *Datasets: A community library for natural language processing*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. 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. *arXiv preprint arXiv:2305.04320*.
- Xiaonan Li and Xipeng Qiu. 2023. *Finding support examples for in-context learning*.
- Xi Victoria Lin, Todor Miyaylov, Mikel Artetxe, Tianlu Wang, Shuhui Chen, Daniel Simig, Myle Ott, Namnan Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. *Few-shot learning with multilingual generative language models*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. *What makes good in-context examples for GPT-3?* In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- 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*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jian Zhang, Huzeifa Rangwala, Christos Faloutsos, and George Karypis. 2023.

Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to gptk’s language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigoris Tsoumakas. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 8(6):4663–4678.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 93–106, Barcelona, Spain (Online). Association for Computational Linguistics.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE:

A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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

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

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

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

A Task selection

We only adopt part of the tasks under GLUE, Tweet-Eval, and Ethos. One reason is that the GPT-3.5 backbone already performs high scores on the other tasks under uniform-sampling prompts. Therefore, it is hard to show our method’s superiority under such tasks. Table 4 shows the result on GLUE-SST2 (SST2), Ethos-Sexual Orientation (E-SO), Ethos-Violence (E-V), Ethos-National Origin (E-NO), and Ethos-Disability (E-D). We didn’t adopt GLUE-MNLI, GLUE-QNLI, GLUE-QQP, TweetEval-Emoji, and TweetEval-Sentiment as they contain too many test cases (9.8K, 5.8K, 293K, 50K, and 12.3K respectively) for us to afford. And we didn’t adopt GLUE-STSB, as it is a numerical inference dataset, unsuitable for our general scope around classification tasks.

Table 4: Performance of Uniform Sampling prompts on the unselected tasks. The performances are either too high or too low, and therefore we didn’t select these tasks.

Task	SST2	E-SO	E-V	E-NO	E-D
F1	95.1	100	91.0	90.2	89.7
Acc	95.0	100	94.3	95.4	90.8

B Datasets and Prompting Template

Sizes of all tasks in our experiments are shown in Table 5. We follow the same data division as of Huggingface Datasets (Lhoest et al., 2021), except for HateSpeech18, where we perform a custom division as only one data set is provided. The prompting templates used for each subtask are shown in Table 6. We adopt a wide range of prompts from Bach et al. (2022).

C Hardware Details

We use the Azure GPT-3.5-turbo-Instruct server to accomplish most experiments except tasks related to hate speech detection, where we use the original OpenAI API due to Azure’s content filter policy. We use one Nvidia RTX A5000 GPU to hold pre-trained SBERT (as in KATE and AMBIG) or GPT2 (as in Topic).

Table 5: Dataset details

Task	Train Size	Test Size
GLUE-COLA	8551	1043
GLUE-MRPC	3668	408
GLUE-WNLI	635	71
GLUE-RTE	2490	277
Ethos-Religion	346	87
Ethos-Race	346	87
Ethos-Gender	346	87
Ethos-Directed vs Generalized	346	87
TweetEval-Hate	9000	2970
TweetEval-Emotion	3257	374
TweetEval-Irony	2862	955
Hate Speech18	8755	500
Poem Sentiment	892	104

Table 6: Prompt Examples

Task	Prompt Example
GLUE-COLA	[Sentence] \n Is this example grammatically correct and sensible?\n [yes/no]
GLUE-MRPC	Do the following two sentences mean the same thing?\n [Sentence 1]\n [Sentence 2]\n [yes/no]
GLUE-WNLI	Entailment means that the second sentence follows from the first sentence. Are the following two sentences an example of entailment?\n [Sentence 1]\n [Sentence 2]\n [yes/no]
GLUE-RTE	Does "[Sentence 1]" imply that "[Sentence 2]"? Please answer either yes or no.\n [yes/no]
Ethos	Text: [Sentence] \n [Religious/Racial/Gender/Generalized] Hate: [yes/no]
TweetEval-Hate	Text: [Sentence] \n Hate: [yes/no]
TweetEval-Emotion	[Sententce] \n \n What is the emotion of the text?\n Hint: anger, joy, optimism, sadness \n [anger/joy/optimism/sadness]
TweetEval-Irony	Is this tweet is ironic? \n \n [Sentence]\n [yes/no]
Hate Speech18	Text: [Sentence] \n Hate: [yes/no]
Poem Senti-ment	[Sentence] Is the sentiment the poet expresses for the poem negative, positive, neutral, or mixed? \n \n [negative/positive/neutral/mixed]