From Fake to Real: Mitigating Out-of-Distribution Bias in In-Context Learning via Feedback Supervision from Large Language Models

Rui Song[1], **Yingji Li**[1], **Jian Li**[2], **Fausto Giunchiglia**[1,3], **Hao Xu**[1*]
[1]College of Computer Science and Technology, Jilin University, China
[2]Solar System of OVB, Tencent, China
[3]Department of Information Engineering and Computer Science, University of Trento, Italy
{songrui,yingjili,xuhao}@jlu.edu.cn,
loucasli@tencent.com, fausto.giunchiglia@unitn.it

Abstract

With the rapid development of Large Language Models (LLMs), In-Context Learning (ICL) has emerged as one of the universal paradigms for unleashing the capabilities of LLMs. However, LLMs are generally plagued by various biases in context example selection, which can distort the model's predictions. Although extensive research has focused on designing heuristic sample selection methods to mitigate biases in ICL, these approaches often struggle to adapt to highly biased out-of-distribution (OOD) scenarios with significant shifts between test samples and context samples. To overcome the aforementioned issue, this paper proposes a LLM-driven iterative derivation method for OOD data pseudo-labeling (named **LPL**), aiming to mitigate the risk of performance degradation caused by OOD bias by avoiding direct use of source data. To mitigate the misleading effects of noise in pseudo-labels, we propose a filtering metric that integrates model confidence and perturbation perplexity to enhance the quality of pseudo-labels. Subsequently, in each iteration, LPL utilizes this metric to expand new pseudo-labeled data as contextual demonstrations and ultimately adopts a voting mechanism to ensure the stability of the predictions. A series of experiments conducted on various LLMs have confirmed that our proposed method can effectively reduce OOD biases, thereby opening up new avenues for research in ICL biases[1].

1 Introduction

In-Context Learning (ICL) can motivate the power of Large Language Models (LLMs) with just a few examples, and has therefore become one of the most common methods for complex Natural Language Understanding tasks (Liu et al., 2022; Xu et al., 2025). Unlike traditional supervised learning, ICL does not involve parameter updates. Instead, it relies on LLMs to learn the patterns hidden in the demonstrations and make correct predictions based on them. Therefore, the quality of the demonstrations is particularly crucial to the performance of ICL (Dong et al., 2024). However, recent research has shown that ICL is very sensitive to the selection of context demonstrations, causing LLMs to be biased towards predicting certain answers (Zhang et al., 2022). Even permutations of the same samples can have an uncertain impact on the results (Zhao et al., 2021). Therefore, systematically discussing the various biases in ICL and proposing feasible mitigation strategies has become an important topic in current LLM studies.

Bias in ICL has multiple categories and can be traced to different causes, including vanilla label bias (Zhao et al., 2021), context-label bias (Tang et al., 2023), and domain-label bias (Fei et al., 2023; Yuan et al., 2023). Intuitively, biases in ICL can be reduced by selecting more reasonable demonstrations, which rely on both the model and the data (Peng et al., 2024a). A series of techniques such as word-overlap similarity (Luo et al., 2023), representational similarity (Liu et al., 2022), informativeness score (Xu and Zhang, 2024) and joint distribution (Ye et al., 2023) are used to find reasonable demonstrations. Another approach determines important samples by quantifying the contribution of demonstrations to LLMs' prediction results (Nguyen and Wong, 2023; S. et al., 2024; Askari et al., 2025). Although effective in many cases, these approaches may fail when faced with large distribution shifts (Fei et al., 2023; Yuan et al., 2023) on the domain bias shown in Figure 1.

Figure 1 demonstrates the distribution of the datasets and the performance differences of various ICL sample selection methods through Kernel Density Estimation (Chen, 2017), specifically when the *amazon* dataset is used as the context for generalization to the other three datasets (Yuan et al., 2023). We observe that when the differences in

*Corresponding author.
[1]https://github.com/songruiecho/LPL

Findings of the Association for Computational Linguistics: ACL 2026, pages 15197–15213

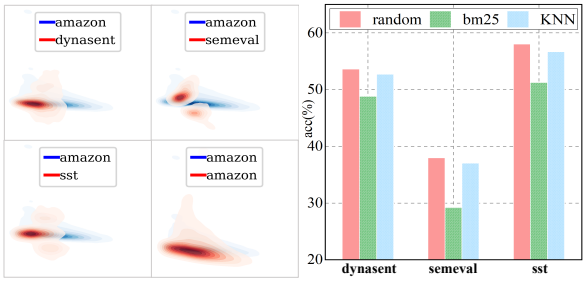

Figure 1: Data distribution visualization and sample selection methods for different datasets, blue represents the source set, red represents the target set, and the overlap area indicates the distributional similarity between datasets. A detailed description of the datasets and tasks can be found in Section 4.

data distribution are significant, similarity-based sample selection methods can even lead to performance degradation compared to random selection. This observation inspires us to develop more reasonable sample selection methods to enhance the performance under the premise of OOD bias.

The paper presents a **L**LM-driven iterative derivation method for OOD data **P**seudo-**L**abeling (**LPL**), which avoids the negative impact of OOD samples by utilizing test samples with pseudo-labels as demonstrations. LPL is based on a trade-off between pseudo-label noise and OOD bias, where, within a certain range of noise contained in the labels, the performance gain from using pseudo-labeled samples as demonstrations outweighs the performance loss under OOD bias. To obtain pseudo-labels with as little noise as possible, we propose a pseudo-label quality quantification metric based on **C**onfidence and **P**erturbation **P**erplexity (**CPP**), aiming to select high-quality labeled samples from the target domain with unknown labels as the initial sample pool for LPL. Subsequently, LLMs select demonstrations from the sample pool based on semantic similarity and iteratively re-label the samples to update the pool, enriching its sample diversity. To maintain the stability of ICL performance during the iterative process, a majority voting mechanism is used on the prediction results from multiple iterations to determine the final prediction (Section 3). The experimental results of several different LLMs on NLP tasks confirm the effectiveness of the proposed method in alleviating OOD bias (Section 4). Our contribution can be summarized as follows:

- We propose CPP, a quantification method based on confidence and perturbation perplexity, to refine the model's pseudo-labeling results.
- We demonstrate the noise tolerance capability of ICL through theoretical analysis, which supports the rationality of CPP for pseudo-label construction.
- We design an iterative process involving LLMs, gradually expanding the selection range of pseudo-labels and obtaining more robust prediction results through majority voting.
- Experimental results from multiple NLP tasks and LLMs confirm that the proposed method can effectively mitigate OOD biases in ICL.

2 Related Work

In-Context Learning. ICL enables LLMs to learn tasks from a few demonstrations (Sun et al., 2022; Yu et al., 2023), making effective sample selection a central challenge (Dong et al., 2022). Early approaches typically rely on similarity-based retrieval, such as cosine similarity (Liu et al., 2022; Qin et al., 2023), or incorporate diversity signals using graphs and confidence scores (Su et al., 2023). More refined metrics, including perplexity (Gonen et al., 2023) and misconfidence (Xu and Zhang, 2024), have also been introduced. However, these samples remain suboptimal because they do not directly reflect their impact on LLM predictions. To address this issue, recent work evaluates sample contributions through model feedback without gradients. InfoScore (Li and Qiu, 2023) and influence-based methods (Nguyen and Wong, 2023) estimate both helpful and harmful examples, though the latter is NP-hard due to the large search space. Consequently, subsequent studies reduce the search scope via similarity-based pre-selection (Wu et al., 2023; Peng et al., 2024a), or train influence estimators on general datasets to eliminate expensive search procedures (S. et al., 2024; Askari et al., 2025).

OOD Robustness of LLMs. Real-world NLP data often violate the i.i.d. assumption, posing substantial challenges for OOD robustness (Arora et al., 2021). Although LLMs demonstrate advantages in OOD scenarios, their robustness remains limited, especially when distribution gaps are large (Wang et al., 2024; Yuan et al., 2023). Improving OOD robustness is therefore essential for the safe deployment of AI systems. Existing approaches refine instructions—e.g., linguistic rule-based prompt construction—to strengthen causal

feature extraction (Jiang et al., 2024), but instruction tuning is considerably more costly than ICL. As a lighter-weight alternative, recent work employs semantic rewriting to generate samples closer to target domain, thereby reducing distribution discrepancies (O'Brien et al., 2024; Madine, 2024).

Pseudo-label Augmentation. Pseudo-labeling is a central technique in semi-supervised learning, enabling models to leverage large amounts of unlabeled data by assigning and using model-generated labels (Yarowsky, 1995). Consequently, identifying high-quality pseudo-labels has been a core research problem. Prior work improves pseudo-label reliability through confidence-based selection (Shi et al., 2018), uncertainty-aware filtering (Rizve et al., 2021), noise-correction frameworks (Wang and Wu, 2020), and class-balanced confidence estimation (Li et al., 2023). Class-distribution awareness has further been shown to enhance pseudo-label quality (Xie et al., 2024). Recently, LLMs have also been introduced to reduce pseudo-label noise and improve label quality (Ding et al., 2025).

3 Methodology

Our motivation is to mitigate OOD bias as much as possible by replacing source domain data with target domain data, based on the theoretical analysis presented in Section 3.1. In this setting, although target domain data is accessible in related cross-domain studies, it lacks available labels (Wu and Shi, 2022; Song et al., 2024a). We propose LPL, a prudent approach to generating low-noise pseudo-labels as much as possible under unknown label conditions. LPL includes three parts: Initial Samples Selection, Iterative Pseudo-label Generation and Majority Voting as shown in Figure 2.

3.1 Label Noise versus OOD Bias

LPL can be regarded as a model that leverages in-distribution data with pseudo-label noise to replace data with OOD bias in constructing in-context learning. Therefore, we can demonstrate how pseudo-labels help the model, even with noise, by comparing the theoretical model error bounds before and after the replacement. Before this, we leverage the theory of (Dai et al., 2023) to interpret ICL as a meta-optimizer capable of performing gradient descent and provide the corresponding analysis from the perspective of an optimizable model. First, the generalization error for in-distribution cases can be easily given by Theorem 1 (Shalev-Shwartz and Ben-David, 2014).

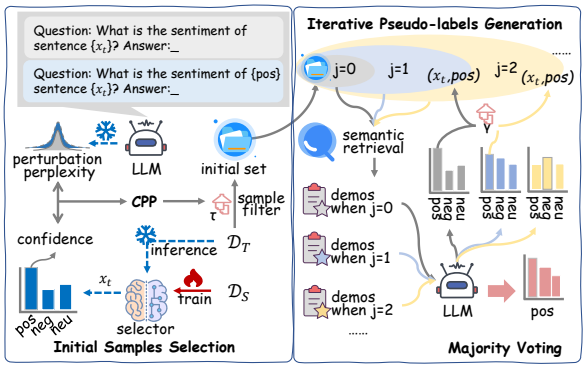

Figure 2: LPL execution process. Arrows of different colors represent the execution at different iterations.

Theorem 1. *For a binary classification problem, when the hypothesis space is a set of a finite number of functions $\mathcal{F} = \{f_1, f_2, ..., f_d\}$, for any function $f \in \mathcal{F}$, with probability at least $1-\delta, 0 < \delta < 1$, the following inequality holds:*

$$R(f) \leq \hat{R}(f) + \epsilon, \tag{1}$$

where $R(f)$ denotes expected risk regarding f, $\hat{R}(f)$ denotes the empirical risk, $\epsilon = \sqrt{\frac{1}{2Num}(\log d) + \log \frac{1}{\delta}}$, Num is the sample size and d is the complexity of the hypothesis space.

Lemma 1. *Following the definition in Thm 1, if noise with probability p is introduced to the samples, with probability at least $1-\delta, 0 < \delta < 1$, the following inequality holds:*

$$R(f)_{noise} \leq p + (1-2p)(\hat{R}(f) + \epsilon). \tag{2}$$

The upper bound derived in the lemma quantifies the impact of pseudo-label noise on model generalization error, demonstrating that the model can still maintain reliable performance within a controllable range even in the presence of noise. Based on this, we further give the corollary to show the comparability between label noise and OOD bias:

Corollary 1. *Under relaxed conditions, if the probability of label noise is as low as possible, then in the presence of strong OOD bias, constructing demonstrations using target domain data with label noise will have a smaller negative impact.*

3.2 Definition

For the convenience of formal description, we first provide some definitions. Given a LLMs $\mathcal{M}$, ICL aims to predict the label y_t of $x_t \in \mathcal{D}_T$ with several

input labeled samples $\{(x_i, y_i)\}_{i\in[1,C\times N]}$, where C is the class number and N is the number of samples per category. For generative tasks without explicit categories, C degenerates to 1. We emphasize that the number of each different category is equal to avoid the bias caused by imbalanced labels (Zhao et al., 2021). Subsequently, $\mathcal{M}$ are expected to predict the labels of test samples based on the aforementioned samples:

$$y_t = \ddot{\mathcal{M}}(y_t|\underbrace{x_1, y_1, ..., x_{C\times N}, y_{C\times N}}_{demonstrations}, x_t), \quad (3)$$

where $\ddot{\mathcal{M}}$ is decoding strategies of LLMs. In the context of general ICL tasks, it is commonly assumed that the demonstrations and the test sample x_t originate from data that follows the same distribution $\mathcal{D}_T$, but in OOD scenario, the demonstrations come from different distribution $\mathcal{D}_S$.

3.3 Initial Samples Selection

Pseudo-labeling is one of the most popular semi-supervised learning strategies, which utilizes the model itself to obtain labels of unlabeled data (Lee et al., 2013). However, when pseudo-labels are inaccurate, confirmation bias can lead to degradation in the performance of models trained with those pseudo-labels (Arazo et al., 2020). Especially when considering the challenge set, where no labels are available, this further exacerbates the difficulty of generating accurate pseudo-labels. To improve the quality of pseudo-labels, we propose a comprehensive quantitative index based on model confidence and perturbation perplexity.

3.3.1 Confidence

For pseudo-label selection tasks, confidence is a commonly used metric to filter out high-quality annotation results (Rizve et al., 2021). Therefore, we first train a lightweight model as confidence selector on $\mathcal{D}_S$ by optimizing cross entropy and use it to assign rough labels to the unlabeled $\mathcal{D}_T$. There are different representations of confidence in different tasks.

Classification Tasks. The confidence of a sample $x_t \in \mathcal{D}_T$ can be expressed as:

$$Conf_t = \max_{y'} p(y'|x_t), \quad (4)$$

where p is the probability distribution predicted by the trained model, y' is is the pseudo-label obtained according to the confidence.

Generation tasks. For other more complex non-sentence level tasks such as Named Entity Recognition, we view them a tokens generation task. For any token t in the output, a similar method as Eq. 4 is used to generate its confidence, then the overall confidence can be computed as the product of individual token confidences:

$$Conf_t = \prod_{i}^{k} \max_{t} p\big(t|\mathcal{P}(x_t)\big), \quad (5)$$

where $\mathcal{P}(x_t)$ denotes the sequence of predicted tokens for the model, k is the output sequence length.

3.3.2 Perturbation Perplexity

In the OOD bias scenario, a model that is well-fitted to the source domain struggles to generalize to the target domain. Even when selection is based on confidence, labeling errors and noise remain unavoidable. To alleviate this, we propose a perturbation perplexity method to further refine the confidence-based results. Perplexity measures LLMs' uncertainty about the data, with lower values indicating more confident predictions (Gao et al., 2024). Formally, the perplexity of model $\mathcal{M}$ can be written as:

$$PPL(x_t) = exp\big(-\frac{1}{T}\sum_{i=1}^{T} logP(t_i|t_{i-1})\big), \quad (6)$$

where T the number of words in x_t and $P(t_i|t_{i-1})$ is the prediction probability of token t_i by given t_{i-1}. Subsequently, we use the confidence-based prediction results as the sole perturbation factor, constructing a nearly identical prompt pair $\{\mathcal{I}(x_t), \hat{\mathcal{I}}(x_t)\}$ shown in Figure 2, and measure the change of perplexity to the perturbation:

$$\Delta_{PPL_t} = \big|PPL(\{\mathcal{I}(x_t)) - PPL(\hat{\mathcal{I}}(x_t))\big|. \quad (7)$$

If the change Δ_{PPL} caused by the perturbation is small, it indicates that the model is not sensitive to the pseudo-label, making it more likely that the pseudo-label is a reasonable annotation. Due to space constraints, we provide only a simple example in Figure 2 to illustrate how perturbations are applied. Specific forms of perturbations for different tasks are detailed in Appendix D.4.

3.3.3 Sample Selection based on CPP

Furthermore, we comprehensively consider both confidence and perturbation perplexity to obtain a more informed quantification of pseudo-label quality. Since perturbation perplexity may be much

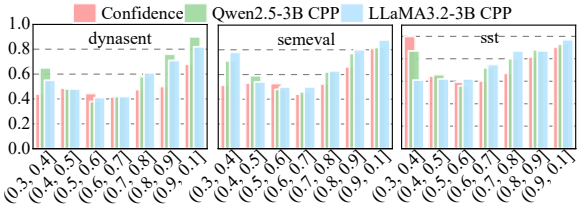

Figure 3: Accuracy under different confidence and CPP intervals on SA with `Qwen2.5-3B` and `LLaMA3.2-3B`.

larger than confidence, we smooth and normalize it, and use the ratio of the two as the final metric:

$$CPP_t = Conf_t \cdot \left(\frac{\Delta_{PPL_t}}{max(\Delta_{PPL})}\right)^{-0.5}, \quad (8)$$

where Δ_{PPL} denote the set of perplexity differences of all samples. Based on the CPP score, samples larger than τ will be are filtered as the initial sample set for the iteration. The initial sample set is represented as $X^{(0)}$.

3.3.4 CPP Screening Results

Figure 3 provides supporting evidence for the validity of CPP, which shows the accuracy of selected samples with different threshold of confidence and CPP on two different NLP tasks. A common trend for confidence and CPP that holds true in most cases is that as the interval threshold gradually increases, the accuracy of predictions within the corresponding region also improves. This suggests selecting samples falling within larger intervals is reasonable. Additionally, we observe that the accuracy of samples within larger intervals corresponding to CPP is higher than that of samples selected solely based on confidence. This indicates that incorporating perplexity can further refine the quality of samples in larger intervals. Further support for the validity of CPP can be found in Section 4.2.3 and Appendix C.

3.4 Iterative Pseudo-label Generation

While the initial pseudo-label selector based on CPP may produce results with high accuracy, considering the specificity of different downstream LLMs, these labels may not be the most suitable for the LLMs. Therefore, we aspire for LLMs to be more involved in the generation of pseudo-labels and iteratively update the pseudo-label set (Cascante-Bonilla et al., 2021). The description of the iterative algorithm begins with demonstrations search.

Demonstrations Search. We search for demonstrations within the selected $X^{(0)}$ because they come from the same distribution as the samples to be predicted. To achieve this, rather than randomly selecting samples, we can employ some widely considered sample selection methods that have proven to be effective. In practical operations, to reduce the time required for text vectorization and search, LPL performs tf-idf vectorization on text data and utilize cosine similarity as the metric for retrieval. For each sample x_t and each category C, N samples with the highest similarity from $X^{(0)}$ are selected. During the process, we need to ensure that x_t is not selected to prevent label leakage if $x_t \in X^{(0)}$.

LLMs Pseudo-label Generation. Based on the searched demonstrations, LLMs determines the confidence of the sample prediction in a similar way to the initial sample selection. Unlike the DeBERTa labeler, the prediction results of LLMs are presented in the form of tokens. For this reason, we select the first token t generate by the LLMs for classification tasks and calculate the confidence score of t as:

$$\hat{Conf}_t = \max_t p_{\mathcal{M}}(t|x_1, y_1, ..., x_{C\times N}, y_{C\times N}, x_t). \quad (9)$$

When multiple tokens are involved, the confidence is the multiplication of the confidence of each token. Subsequently, we use the same method as shown in Eq. 8 to calculate the CPP score $\hat{CPP}_t$ derived from LLMs during the iterative process. Predicted samples with high $\hat{CPP}_t$ are added to the initial sample set:

$$X^{(j+1)} = \{\forall x_t \in X | \hat{CPP}_t > \gamma \wedge x_t \notin X^{(j)}\} \cup X^{(j)}, \quad (10)$$

where γ is a new threshold that is more suitable for LLMs, j indicates the current number of iterations. Each iteration is a response to the result of the previous iteration, thereby prompting LLMs to consider more samples with high confidence. Furthermore, more reasonable samples can prompt LLMs to make better predictions, further reinforcing the pseudo-labels.

3.5 Majority Voting

Although consistently selecting samples with high CPP scores seems to contribute to more robust predictions for LLMs, LLMs are sensitive to changes in the demonstrations, which may lead to significant differences in predictions across different iterative processes (Song et al., 2024b). To ensure that the final predictions remain stable across multiple

Tasks	Source datasets	Target datasets		
SA	amazon	dynasent	semeval	sst
TD	civil_comments	adv_civil	implicit_hate	toxigen
NLI	mnli	anli	contrac_nli	wanli
NER	FewNerd	conll	ener	wnut

Table 1: Experimental tasks and corresponding datasets.

iterations, we employ the Majority Voting method to mitigate potential performance degradation during the iterations (Xie et al., 2023). Specifically, for each iteration, we get the corresponding predicted labels from LLMs. Then, we take the prediction that appears most frequently among the J iterations for each sample as the final result. In cases where multiple predictions have the same frequency, we randomly select one label as the final result. The iterative process of LPL described above is illustrated in Algorithm 1.

4 Experiments

4.1 Datasets and Baselines

Datasets. In this paper, we adopt the four sub-tasks under the challenging distribution shift benchmark BOSS (Yuan et al., 2023) to build experiments, including Sentiment Classification (SA), Toxicity Detection (TD), Natural Language Inference (NLI) and Name Entity Recognition (NER) as shown in Table 1. The target datasets all have low SimCSE scores with source datasets (Gao et al., 2021), indicating strong OOD bias. Among them, the first three can be regarded as classification tasks, while the last one is generation tasks. A brief description of the tasks and datasets is in Appendix D.1.

Baselines and LLMs. To verify the universality of LPL, we conduct experiments on a number of different LLMs, including `GPT2-xl`[2] (Radford et al., 2019), `Qwen2.5-3B`[3] (qwe, 2024), `LLaMA3.1-8B` and `LLaMA3.2-3B`[4] (AI@Meta, 2024), and `Mistral-7B-v0.3`[5]. In addition, we also perform LPL on `GPT-3.5-turbo`[6] and `GPT-4o-mini`[7] to verify the scalability on non-local LLMs. For each LLM, we compare different sample baselines including **Random**, **Bm25** (Luo et al., 2023), **KNN** (Liu et al., 2022), **TopK+MDL** (Wu et al., 2023), **DrICL** (Luo et al., 2023), **TopK+ConE** (Peng et al., 2024b), and **DICL** (Kapuriya et al., 2025). In addition, we also report the generalization performance of DeBERTa/T5 fine-tuned on the source dataset, which we refer to as DeBERTa/T5 fine-tuning. A brief description of the baselines is in Appendix D.3.

[2]https://huggingface.co/openai-community/gpt2-xl
[3]https://huggingface.co/Qwen
[4]https://huggingface.co/meta-llama
[5]https://huggingface.co/mistralai/`Mistral-7B-v0.3`
[6]https://platform.openai.com/docs/models/gpt-3-5-turbo
[7]https://platform.openai.com/docs/models/#o4-mini

4.2 Experimental Results and Analysis

4.2.1 Main Experimental Results

In Table 2 and Table 3, we present the comparative results of the proposed method and the baseline models on different datasets and have the following observations.

Under OOD bias, semantic similarity-based retrieval methods do not consistently outperform random sampling. For instance, on SA with `Mistral-7B-v0.3` and `GPT2-xl`, random selection occasionally surpasses bm25, KNN, and TopK+MDL. This suggests that substantial OOD bias makes beneficial source-domain sample selection challenging, requiring more context-aware retrieval strategies. The issue is even more pronounced in generative tasks, whose higher complexity and lack of explicit labels demand deeper semantic understanding and make OOD demonstration quality particularly critical.

While LLMs tend to achieve better overall performance compared to fine-tuning approaches, they remain evidently susceptible to OOD bias. Different LLMs exhibit performance variations across different tasks. For simple classification tasks, LPL outperforms the fine-tuned DeBERTa in most cases except for the relatively small `GPT2-xl`, demonstrating the effectiveness of the ICL paradigm. For the more challenging NER task, only LLaMA-3.1-8B achieved an improvement in average performance. Moreover, although LPL generally demonstrates a performance advantage, the baseline methods under the corresponding LLMs may underperform. For example, `LLaMA3.2-3B` performs poorly on the SA task, and Mistral7B-v0.3 shows weaker results on the NLI task. This indicates the widespread impact of OOD bias across different models and tasks.

The proposed LPL is capable of fully exploiting in-domain information from the target domain, yielding substantial performance gains despite the presence of label noise. We observe that the proposed LPL achieves an average performance improvement over the best baseline across all LLMs and tasks, with gains exceeding 6% in the best case (`GPT2-xl` on SA). This supports the

Methods	SA				TD				NLI			
	dynasent	semeval	sst	Avg.	adv	implicit	toxigen	Avg.	anli	contract	wanli	Avg.
DeBERTa fine-tuning	49.84	57.52	75.73	61.03	40.46	61.24	62.45	54.72	35.38	35.72	62.70	44.60
Qwen2.5-3B												
random	69.07	53.68	65.70	62.82	81.41	41.80	51.70	58.30	40.56	45.43	48.98	44.99
bm25	67.29	53.78	66.64	62.57	81.29	41.02	51.17	57.83	40.88	46.53	48.68	45.36
KNN	68.80	55.26	66.73	63.60	81.89	41.16	52.34	58.46	40.63	44.74	47.10	44.16
TopK+MDL	68.54	54.63	67.21	63.46	81.21	42.34	53.63	59.06	41.89	44.67	48.92	45.16
DrICL	67.33	53.78	68.34	63.15	81.92	42.42	52.51	58.95	42.17	45.18	48.36	45.24
TopK+ConE	69.47	54.60	66.93	63.67	81.56	42.85	52.57	58.99	43.19	45.60	48.38	45.72
DICL	66.99	54.02	65.14	62.05	81.53	41.32	51.00	57.95	43.31	45.84	48.18	46.11
LPL*	71.16	55.70	70.38	**65.75**	82.16	43.02	57.66	**60.95**	47.78	47.05	52.38	**49.07**
LLaMA3.1-8B												
random 69.56	64.18	78.06	70.93	80.68	55.00	78.09	71.26	44.09	46.05	52.00	47.38	
bm25	66.25	63.32	75.16	68.24	77.76	56.82	79.36	71.31	42.88	45.29	51.60	46.59
KNN	69.12	63.70	75.54	69.45	77.16	59.86	80.87	72.63	43.54	45.58	51.98	47.03
TopK+MDL	70.87	64.86	78.15	71.29	78.80	60.19	80.23	73.10	44.68	45.89	51.58	47.38
DrICL	68.65	63.42	76.17	69.41	77.31	59.44	80.84	72.53	47.06	45.25	52.56	48.29
TopK+ConE	71.66	63.12	77.08	70.62	79.52	61.04	80.76	73.77	45.65	44.86	53.22	47.91
DICL	68.56	63.70	75.82	69.36	77.85	58.56	80.15	72.19	42.90	45.78	51.66	46.78
LPL*	75.37	65.08	78.79	**73.08**	80.84	65.24	82.66	**76.25**	49.93	46.27	56.66	**50.95**
LLaMA3.2-3B												
random	54.68	45.10	67.29	55.69	56.01	56.58	65.43	59.34	33.50	37.45	41.84	37.60
bm25	53.61	47.52	65.60	55.58	60.51	55.68	68.19	61.46	34.16	39.41	41.34	38.30
KNN	58.19	47.30	68.17	57.89	56.13	59.22	70.11	61.82	35.66	39.17	41.45	38.76
TopK+MDL	59.04	48.12	68.61	58.59	57.56	60.03	69.98	62.80	35.73	40.11	41.60	39.15
DrICL	58.12	46.70	67.82	57.55	59.60	69.37	63.44	64.00	34.58	41.08	42.30	39.32
TopK+ConE	58.40	47.88	69.25	58.51	61.34	59.60	69.37	63.44	36.05	40.56	41.84	39.48
DICL	58.80	46.29	66.17	57.09	59.72	56.06	68.14	61.31	35.38	39.42	41.07	38.62
LPL*	62.25	51.80	72.18	**62.08**	61.76	63.98	73.80	**66.51**	35.43	42.17	43.90	**40.50**
Mistral-7B-v0.3												
random	62.99	51.74	70.67	61.80	42.52	64.28	73.51	60.10	35.81	41.14	46.68	41.21
bm25	60.23	53.36	68.69	60.76	46.94	64.14	72.65	61.24	37.84	45.77	48.02	43.88
KNN	62.89	51.47	70.56	61.64	42.53	63.48	73.05	59.69	35.88	44.56	48.36	42.93
TopK+MDL	62.16	52.78	70.23	61.72	44.50	64.65	72.82	60.66	38.65	44.38	49.43	44.15
DrICL	61.84	51.23	70.58	61.22	44.68	64.80	72.57	60.68	37.56	44.29	47.16	43.00
TopK+ConE	62.76	52.91	70.53	62.07	46.71	64.34	72.79	61.28	39.71	45.20	48.62	44.51
DICL	62.66	52.07	69.38	61.37	43.22	63.53	72.60	59.78	35.76	44.18	47.65	42.53
LPL*	65.35	54.04	72.11	**63.80**	46.95	66.98	75.81	**63.25**	42.75	44.15	51.58	**46.16**
GPT2-xl												
random	39.54	50.32	53.80	47.89	27.58	58.90	57.55	48.01	–	–	–	–
bm25	40.35	48.30	53.51	47.39	33.97	57.06	56.38	49.14	–	–	–	–
KNN	40.28	50.22	47.61	46.04	26.59	58.16	56.06	46.94	–	–	–	–
TopK+MDL	40.78	50.62	48.90	46.77	30.46	58.88	57.70	49.01	–	–	–	–
DrICL	38.93	47.91	49.05	45.30	31.13	59.94	59.32	50.13	–	–	–	–
TopK+ConE	36.61	49.86	53.14	46.54	27.38	59.54	58.10	48.34	–	–	–	–
DICL	39.88	49.16	48.82	45.95	27.50	57.94	56.69	47.38	–	–	–	–
LPL*	42.23	55.24	64.29	**53.92**	31.13	59.94	59.32	**50.13**	–	–	–	–

Table 2: Comparison of experimental results across different LLMs and baseline methods. **Boldface** denotes the best average result. * indicates statistically significant improvement over all baselines under the one-sided Wilcoxon signed-rank test ($p < 0.01$). '–' indicates that the input length exceeds the model's maximum limit.

hypothesis in Section 3.1, which suggests that when label noise is relatively low, using noisy in-distribution samples to construct demonstrations can effectively raise the performance upper bound of the model. But we also acknowledge that in cases where the reliability of pseudo-labels is insufficient, such as in adversarial datasets like adv_civil and anli, LPL sometimes yields only marginal performance improvements. This may indicate a key direction for future enhancements.

4.2.2 Results on Non-local LLMs

Given that confidence and perplexity are difficult to obtain from non-locally deployed LLMs, especially black-box models with undisclosed parameters, we employ a lightweight proxy LLM to select samples during the iterative process. Specifically, for each input to the black-box model, we feed the same prompt into a lightweight, locally deployed LLM (in our case, `LLaMA3.2-3B`), and use the CPP value computed by this lightweight model as the criterion for selecting samples during the iterative process. To ensure prediction stability, only those samples whose predicted labels align with the outputs of the black-box model are incorporated into $X^{(j)}$. The experimental results of two different non-locally deployed LLMs are shown in Table 4. LPL consistently outperforms KNN and DICL across all tasks on both non-locally deployed LLMs (`GPT-3.5-turbo` and `GPT-4o-mini`), it even achieves over a 4% performance gain on NLI and NER tasks with the large-scale `GPT-3.5-turbo`, which indicates that LPL can be broadly applied across different LLMs.

Methods	conll	ener	wnut	Avg.
T5 fine-tuning	61.72	45.05	46.46	51.08
Qwen2.5-3B				
random	39.17	39.18	39.02	39.12
bm25	38.29	38.93	38.24	38.49
KNN	37.48	38.08	36.78	37.45
TopK+MDL	39.65	37.01	40.83	39.16
DrICL	38.94	36.72	37.50	37.72
TopK+ConE	38.75	39.92	39.60	39.42
DICL	38.54	39.17	37.75	38.49
LPL*	41.91	41.89	39.53	**41.11**
LLaMA3.2-3B				
random	30.18	30.43	31.00	30.54
bm25	28.30	28.25	29.69	28.75
KNN	29.29	26.52	28.70	28.17
TopK+MDL	29.72	28.63	29.01	29.12
DrICL	31.86	29.33	28.04	29.74
TopK+ConE	28.85	27.56	30.58	29.00
DICL	30.41	27.82	28.18	28.80
LPL*	36.75	31.56	31.94	**33.42**
LLaMA3.1-8B				
random	50.39	50.93	51.20	50.84
bm25	50.24	49.81	49.97	50.01
KNN	51.61	49.22	49.79	50.21
TopK+MDL	50.86	50.17	50.89	50.64
DrICL	51.03	49.17	50.12	50.11
TopK+ConE	51.23	50.05	51.42	50.90
DICL	51.19	49.63	50.06	50.29
LPL*	54.43	49.89	52.34	**52.22**
Mistral-7B-v0.3				
random	36.96	37.64	37.76	37.45
bm25	35.40	36.26	35.91	35.86
KNN	32.98	35.67	34.55	34.40
TopK+MDL	33.60	37.77	37.61	36.33
DrICL	33.57	36.50	35.72	35.26
TopK+ConE	35.73	36.30	36.86	36.30
DICL	32.57	35.65	35.31	34.51
LPL*	39.15	38.85	41.22	**39.74**
GPT2-xl				
random	16.89	16.68	17.10	16.89
bm25	14.47	12.87	15.40	14.25
KNN	18.67	11.15	13.60	14.47
TopK+MDL	15.56	14.54	16.78	15.63
DrICL	17.65	15.68	17.22	16.85
TopK+ConE	15.91	15.36	17.40	16.22
DICL	18.36	12.92	13.08	14.79
LPL*	21.03	19.82	18.33	**19.73**

Table 3: Results on NER tasks.

4.2.3 Ablation Study

An ablation study is conducted to compare two variants of the proposed LPL framework, namely LPL$_{-PPL}$ and LPL$_{-Conf}$. We do not explore ablation studies related to majority voting in this section, as a more detailed investigation is presented in a later section 4.2.5. In LPL$_{-PPL}$, sample selection is performed using model confidence alone, excluding perplexity. Conversely, LPL$_{-Conf}$ uses perplexity for selection while ignoring model confidence. As shown in Table 5, removing either confidence (LPL$_{-Conf}$) or perplexity (LPL$_{-PPL}$) leads to a consistent performance drop across all tasks and models, confirming the effectiveness of combining both signals in the LPL framework. Notably, removing confidence results in a larger decline, indicating that confidence plays a more critical role on them, while perplexity offers complementary benefits. In addition, we observe that in certain cases, such as the TD task, the performance drop caused by LPL$_{-PPL}$ is also non-negligible, indicating that perplexity is indispensable in the process of high-quality sample selection.

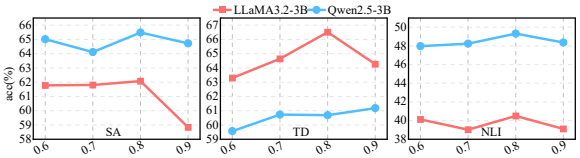

Figure 4: The changes in average model performance across different tasks.

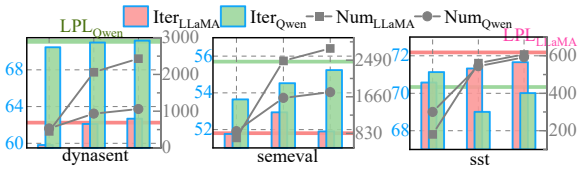

Figure 5: Model performance and sample space evolution with iterations on SA (left: accuracy(%), right: sample count). The horizontal lines indicate the LPL majority vote results under the corresponding LLMs.

4.2.4 Hyperparameter Settings

The most critical parameters in LPL are the filtering thresholds τ and γ used during the label generation process. Determining their optimal values typically requires a grid search, and the best settings may vary across tasks and datasets. To reduce the search space and facilitate practical use of LPL, we fix both τ and γ at 0.8, as illustrated in Figure 4. We observe that, except for `Qwen2.5-3B` on the TD task, the optimal performance is consistently achieved around this value. This aligns with our intuition: a moderate threshold effectively balances a sufficiently large sample search space with low pseudo-label noise.

4.2.5 Effect of Iterations and Majority Voting

To demonstrate the effectiveness of iteration and majority voting, we present Figure 5, which illustrates the changes in the number of selected samples and the performance of LPL on SA. While variations exist across datasets and LLMs, the number of selected samples generally increases with more iterations, indicating that iterative selection helps explore a broader demonstration space. However, the performance of LPL does not always improve consistently with more iterations. This is attributed to the inherent sensitivity of LLMs to demonstrations—samples in $X^{(j)}$ may not be equally suitable for all models and tasks. Consequently, majority

Methods	SA				TD				NLI				NER			
	dynasent	semeval	sst	Avg.	adv	implicit	toxigen	Avg.	anli	contr	wanli	Avg.	conll	ener	wnut	Avg.
GPT3.5-turbo																
KNN	69.80	67.20	77.20	71.40	81.80	56.00	84.80	74.20	53.80	41.20	56.60	50.53	54.13	35.18	44.71	44.67
DICL	70.50	66.80	75.40	70.90	82.50	58.00	84.80	75.10	54.00	40.00	55.40	49.80	55.00	35.50	46.00	45.50
LPL	76.40	68.20	79.40	**74.60**	80.72	60.20	85.00	**75.30**	58.80	41.60	64.00	**54.80**	60.05	37.97	48.58	**48.87**
GPT4o-mini																
KNN	68.40	71.80	77.40	72.53	68.40	55.60	84.60	69.53	53.60	38.00	55.00	48.87	51.90	40.57	45.04	45.84
DICL	67.50	72.20	76.00	71.90	69.20	56.50	83.50	69.73	52.00	38.80	54.20	48.33	53.00	41.00	44.50	46.17
LPL	69.60	75.80	79.20	**74.87**	69.60	59.40	84.80	**71.23**	52.70	41.40	62.00	**52.03**	55.21	42.55	48.21	**48.66**

Table 4: Experimental results of non-locally deployed LLMs across all tasks.

Variants	SA	TD	NLI	NER
Qwen2.5-3B				
LPL$_{-PPL}$	65.01	59.57	47.98	40.32
LPL$_{-Conf}$	61.72	58.74	47.56	39.04
LPL	**65.75**	**60.95**	**49.07**	**41.11**
LLaMA3.2-3B				
LPL$_{-PPL}$	61.78	63.30	40.13	32.78
LPL$_{-Conf}$	57.70	58.24	37.87	30.68
LPL	**62.08**	**66.51**	**40.50**	**33.42**

Table 5: The ablation results of `Qwen2.5-3B` and `LLaMA3.2-3B`. For simplicity, we only present the average performance across three datasets per task.

voting plays a crucial role by aggregating predictions from different iterations to produce more stable results and mitigate the impact of outliers. In some cases, it even outperforms any single iteration, as observed with `Qwen2.5-3B` on Semeval and `LLaMA3.2-3B` on sst and toxigen. More results can be found in Appendix F.

4.2.6 Visualization of Selected Samples

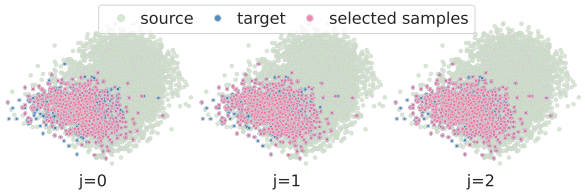

Figure 6: 2D visualization of sst.

To further illustrate the changes in selected samples during the iterations, we choose sst, and perform 2D visualization using t-SNE based on their representations obtained from SimCSE as the iterations progress. In Figure 6, the green dots (from the source domain) and the blue dots (from the target domain) exhibit a clear distributional discrepancy, which leads to the negative impact of OOD bias on ICL. When the demonstration samples are replaced with those selected based on CPP (pink), the distributional gap between the demonstrations and the target domain is reduced. As the iterations progress, the selected samples gradually expand their coverage, which facilitates ICL within a distribution space closer to the target. The visualization results further validate the core contribution of LPL.

5 Conclusion

This paper proposes LPL, an iterative target-domain pseudo-labeling framework for mitigating OOD bias in ICL by replacing OOD samples with refined pseudo-labeled instances, where theoretical analysis and extensive experiments across multiple NLP tasks and LLMs demonstrate its effectiveness and robustness.

Limitations

Although effective, LPL requires multiple iterations to achieve stable results, leading to additional computational overhead. Moreover, LPL still needs to be evaluated on a broader range of LLMs to further verify its generalization capability. For closed-source models, LPL relies on proxy models to generate pseudo-labels, which may introduce potential errors. Finally, the performance gains of LPL on adversarial datasets remain limited, as the initial pseudo-label generation strategy is less effective in adversarial settings and may negatively affect subsequent iterations. Therefore, developing more robust pseudo-label initialization strategies remains an important direction for future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC): “Research on Understanding Ancient Characters Based on Multimodal Large Models” (Grant No. 62476111), China Postdoctoral Science Foundation Funded Project (Grant No. 2024M761122), the Scientific Research Project of the Education Department of Jilin Province (Grant No. JJKH20261299KJ) and the Industry University Research Innovation Fund of the Ministry of Education project (Grant No. 2022XF017).

References

2024. Qwen2 technical report.

AI@Meta. 2024. Llama 3 model card.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–8.

Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10687–10701.

Hadi Askari, Shivanshu Gupta, Terry Tong, Fei Wang, Anshuman Chhabra, and Muhao Chen. 2025. Unraveling indirect in-context learning using influence functions. *CoRR*, abs/2501.01473.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, 2021*, pages 6912–6920.

Yen-Chi Chen. 2017. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019.

Yuyang Ding, Dan Qiao, Juntao Li, Jiajie Xu, Pingfu Chao, Xiaofang Zhou, and Min Zhang. 2025. Towards ds-ner: Unveiling and addressing latent noise in distant annotations. *IEEE Transactions on Knowledge and Data Engineering*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1107–1128.

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

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14014–14031.

Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. 2024. On the noise robustness of in-context learning for text generation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10136–10148.

Shuoran Jiang, Qingcai Chen, Yang Xiang, Youcheng Pan, and Yukang Lin. 2024. Linguistic rule induction improves adversarial and OOD robustness in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10565–10577.

Janak Kapuriya, Manit Kaushik, Debasis Ganguly, and Sumit Bhatia. 2025. Exploring the role of diversity in example selection for in-context learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 2962–2966.

Dong-Hyun Lee and 1 others. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta.

Ming Li, Qingli Li, and Yan Wang. 2023. Class balanced adaptive pseudo labeling for federated semi-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 16292–16301.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6219–6235.

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: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022*, pages 100–114.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Seyed Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y. Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *CoRR*, abs/2305.14128.

Manas Madine. 2024. Bridging distribution gap via semantic rewriting with llms to enhance OOD robustness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Student Research Workshop, Bangkok, Thailand, August 11-16, 2024*, pages 458–468.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *CoRR*, abs/2302.11042.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9844–9855.

Kyle O'Brien, Nathan Ng, Isha Puri, Jorge Mendez, Hamid Palangi, Yoon Kim, Marzyeh Ghassemi, and Thomas Hartvigsen. 2024. Improving black-box robustness with in-context rewriting. *CoRR*, abs/2402.08225.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024a. Revisiting demonstration selection strategies in in-context learning. *CoRR*, abs/2401.12087.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024b. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9090–9101. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *CoRR*, abs/2310.09881.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Vinay M. S., Minh-Hao Van, and Xintao Wu. 2024. In-context learning demonstration selection via influence analysis. *CoRR*, abs/2402.11750.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. 2018. Transductive semi-supervised deep learning using min-max features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, volume 11209, pages 311–327.

Rui Song, Fausto Giunchiglia, Yingji Li, Mingjie Tian, and Hao Xu. 2024a. TACIT: A target-agnostic feature disentanglement framework for cross-domain text classification. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18999–19007.

Rui Song, Yingji Li, Lida Shi, Fausto Giunchiglia, and Hao Xu. 2024b. Shortcut learning in in-context learning: A survey. *arXiv preprint arXiv:2411.02018*.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 20841–20855. PMLR.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4645–4657.

Guo-Hua Wang and Jianxin Wu. 2020. Repetitive reprediction deep decipher for semi-supervised learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6170–6177.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. 2024. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *IEEE Data Eng. Bull.*, 48(1):48–62.

Hui Wu and Xiaodong Shi. 2022. Adversarial soft prompt tuning for cross-domain sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2438–2447.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1423–1436.

Ming-Kun Xie, Jiahao Xiao, Hao-Zhe Liu, Gang Niu, Masashi Sugiyama, and Sheng-Jun Huang. 2024. Class-distribution-aware pseudo-labeling for semi-supervised multi-label learning. *Advances in Neural Information Processing Systems*, 36.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot NER with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7935–7956.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering*.

Shangqing Xu and Chao Zhang. 2024. Misconfidence-based demonstration selection for LLM in-context learning. *CoRR*, abs/2401.06301.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings*, pages 189–196.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.

Lang Yu, Qin Chen, Jiaju Lin, and Liang He. 2023. Black-box prompt tuning for vision-language model as a service. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 1686–1694. ijcai.org.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in NLP: benchmarks, analysis, and llms evaluations. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 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 *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139, pages 12697–12706. PMLR.

A Algorithm Description

B Theoretical Proof

Lemma 2. *Following the definition in Thm 1, if noise with probability p is introduced to the samples, with probability at least $1-\delta, 0<\delta<1$, the following inequality holds:*

$$R(f)_{noise} \leq p + (1-2p)(\hat{R}(f) + \epsilon). \quad (11)$$

Proof. Given labeled samples $(X, Y) = \{(x_1, y_1), ...(x_N, y_N)\}$, the risk of the model is $R(f) = Pr(f(x) \neq y)$. When each sample label is wrong with probability p, for any function f, there are two possible scenarios:

$$\begin{cases} Pr(f(x_i) \neq y_i \wedge y_i = y_i') \\ = Pr(f(x_i) \neq y_i) Pr(y_i = y_i') \\ = R(f)(1-p) \\ Pr(f(x_i) = y_i \wedge y_i \neq y_i') \\ = Pr(f(x_i) = y_i) Pr(y_i \neq y_i') \\ = (1 - R(f))p. \end{cases} \quad (12)$$

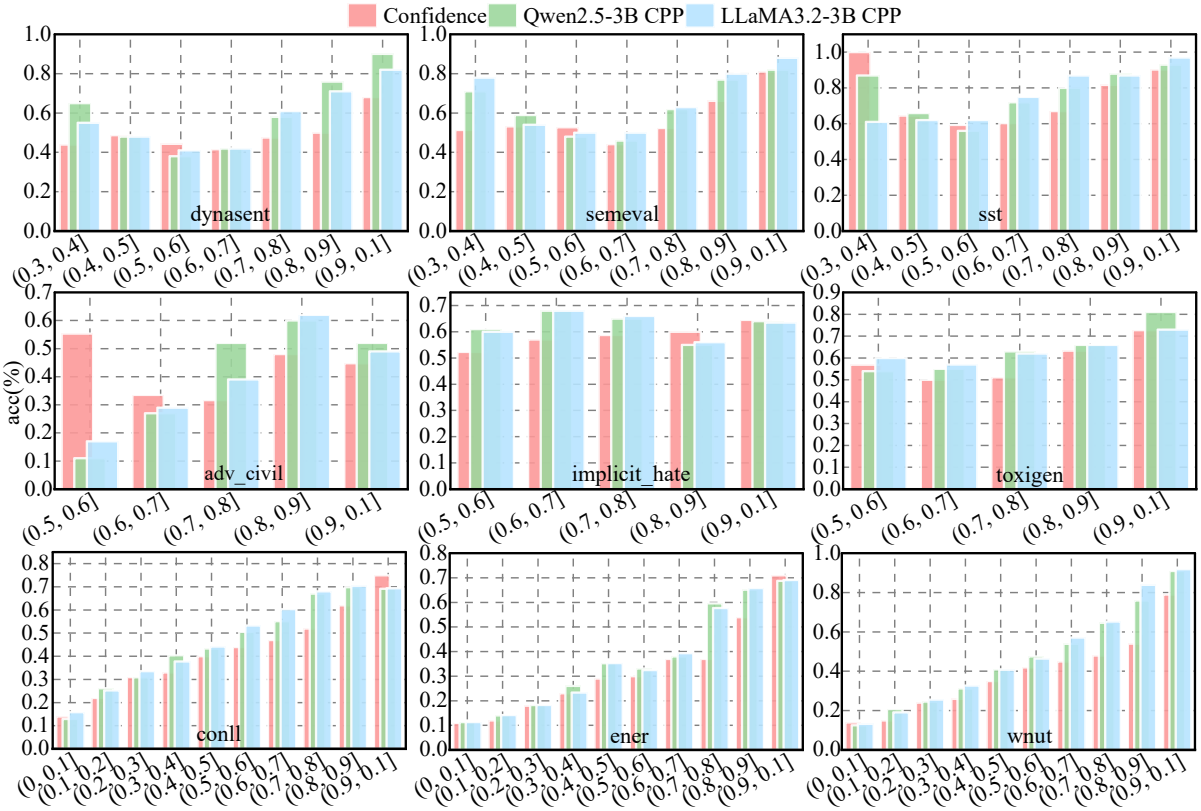

Figure 7: Accuracy under different confidence and CPP intervals on SA, TD, and NER tasks with `Qwen2.5-3B` and `LLaMA3.2-3B`.

Therefore, $R(f)_{noise}$ is the sum of the above two cases:

$$R(f)_{noise} = p + (1-2p)R(f). \quad (13)$$

By plugging Thm 1 into it, we obtain:

$$R(f)_{noise} \leq p + (1-2p)(\hat{R}(f) + \epsilon). \quad (14)$$

The above lemma is proved. □

Corollary 2. *Under relaxed conditions, if the probability of label noise is as low as possible, then in the presence of strong OOD bias, constructing demonstrations using target domain data with label noise will have a smaller negative impact.*

Proof. According to (Ben-David et al., 2010), for source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$, the error upper bound can be expressed as:

$$R(f)_S \leq R(f)_T + d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \mathcal{C}, \quad (15)$$

where $R(f)_S$ and $R(f)_T$ denote the training error under different data distributions, respectively. $d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$ represents the distance between the training distribution $\mathcal{D}_S$ and the test distribution $\mathcal{D}_T$, and $\mathcal{C}$ denotes the inevitable error difference between the training distribution and the test distribution under an optimal classifier.

According to Thm 1, we can rewrite $R(f)_S$ to:

$$R(f)_S \leq \hat{R}(f) + \epsilon + d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \mathcal{C}, \quad (16)$$

where we omit the lower corner T for uniform presentation. Then, the relationship between the noise and the OOD bias error upper bound is given by:

$$\begin{aligned}\Delta &= \hat{R}(f) + \epsilon + d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \mathcal{C} \\ &\quad - p - (1-2p)(\hat{R}(f) + \epsilon) \\ &= d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \mathcal{C} + p(2\hat{R}(f) + 2\epsilon - 1)\end{aligned} \quad (17)$$

When the probability $1-\delta$ is high (usually more than 0.9), $\delta < 0.1$, leading to $\epsilon > \sqrt{\log \frac{1}{\delta}} > 1$, so $p(2\hat{R}(f) + 2\epsilon - 1) > 0$. For $\mathcal{D}_S$ and $\mathcal{D}_T$ which has large OOD bias, $d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \mathcal{C} > 0$. Therefore, $\Delta > 0$, which shows that $R(f)_{noise}$ possesses a tighter error upper bound. In this case, introducing label noise instead of OOD bias is more promising for improving the model's performance. □

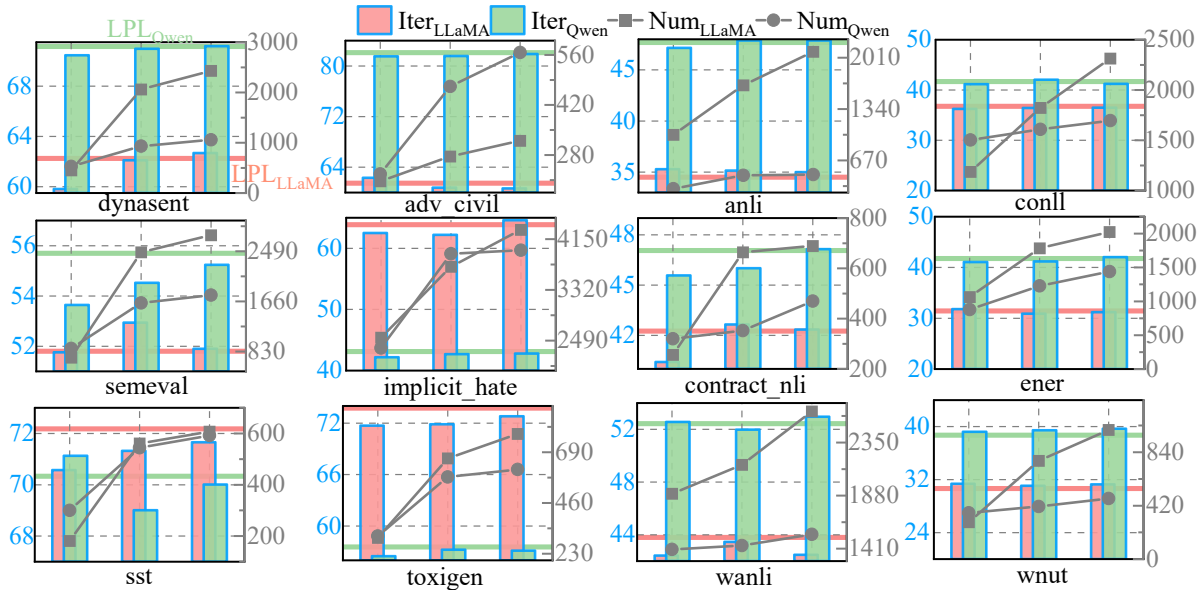

Figure 8: Model performance and sample space evolution with iterations across datasets (left: accuracy(%), right: sample count). The horizontal lines indicate the LPL majority vote results under the corresponding LLMs.

C Comparison of CPP on More Datasets

We present more results in Figure 7 to demonstrate the effect of CPP on TD and NER tasks. Except for a few cases such as CoNLL, CPP consistently achieves performance comparable to or even better than confidence within high-confidence intervals, indicating that the use of CPP indeed helps improve the quality of selected samples.

D More Experimental Details

D.1 Datasets

We have detailed the tasks and datasets used in the experiment as follows.

- SA is described as a three-classification task, including positive, negative, and neutral. *Amazon* dataset contains reviews of products from 29 different categories. *Dynasent* is a dataset created using a human-in-the-loop annotation method to generate adversarial sentence sets. *Semeval* is a sentiment three-classification dataset sourced from Twitter, and *sst* is based on sentence-level movie reviews.

- TD is described as a sentence-level binary classification task, including two categories: toxic and benign. *Civil_comments* contains public comments from a civil commentary platform, involving different user groups and subcategories of toxicity. *Adv_civil* is generated from Civil Comments through text-based adversarial attacks. *Implicit_hate* includes both explicit and implicit forms of toxic tweets, where implicit toxicity does not contain common or toxicity-related keywords, making it more challenging to identify. *Toxigen* is composed of implicit toxic texts generated by demonstration-induced GPT-3.

- NLI involves determining the relationship between a premise and a hypothesis, specifically whether the hypothesis is entailed by, contradicts, or is neutral with respect to the premise. *Mnli* includes ten different types of written and spoken sentence pairs, featuring various styles, topics, and levels of formality. *Anli* is an adversarial dataset collected in a "human-in-the-loop" manner, with each premise primarily sourced from Wikipedia and hypotheses generated by human adversaries. *Contract_nli* treats each contract as a premise and holds a fixed set of hypotheses across the entire dataset. *Wanli* is synthesized by GPT-3, with each example containing challenging patterns identified in *mnli*.

- NER involves identifying and classifying entities within text into predefined categories such as person names, locations, organizations, etc. *FewNerd* is arguably the largest NER dataset, labeling approximately 188k Wikipedia sentences into eight coarse-grained entity types. *Conll* uses stories from Reuters news, encompassing four basic entity types. *Ener* focuses on legal texts, and we use its four-category version in this paper, treating all legal entities as miscellaneous

Algorithm 1 Iterative Pseudo-label Generation Algorithm

Input: Initial samples set $X^{(0)}$, an arbitrary LLM $\mathcal{M}$, test sample set X and the corresponding label set Y.
Parameter: Confidence threshold γ, and maximum iterations J.
Output: Test accuracy on X.

for $j = 0$ **to** $J - 1$ **do**
  $\tilde{Y}^{(j)} = \emptyset$.
  for $x_t \in X$ **do**
    Search for $x_1, y_1, ..., x_{C\times N}, y_{C\times N}$ based on tf-idf similarity where $x_t \notin \{x_1, ..., x_{C\times N}\}$.
    Calculate $C\hat{P}P_t$ for $\mathcal{M}$ and the corresponding prediction results $\tilde{y}$.
    $\tilde{Y}^{(j)} \leftarrow \tilde{y}$.
    if $C\hat{P}P_t > \gamma$ and $x_t \notin X^{(j)}$ **then**
      $X^{(j)} \leftarrow x_t$.
    end if
  end for
end for
$\tilde{Y}$ =MajorityVoting$(\tilde{Y}^{(0)}, \tilde{Y}^{(1)}, ..., \tilde{Y}^{(J-1)})$.
Evaluate and return the test accuracy by $\tilde{Y}$ and Y.

ones. *Wnut* gathers training data from Twitter and mining test data from sources like Reddit, StackExchange, Twitter, and YouTube, containing six entity types.

D.2 Experimental Setup

Training details of confidence selectors. For SA, TD, and NLI, we train 3 epochs on the OOD dataset with a learning rate of 2e-5 on DeBERTa[8]. For NER, we train a T5-base[9] with the same configuration (Raffel et al., 2020). The reason for using them is that they are sufficiently light compared to LLMs, allowing them to learn the label information of the source domain effectively within a limited time.

Inference details for ICL. For all input instructions for LLMs, we adopt the instruction templates reported by (Yuan et al., 2023). Referring to the settings of (Liu et al., 2022), we set N of the main experiment to 3, and the number of iterations J to 3. To ensure stable performance and avoid issues stemming from sample size imbalance and label order variance, we deliberately fix both the number of sample categories and their ordering throughout the process. We use accuracy as the measure of performance. All experiments are run on a single NVIDIA A40 GPU. For large-scale models that can not be deployed locally, we utilize the corresponding APIs[10] and systematically parse their responses for downstream processing.

Data Processing. We carry out our experiment using the data division given by BOSS. But for the three datasets in the NER task, we select only three types of entities (Person, Organization, and Location) to unify the label space. To avoid excessively reducing the amount of test data, we further sample from the training sets reported in the original papers of the corresponding datasets. As a result, the maximum number of available samples for each dataset is 5,000. Moreover, to reduce inference costs for large-scale LLMs, we use only the first 500 samples from each dataset for experimentation if the corresponding test set contains more than 500 samples following (Fei et al., 2023).

D.3 Baselines

- Random. The demonstrations for each test sample are randomly selected in the source datasets.
- Bm25 (Luo et al., 2023). It calculates the word overlap similarity between source samples and target samples, and selects samples with high similarity for ICL.
- K-nearest neighbors (KNN) (Liu et al., 2022). We use SimCSE as the sample encoder for sample embeddings. Then, for each sample from the target set, k-nearest neighbors from the source samples are selected as the demonstrations.
- TopK+MDL (Wu et al., 2023). It first selects the 30 most similar samples as candidates based on KNN, and then utilizes the Minimum Description Length (MDL) principle to further select the most influential combination of samples from the candidates.
- DrICL (Luo et al., 2023). Bm25 is employed to retrieve samples similar to the one being predicted. Subsequently, a LLM is used to provide feedback that distinguishes between beneficial and detrimental samples with respect to the prediction outcome. These samples are then used

[8]https://huggingface.co/microsoft/deberta-v3-base
[9]https://huggingface.co/google-t5/t5-base
[10]https://api.xi-ai.cn/

Question: What is the label of text "{}", negative, positive or neutral?
Answer:
Question: What is the label of "{label}" text "{}", negative, positive or neutral?
Answer: SA

Question: What is the label of text "{}", toxic or not? \n Answer:
Question: What is the label of "{label}" text "{}", toxic or not? \n Answer: TD

Question: What is the label of text "{}", entailment, neutral or contradiction?
Answer:
Question: What is the label of "{label}"
text "{}", entailment, neutral or contradiction? Answer: NLI

Question: What entities are there in the text "{}"? Answer:
Question: What entities are there in the text "{}" with entities "{Person: xxx, Location: xxx .....}"? Answer: NER

Figure 9: Templates for constructing perturbation sample pairs for different tasks.

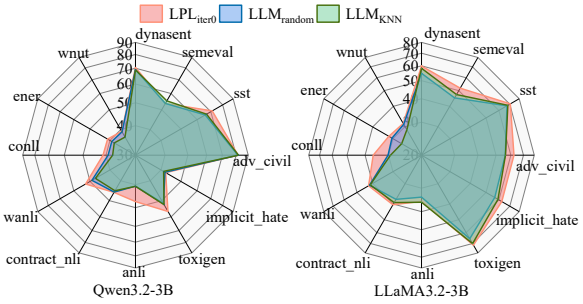

Figure 10: Comparison of LPL (without iteration) with other methods across all datasets.

as positive and negative instances to fine-tune GTR (Ni et al., 2022) as a dense retriever.

- TopK+ConE (Peng et al., 2024b). It searching for demonstrations that minimize the difference of the crossentropy between the prompt and the demonstrations. To mitigate the large search space introduced by exhaustive enumeration, the method first employs KNN to retrieve the top 10 most similar samples to the target instance, and searches for the optimal demonstration combination within this narrowed candidate set.

- DICL (Kapuriya et al., 2025). It utilizes Maximum Marginal Relevance (MMR), which balances topic similarity and diversity among examples during demonstration retrieval. In practice, the parameter for trading off similarity and diversity is set to 0.5, indicating equal importance for both similarity and diversity.

D.4 Perturbation Form

We provide detailed perturbation prompt pairs for different tasks in Figure 9. Unlike the prompts required in ICL, the perturbation prompt pair targets only the test sample and does not include any demonstrations, as its primary purpose is to quantify the change in perplexity of the test sample's prediction results with and without labels. On the other hand, this also reduces the time cost required for inference.

E Complexity Analysis

One concern regarding the use of LPL is complexity, as the iterative process introduces additional time overhead. Considering that the inference complexity of LLMs depends not only on the model size but also on the number of demonstrations, we can approximate the model's complexity in our LPL demonstration setup as $O(N \times C)$, where $N \times C$ is the number of demonstrations and $O(1)$ is the cost per demonstration. The CPP computation, which requires estimating a sample pair, introduces an additional complexity of $O(2)$. Taking the iterative process into account, the overall complexity can be expressed as $J \times O(N \times C) + (J-1) \times O(2)$, since the CPP computation for the first iteration can be pre-executed. In practice, the number of iterations can also be reduced to one to achieve a trade-off between performance and computational overhead.

F More Results of Iteration and Majority Voting

Figure 8 shows the performance of LPL across all datasets, as well as the change in the number of included samples with the number of iterations. Similar to the results presented in Section 4.2.5, the number of selected samples increases with iterations, but performance does not always improve accordingly in some cases (e.g., wanli, wnut), indicating that the iterative outputs of LLMs can be unstable. Majority voting can help mitigate the worst-case predictions to some extent. For example, on toxigen, the result obtained by majority voting even surpasses that of any single iteration.

G LPL without Iteration

As discussed in Appendix E, the iterative process of LPL introduces additional computational overhead. When the number of iterations is set to 0, the cost of LPL becomes more aligned with practical constraints. Therefore, Figure 10 presents a detailed comparison between single-iteration LPL, the random selection method, and the KNN-based approach. The results show that even with a single iteration, LPL consistently outperforms the baseline methods on most datasets, regardless of the

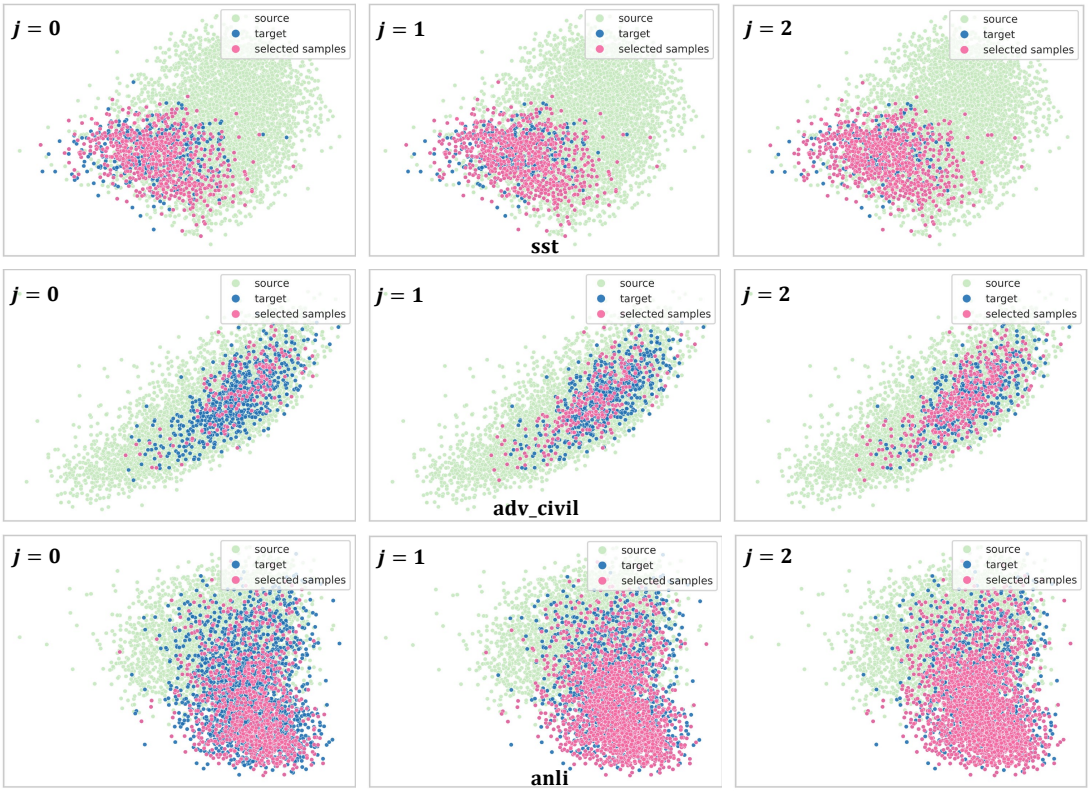

Figure 11: 2D visualization of samples from different sources.

underlying LLMs. This indicates that LPL can be flexibly adjusted to meet practical requirements to reduce computational overhead.

H More Visualization Results

Our visualization results on the adv_civil and anli datasets further confirm that the samples selected by LPL help mitigate distributional biases in the data. As the iterations proceed, the selected samples increasingly resemble the overall data distribution of the target domain, regardless of the source domain distribution.

I Case Study

Further case studies demonstrate how the iteration process can lead to more reasonable samples being included in the demonstrations in Figure 12. To present more intuitively, we adjust N to 1. In the initial sample selection, the positive sample "*Andy Garcia enjoys one of his richest roles in years and...*" is considered due to its high confidence score in the DeBERTa prediction. As the iteration progresses ($j = 1$), the sample "*Whether (Binoche and Magimel) are being charming .....*" which is more similar to the sample to be predicted, is replaced in the demonstration due to the high confidence score given by the LLMs. Samples that are more similar to the test sample tend to help LLMs make better predictions. In further iterations ($j = 2$), the negative sample is also replaced, which increases the diversity of the demonstrations and helps to consider a wider range of samples in the majority voting.

Text: Leave these Flowers unpicked -- they 're dead on the vine .
Prediction: negative
Text: Andy Garcia enjoys one of his richest roles in years and Mick Jagger gives his best movie performance since , well , Performance .
Prediction: positive
Text: A few artsy flourishes aside , Narc is as gritty as a movie gets these days .
Prediction: neutral
Text: Binoche and Magimel are perfect in these roles . Prediction: **j=0**

Text: Leave these Flowers unpicked -- they 're dead on the vine .
Prediction: negative
Text: Whether (Binoche and Magimel) are being charming or angst-ridden , they easily fill their scenes and , fine judges both , never overcook the hysteria .
Prediction: positive
Text: A few artsy flourishes aside , Narc is as gritty as a movie gets these days .
Prediction: neutral
Text: Binoche and Magimel are perfect in these roles . Prediction: **j=1**

Text: These are names to remember , in order to avoid them in the future .
Prediction: negative
Text: Whether (Binoche and Magimel) are being charming or angst-ridden , they easily fill their scenes and , fine judges both , never overcook the hysteria .
Prediction: positive
Text: A few artsy flourishes aside , Narc is as gritty as a movie gets these days .
Prediction: neutral
Text: Binoche and Magimel are perfect in these roles . Prediction: **j=2**

Figure 12: Case study on *sst*. We use different colored fonts to highlight the changes during iterations.