Performance-Driven Demonstration Selection for In-Context Learning

Wenqiang Wang[1†], **Mingbo Yang**[1†], **Aiping Zhang**[1], **Yan Xiao**[1]
Peng Chen[1], **Jianjie Huang**[1,3] and **Xiaochun Cao**[1,2*]
[1]Shenzhen Campus of Sun Yat-sen University, China
[2]Peng Cheng Laboratory, China
[3]Zhongguancun Academy, China
wangwq69@mail2.sysu.edu.cn, yangmb3@mail2.sysu.edu.cn, zhangaip7@mail2.sysu.edu.cn,
xiaoyan.hhu@gmail.com, 23pchen@stu.edu.cn, huangjj67@mail2.sysu.edu.cn,
caoxiaochun@mail.sysu.edu.cn

Abstract

In-context learning (ICL) enables large language models (LLMs) to adapt to new tasks with considerable performance gains, yet its effectiveness is highly sensitive to the choice of demonstrations. Most existing selection methods rely on heuristic or proxy signals, such as similarity, diversity, or uncertainty, and select demonstrations independently, which may misalign with downstream performance and overlook set-level composition effects. Therefore, we propose Performance-Driven Demonstration Selection (PDDS), which directly aligns demonstration selection with ICL performance. PDDS formulates selection as predicting the target LLM's downstream task performance for a given query–in-context pair, replacing proxy heuristics with a performance-aware objective. Unlike prior approaches that rank demonstrations independently before composing the prompt, PDDS evaluates the entire in-context set, capturing inter-demonstration interactions and composition effects. PDDS trains an end-to-end scorer using supervision from the target LLM's actual task outcomes. At inference time, it selects high-scoring in-context sets without additional target LLM calls or validation-time feedback. Across 6 NLP tasks, 8 datasets, and 8 target LLMs ranging from 1B to 70B parameters, PDDS achieves state-of-the-art results and generalizes well across LLMs and datasets. PDDS remains effective with as few as 30 training instances and can be integrated as a plug-and-play component to enhance existing ICL methods.

1 Introduction

In-context learning (ICL) allows large language models (LLMs) to adapt to new tasks with strong performance using only demonstrations, without parameter updates. [Ding *et al.*, 2026]; [Otero *et al.*, 2026]; [Jiang *et al.*, 2026] Despite its effectiveness, ICL is highly sensitive to the choice of demonstrations: different demonstration selections can lead to substantial performance variations, even when the demonstration number, the prompt, and the target LLM are fixed [Liu *et al.*, 2025];[Peng *et al.*, 2024] [Wang *et al.*, 2023];[Shi *et al.*, 2026];[Zuo *et al.*, 2025].

Most existing approaches select demonstrations by optimizing heuristic or proxy criteria, such as semantic similarity between the input query and candidate demonstrations, diversity among selected demonstrations, uncertainty of demonstrations, or likelihood-based scores computed under the target LLM. [Li *et al.*, 2025]; [Li and Qiu, 2023]; [Cai *et al.*, 2025]; [Liu *et al.*, 2024] While these criteria capture certain desirable properties of demonstrations, they are not optimized end-to-end with respect to the target LLM's downstream task performance. Consequently, demonstrations selected according to such proxy objectives do not necessarily translate into improved ICL performance and have been empirically shown to be suboptimal. This is corroborated by Table 1, where proxy-based selection methods (e.g., BM25, PPL [Gonen *et al.*, 2023], and CD [Naik *et al.*, 2023]) consistently underperform the best-performing approach.

Meanwhile, a growing line of work explores training-based supervised demonstration selection, where a retriever is trained using downstream task supervision [Liu *et al.*, 2025; Ye *et al.*, 2023; Wang *et al.*, 2024]; [Chen *et al.*, 2026]; [Chen *et al.*, 2025] [Wang *et al.*, 2025]. Despite being more principled than purely heuristic approaches, these methods still exhibit the following two limitations. On the one hand, many supervised selectors continue to rely on indirect surrogate signals, such as similarity- or uncertainty-derived labels or learned ranking scores. These signals do not explicitly model how the selected demonstrations align the LLM's performance. On the other hand, most existing training-based approaches define utility at the level of individual demonstrations and construct the in-context with k demonstrations via per-demonstration retrieval and ranking. They do not directly evaluate or optimize the set-level utility of a k-demonstration set. As a result, they fail to systematically capture inter-demonstration interactions, including complementarity, redundancy, and other composition-induced effects.

Therefore, we propose that demonstration selection for ICL can be directly aligned with the target LLM performance and the ultimate objective: how well the target LLM performs on the target task given an input query–in-context pair. Meanwhile, we argue that **constructing an in-context should ac-**

count not only for the contribution of individual demonstrations, but also for the joint effect induced by their combination within the in-context. We therefore reformulate demonstration selection as a performance prediction problem by training a predictor (scoring model) $f_\theta(x, C)$ that jointly encodes the query x and a candidate k-shot context C, and estimates the target LLM's downstream task performance for the pair (x, C). Supervised by the target LLM's actual task performance, f_θ is sensitive to the set of demonstrations in C rather than any individual demonstration, thereby capturing composition effects (e.g., complementarity and redundancy) and enabling set-level optimization beyond per-demonstration ranking.

To realize this formulation, we propose Performance-Driven Demonstration Selection (PDDS). PDDS learns a scoring model that predicts the target LLM's task performance for an input query-in-context pair. The scoring model is trained using supervision derived from the actual outputs of the target LLM, allowing it to capture how different demonstrations and in-contexts affect performance. Once trained, the scoring model can be used to efficiently evaluate candidate demonstrations and in-context at inference time, without requiring additional LLM calls or validation-time feedback. This design enables demonstration selection to be both performance-aligned and deployment-friendly.

A key challenge in PDDS stems from the combinatorial nature of in-context demonstration construction, which makes exhaustive evaluation over demonstration subsets computationally intractable. To address this challenge, we adopt a lightweight two-stage inference-time procedure guided by the PDDS scoring model. Specifically, we preselect the top-wk most promising demonstrations based on their predicted scores and sort them in descending order. We then form a small set of candidate k-shot contexts from this sorted pool (e.g., via a sliding window), score each candidate in-context *as a whole*, and select the highest-scoring one as the final in-context. By leveraging the learned scorer, our procedure avoids combinatorial search while staying directly aligned with the target objective.

We conduct extensive experiments on 6 NLP tasks and 8 LLMs (ranging from 1B to 70B parameters) to evaluate the effectiveness of PDDS. Results show that performance-driven demonstration selection consistently improves ICL performance over strong heuristic- and learning-based baselines. Further analyses demonstrate that our method generalizes across different LLMs and datasets, highlighting the benefits of directly modeling LLM performance for in-context construction. Meanwhile, the PDDS scoring model can be applied as a plug-and-play component to other ICL methods and improve their performance.

Our contributions can be summarized as follows: ❶ We reformulate demonstration selection for in-context learning as a performance prediction problem, providing a new perspective that directly aligns selection with downstream task performance. ❷ We propose a performance-driven demonstration selection framework that learns to predict LLM task performance from input query-in-context pairs and uses this prediction to construct effective in-contexts. ❸ We demonstrate through extensive experiments that our approach outperforms existing demonstration selection methods in both effectiveness and robustness, while remaining efficient at inference time.

2 Problem Formulation

We consider the standard ICL setting with a frozen target LLM. Let $\mathcal{D}_\text{r} = \{(x_i, y_i)\}$ denote a labeled retrieval dataset, where x_i is a text and y_i is its corresponding ground-truth label. Given a test input x_t, the goal of ICL is to construct a context $C = \{(x_j, y_j)\}, j = 1..k$ consisting of k demonstrations, such that the LLM achieves strong task performance when conditioned on (x_t, C).

2.1 Performance-Driven Demonstration Selection

In contrast, we formulate demonstration selection as a performance prediction problem. Our objective is to directly model the task performance of the target LLM conditioned on an input query-in-context pair. Let $\text{Perf}(\text{LLM}; x_\text{t}, C)$ denote a task-specific performance metric (e.g., MSE [Hyndman and Koehler, 2006], BLEU [Post, 2018], ROUGE-1 [Ganesan, 2018]) achieved by the LLM when conditioned on input query x_t and in-context C. We aim to learn a scoring model $f_\theta(\cdot)$:

$$f_\theta(x_\text{t}, C) \approx \text{Perf}(\text{LLM}; x_\text{t}, C), \tag{1}$$

where f_θ predicts the downstream task performance of the LLM for a given input query-in-context pair. Under this formulation, demonstration selection is defined as

$$C^* = \underset{C \subseteq \mathcal{D}_\text{r}, |C|=k}{\arg\max} f_\theta(x_t, C), \tag{2}$$

where $\mathcal{D}_\text{r}$ denotes the retrieval dataset. This formulation directly aligns the selection objective with downstream task performance.

3 Method

Figure 1 illustrates the overall framework of PDDS. Given an input query, PDDS first retrieves a pool of candidate demonstrations and constructs a set of candidate in-contexts. Unlike heuristic criteria, PDDS relies on a lightweight performance predictor trained with supervision from the *frozen* target LLM: for each query we construct candidate in-contexts and use the target LLM's task metric as labels to fit $f_\theta(x, C) \approx \text{Perf}(\text{LLM}; x, C)$. At inference time, PDDS performs a two-stage in-context construction by (i) preselecting and ranking promising demonstrations and (ii) selecting the final k demonstrations at the context level, where all scores are produced by f_θ without additional target-LLM calls. The target LLM is invoked only once with the selected in-context to produce the final prediction.

3.1 Training Data Construction

As the subfigure (a) Dataset construction in Figure 1 shows, training the scoring model for input query-in-context pairs requires supervision that reflects the downstream task performance of the target LLM. Therefore, we randomly split the retrieval dataset $\mathcal{D}_\text{r}$ into a query set $\mathcal{Q}$ and a demonstration pool $\mathcal{D}$. For each query instance $(x, y) \in \mathcal{Q}$, we first retrieve candidate demonstrations from $\mathcal{D}$ based on semantic

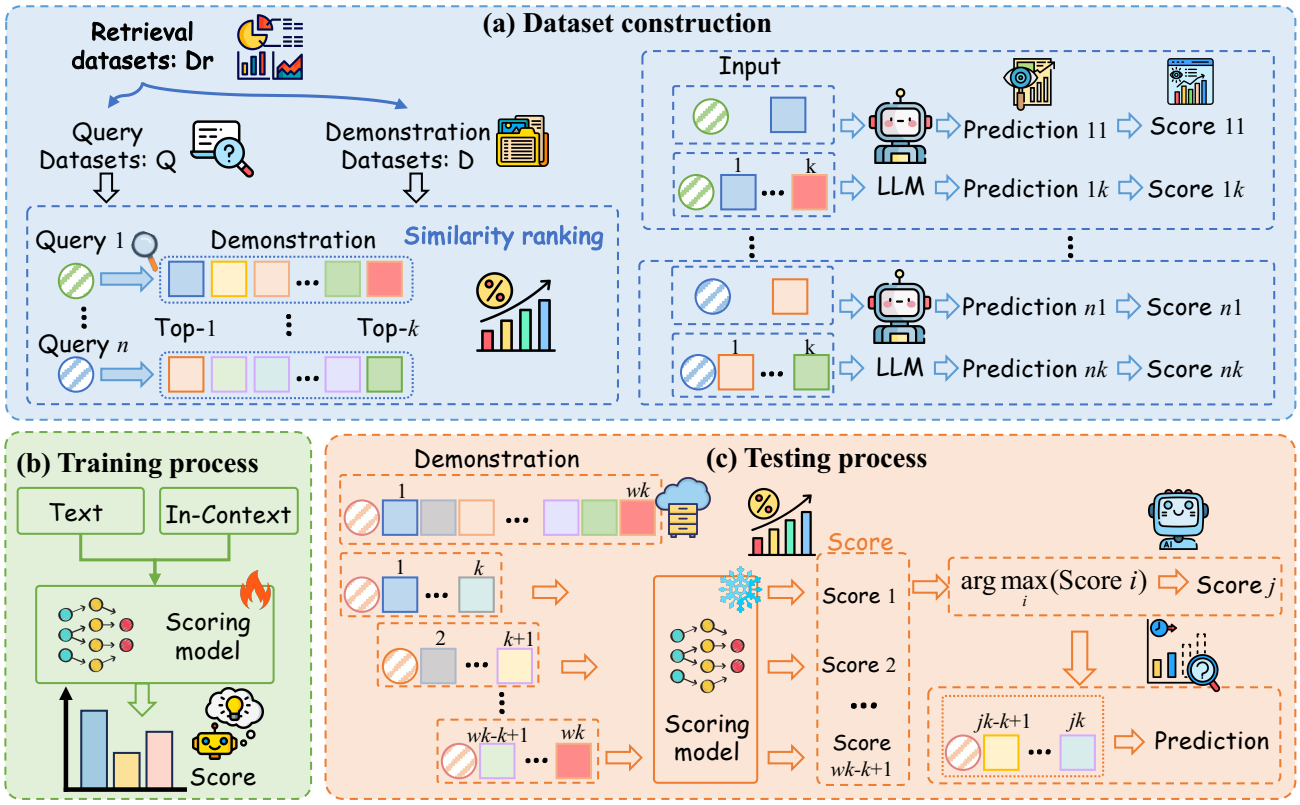

Figure 1: Overview of PDDS: a two-stage selection pipeline that predicts downstream LLM performance to construct in-context.

similarity. Specifically, let $\phi(\cdot)$ denote a sentence embedding function and $s(\cdot,\cdot)$ a similarity measure. Each candidate $(x', y') \in \mathcal{D}$ is ranked by

$$\mathrm{sim}(x, x') = s(\phi(x), \phi(x')). \tag{3}$$

Let π_x denote the ranking permutation in descending order of $\mathrm{sim}(x, x')$. Based on this ranking, we construct two types of in-contexts for each query. The single-demonstration context contains the most similar demonstration,

$$C^{(1)}(x) = \{(x_{\pi_x(1)}, y_{\pi_x(1)})\}, \tag{4}$$

while the k-demonstration in-context consists of the top-k ranked demonstrations,

$$C^{(k)}(x) = \{(x_{\pi_x(j)}, y_{\pi_x(j)})\}_{j=1}^{k}. \tag{5}$$

We then query the frozen target LLM with each input query-in-context pair (x, C) and compute a task-specific performance score. Formally, given a performance metric $\mathrm{Perf}(\cdot)$, we obtain

$$\begin{aligned} r^{(1)}(x) &= \mathrm{Perf}(\mathrm{LLM}; x, C^{(1)}(x)), \\ r^{(k)}(x) &= \mathrm{Perf}(\mathrm{LLM}; x, C^{(k)}(x)). \end{aligned} \tag{6}$$

These scores serve as supervision signals for learning the performance predictor (scoring model). This construction yields informative training signals without requiring exhaustive in-context enumeration. By contrasting in-contexts of different sizes and informational content, the scoring model is encouraged to learn how contextual composition influences downstream performance, rather than relying on superficial properties of individual demonstrations. As a result, the learned predictor (scoring model) can generalize beyond the specific contexts observed during training and be applied to arbitrary candidate in-contexts at inference time.

3.2 Performance Scoring Model

Given the constructed training set of input query-in-context pairs and their corresponding performance scores, we aim to learn a scoring model that predicts the downstream task performance of the target LLM. The scoring model takes an input query x together with a candidate in-context C as inputs and outputs a scalar score estimating the LLM's performance when conditioned on (x, C).

Performance Definition. Let $\mathrm{Perf}(\mathrm{LLM}; x, C) \in \mathbb{R}$ denote a task-specific performance score of the LLM under input x and context C. Without loss of generality, we define $\mathrm{Perf}(\cdot)$ such that larger values indicate better performance. For evaluation metrics where lower values correspond to better performance (e.g., error or loss), we use the negated metric value as $\mathrm{Perf}(\cdot)$. This convention ensures a consistent maximization objective across different tasks.

Model Input Representation. The scoring model operates on the full input query-in-context pair (x, C). The in-context $C = \{(x_j, y_j)\}_{j=1}^{|C|}$ is treated as an unordered set of demon-

strations, where each demonstration consists of an input query-in-context pair. Rather than scoring individual demonstrations independently, the model evaluates the in-context as a whole, enabling it to capture interactions among demonstrations that jointly influence downstream performance.

Scoring Function. Let $f_\theta(\cdot)$ denote the performance scoring model parameterized by θ. The scoring model predicts a scalar score

$$\hat{r}(x, C) = f_\theta(x, C), \tag{7}$$

which serves as an estimate of the true performance $\text{Perf}(\text{LLM}; x, C)$.

Learning Objective. During training, supervision for the scoring model is obtained from the actual outputs of the target LLM. Given a training instance (x, C, r), where $r = \text{Perf}(\text{LLM}; x, C)$, we optimize the model by minimizing a regression mean squared error (MSE) loss [Hyndman and Koehler, 2006]:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{(x,C,r)} \left[\left(f_\theta(x, C) - r \right)^2 \right]. \tag{8}$$

This objective trains the scoring model to accurately rank different in-contexts by their downstream task performance. Unlike prior demonstration selection methods that rely on proxy objectives or independently score individual demonstrations, our scoring model directly estimates the task performance of the target LLM under a complete input query-in-context pair. Therefore, the scoring model is trained in an end-to-end manner aligned with the downstream performance. This alignment empirically leads to more effective demonstration selection than proxy-driven criteria.

3.3 Inference-Time In-Context Construction

As the subfigure (c) Testing process in Figure 1 shows, at inference time, given a test input query x_{t}, we aim to construct a k-demonstration in-context that maximizes the predicted task performance. However, directly enumerating and evaluating all possible in-contexts is computationally infeasible due to the combinatorial nature of demonstration selection. Therefore, we use a two-stage procedure guided by the learned performance scoring model.

Stage 1: **Preselect promising demonstrations.** For each candidate demonstration $(x_i, y_i) \in \mathcal{D}_{\text{r}}$ (the retrieval dataset), we score the corresponding single-demonstration in-context:

$$\hat{r}_i = f_\theta(x_{\text{t}}, \{(x_i, y_i)\}). \tag{9}$$

We then keep the top-wk demonstrations with the largest $\hat{r}_i$, denoted as $\mathcal{D}_{wk}(x_{\text{t}})$.

Stage 2: **Construct and score candidate contexts.** We sort $\mathcal{D}_{wk}(x_{\text{t}})$ by $\hat{r}_i$ in descending order and generate candidate in-contexts using a sliding window of size k:

$$\mathcal{C}(x_{\text{t}}) = \left\{ \{(x_{(j)}, y_{(j)})\}_{j=t}^{t+k-1} \;\middle|\; t = 1, \ldots, wk - k + 1 \right\}, \tag{10}$$

where $(x_{(j)}, y_{(j)})$ denotes the j-th demonstration in the sorted list. Each candidate in-context $C \in \mathcal{C}(x_{\text{t}})$ is evaluated by

$$\hat{r}(x_{\text{t}}, C) = f_\theta(x_{\text{t}}, C), \tag{11}$$

and the final in-context is selected as

$$C^\star = \arg\max_{C \in \mathcal{C}(x_t)} \hat{r}(x_t, C). \tag{12}$$

All inference-time scores are computed by the lightweight predictor (scoring model) f_θ, without any additional calls to the target LLM. We feed the query and selected in-context to the target LLM for final prediction.

4 Experiment

4.1 Experimental Setup

(i)**Tasks and Datasets.** We cover six task types over eight datasets: Translation (WMT19 En–Zh dataset [Barrault *et al.*, 2019]), text expansion (Gigatiny dataset [Rush *et al.*, 2015]), summarization (Gigaword dataset), Semantic Textual Similarity (STS) (STS-B/STS14 datasets [Cer *et al.*, 2017; Agirre *et al.*, 2014]), classification (SST-5/Emotion datasets [Socher *et al.*, 2013; Saravia *et al.*, 2018]), and Translation quality evaluation (TQA) (EN–CS dataset [Ma *et al.*, 2019; Barrault *et al.*, 2019]), spanning classification/generation and discrete/continuous outputs.

(ii) **Metrics and Baselines.** We use BLEU [Post, 2018] (Translation), ROUGE-1 [Ganesan, 2018] (expansion/summarization), accuracy (classification), and report Mean Squared Error (MSE) [Hyndman and Koehler, 2006] for the STS and translation-quality evaluation tasks. Baselines include proxy-based demonstration selection methods DKNN [Li *et al.*, 2025], CR [Li and Qiu, 2023], KATE [Liu *et al.*, 2022], CD [Naik *et al.*, 2023], ICCL [Liu *et al.*, 2024], PPL [Gonen *et al.*, 2023] (k=10), and training-based demonstration selection methods TTF [Liu *et al.*, 2025], CEIL [Ye *et al.*, 2023], and MoD [Wang *et al.*, 2024].

(iii) **Target LLMs and Other Setups.** Our main LLMs are GLM-4 9B, LLaMA 3.1 8B, and Qwen2.5 7B; we further test 1B–70B LLMs (LLaMA 3.2 1B, DeepSeek 32B [Guo *et al.*, 2025], LLaMA 3.3 70B and Qwen3 4B/8B [Yang *et al.*, 2025]). The number of demonstrations for all methods is 10. We randomly split the retrieval dataset $\mathcal{D}_{\text{r}}$ into a query set $\mathcal{Q}$ and a demonstration pool $\mathcal{D}$, where $\mathcal{Q}$ and $\mathcal{D}$ account for 20% and 80% of the data, respectively. The wk in Eq. 10 is 20.

4.2 Main Results

Table 1 reports the overall performance across six tasks and eight datasets. PDDS consistently achieves strong results and attains the best performance in the majority of settings. Specifically, it improves BLEU by 0.032 on machine translation, increases ROUGE-1 by 0.009 and 0.016 on summarization and expansion, respectively, reduces MSE by 0.060 on STS14, and exceeds the strongest competing baseline by 5.8% accuracy on text classification.

Overall, these results show that directly optimizing predicted LLM performance leads to more effective and robust demonstration selection than proxy-driven heuristics or contrastive-learning-based methods.

4.3 Ablation Study

We conduct ablation studies to quantify the contribution of each design choice to overall performance, including the

LLMs	**GLM**	**Llama**	**Qwen**	**GLM**	**Llama**	**Qwen**	**GLM**	**Llama**	**Qwen**	**GLM**	**Llama**	**Qwen**
Metric	Accuracy (%) ↑			Accuracy (%) ↑			ROUGE-1 ↑			ROUGE-1 ↑		
Method	Classification: Emotion			Classification: SST5			Summarization: Gigaword			Expansion: Gigatiny		
BM25	54.3	38.0	59.0	34.2	31.4	49.9	0.024	0.020	0.108	0.122	0.128	0.135
CD	46.0	39.8	55.5	43.3	36.4	45.9	0.043	0.025	0.134	0.172	0.102	0.240
Kate	63.5	40.0	72.3	43.9	35.7	51.8	0.057	0.032	0.106	0.203	0.125	0.253
DKNN	68.8	46.5	43.0	49.8	35.7	45.0	0.033	0.028	0.114	0.167	0.107	0.237
TTF	52.5	38.5	52.9	45.0	38.5	46.8	0.043	0.036	0.114	0.163	0.099	0.264
CEIL	68.7	45.4	44.9	49.2	36.8	45.3	0.040	0.033	0.118	0.192	0.110	0.242
MoD	57.3	41.8	52.6	47.4	39.7	47.7	0.048	0.039	0.116	0.192	0.108	0.237
ICCL	52.3	38.2	69.7	46.2	31.7	48.8	0.043	0.032	0.106	0.194	0.115	0.230
PPL	60.2	39.4	51.0	48.2	32.1	48.7	0.043	0.037	0.115	0.197	0.110	0.249
PDDS (Ours)	**73.1**	**62.1**	**75.7**	**53.8**	**44.2**	**55.1**	**0.063**	**0.048**	**0.147**	**0.220**	**0.137**	**0.287**
Metric	MSE ↓			MSE ↓			BLEU ↑			MSE ↓		
Task & Data	STS:STS14			STS:STSB			Translation: WMT19 En-Zh			TQA EN-CS		
BM25	2.192	1.404	0.797	0.993	1.146	0.763	0.032	0.033	0.040	3489.0	531.9	352.0
CD	2.831	1.263	0.965	1.272	1.291	0.898	0.204	0.134	0.225	493.3	686.1	354.2
Kate	0.920	1.441	0.813	1.070	1.138	0.747	0.211	0.137	0.104	452.6	499.8	343.9
DKNN	1.087	1.511	0.804	1.258	1.425	0.781	0.202	0.138	0.174	453.1	517.1	355.0
TTF	3.041	1.578	0.748	1.343	1.583	0.982	0.231	0.146	0.008	498.6	666.2	401.8
CEIL	1.388	1.366	0.972	1.202	**1.021**	0.790	0.219	0.108	0.175	496.5	493.4	392.1
MoD	1.343	1.376	1.074	1.101	1.230	0.805	0.222	0.073	0.238	436.4	473.4	358.7
ICCL	2.252	1.361	0.727	1.118	1.199	0.786	0.214	0.139	0.111	427.6	484.4	348.7
PPL	2.538	1.372	0.989	1.101	1.403	0.816	0.219	0.144	0.165	410.3	462.4	347.5
PDDS (Ours)	**0.832**	**1.249**	**0.650**	**0.813**	1.122	**0.718**	**0.239**	**0.195**	**0.277**	**405.4**	**430.0**	**336.9**

Table 1: Results of PDDS and other ICL methods. The best and second-best results are highlighted in bold and underline, respectively. Model: GLM, Llama, and Qwen indicate GLM4-9B, LLAMA3.1-8B, and Qwen2.5-7B, respectively.

scoring model, the two-stage in-context construction strategy, and the preselection size w_k. **Ablating performance-driven scoring model (f_θ vs. proxy criteria).** We ablate f_θ by constructing in-contexts using only Eq. 5 (KATE-style retrieval) without performance scoring (Figure 2). Without the scoring model, the performance of PDDS degrades noticeably. Specifically, the average accuracy drops by 9.47%, the average MSE worsens by 0.12, the average ROUGE-1 decreases by 0.02, and the average BLEU drops by 0.09. These consistent drops underscore the scoring model's critical role in PDDS.

Ablating two-stage in-context construction. In PDDS, in-context construction is performed in two stages. In Stage 1, PDDS retrieves w_k candidate demonstrations and ranks them in descending order according to the predicted performance scores. In Stage 2, the scoring model is further applied to select the final k demonstrations from the ranked candidate set to form the final in-context. We now analyze the contribution of each stage separately.

Ablating Stage 1. To evaluate the role of Stage 1, we replace performance-driven preselection with the similarity criterion in Eq. 3 to retrieve the top-w_k most similar candidates, and then apply Stage 2 to construct the final in-context. As shown in Figure 2, removing Stage 1 consistently hurts performance: the average accuracy drops by 4.52%, the average MSE worsens by 0.04, the average ROUGE-1 decreases by 0.01, and the average BLEU drops by 0.01. This underscores the importance of Stage 1 in PDDS.

Settings	With scoring model			Without scoring model		
LLMs	GLM	Llama	Qwen	GLM	Llama	Qwen
Method	Task: Classification Dataset: Emotion Metric: Accuracy (%) ↑					
BM25	63.7	44.1	64.3	54.3	38.0	59.0
CD	53.7	43.5	62.6	46.0	39.8	55.5
Method	Task: Summarization Dataset: Gigaword Metric: ROUGE-1 ↑					
BM25	0.034	0.026	0.116	0.024	0.020	0.108
CD	0.051	0.031	0.141	0.043	0.025	0.134

Table 2: The results of cross-retrieval generalization (plug-and-play). BM25 and CD denote using the corresponding method to retrieve and rank a candidate pool of wk demonstrations. With scoring model uses PDDS Stage 2 to construct the final context from BM25/CD-ranked wk candidates. PDDS can be a plug-and-play module for diverse ICL pipelines.

Ablating Stage 2. To ablate Stage 2, we form the in-context by directly selecting the top-k demonstrations from the ranked w_k candidates produced in Stage 1. As shown in Figure 2, excluding Stage 2 consistently degrades performance: average accuracy drops by 5.02%, MSE increases by 0.26, and ROUGE-1/BLEU each decrease by 0.01. These results underscore the importance of Stage 2 in PDDS.

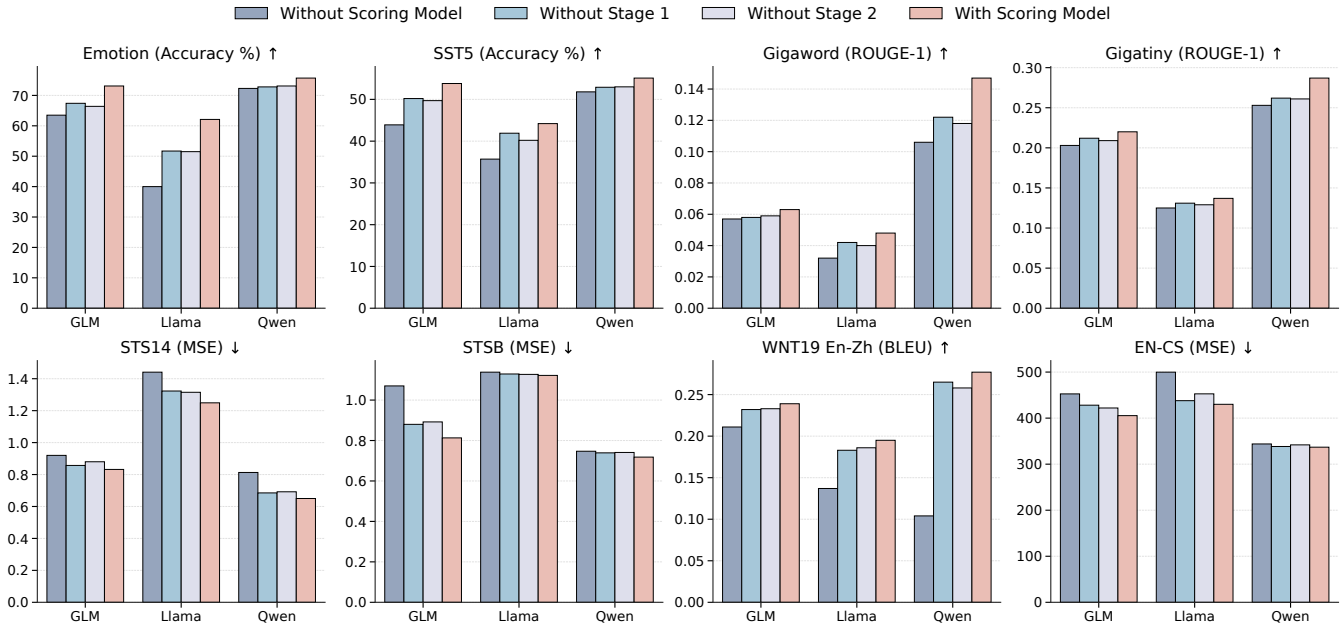

Figure 2: Ablation results of the performance-driven scoring model f_θ and the two-stage in-context construction in PDDS.

LLMs	GLM	Llama	Qwen	GLM	Llama	Qwen
Metric	Accuracy (%) ↑			MSE ↓		
Method	Training dataset: SST5 Test dataset: Emotion			Training dataset: STSB Test dataset: STS14		
PDDS	**70.2**	**54.6**	**71.6**	**0.885**	**1.287**	**0.720**

Table 3: The results of cross-dataset generalization. Training dataset denotes the dataset used to train the PDDS scoring model. Test dataset denotes the dataset on which the scoring model is applied to construct the final in-context prompt.

Sensitivity to preselection size wk. We set $wk = 20$ by default and evaluate $wk \in \{10, 15, 20, 25\}$. Performance improves as wk increases but quickly saturates; e.g., on Emotion with Llama, accuracy rises from 51.5% ($wk = 10$) to 62.1% ($wk = 25$) and then stays around 62.1%. A moderate candidate pool captures most gains, while larger wk mainly adds computation.

5 Generalization

Cross-retrieval generalization (plug-and-play). Given a fixed external retriever, we *only* score the retrieved candidate pool (with wk candidates) using our performance-driven scoring model and then select the top-k demonstrations accordingly, without retraining or modifying the retriever. As shown in Table 2, replacing the original ranking/selection strategy with our plug-and-play scoring consistently improves performance. Specifically, it increases the average accuracy from 48.77% to 55.32% and the average ROUGE-1 from 0.059 to 0.067. These results indicate that the scoring model captures performance-relevant signals beyond retriever-specific heuristics and can be integrated in a plug-and-play manner into existing ICL pipelines.

Cross-dataset generalization. We examine whether the scoring model generalizes to unseen datasets beyond those used for training. Specifically, we train the scoring model on SST5 and evaluate it on Emotion under the same classification setting. As shown in Table 3, the transferred scoring model achieves an average accuracy of 65.47% and consistently outperforms the baselines reported in Table 1. These results indicate that performance-driven selection learns dataset-agnostic signals that remain predictive across distribution shifts, rather than overfitting to the training corpus.

Cross-LLM generalization. Finally, we evaluate whether the scoring model transfers across target LLMs. As shown in Table 5, a scoring model trained with one target LLM can be directly applied to a different target LLM. Although cross-LLM transfer is slightly worse than training on the target LLM itself (by Llama3.1 8B), it still yields strong performance: when trained on GLM4 9B, the PDDS achieves 53.2 accuracy on Llama3.1 8B. We conjecture that, since both GLM4 9B and Llama3.1 8B are trained to approximate the same underlying text-to-label mapping, they may induce similar decision boundaries (or decision regions) over the input query-in-context space. As a result, they tend to exhibit similar behaviors on the same input query-in-context pair, which helps explain why the scoring model can generalize across LLM backbones. Overall, these results suggest that the scoring model captures relatively model-agnostic signals correlated with in-context performance.

Method	BM25	CD	Kate	DKNN	TTF	ICCL	PPL	CEIL	MoD	PDDS
Token ↓	195.9	202.3	189.3	203.8	182.5	196.2	178.6	194.5	199.6	189.5
Time (s) ↓	1.95	0.26	0.27	0.22	0.27	0.36	0.32	0.40	0.43	0.31

Table 4: Results of token and time costs on the SST-5 dataset, averaged across three LLMs. Time is measured in seconds per text, and token cost is measured in tokens per text.

Dataset & Metric	Emotion & Accuracy (%) ↑			Gigaword & ROUGE-1 ↑		
Testing LLM	Llama			Llama		
Training LLM	GLM	Llama	Qwen	GLM	Llama	Qwen
PDDS	53.2	62.1	52.6	0.046	0.048	0.044

Table 5: The results of cross-LLM generalization. Training LLM denotes the LLM used to provide performance scores for constructing PDDS training data. Testing LLM denotes the LLM used to construct the in-context during inference.

LLMs	Llama3.2-1b	Llama-3.3-70b	DeepSeek-32b	Qwen3 4b	Qwen3 8b
	Emotion-Accuracy (%) ↑				
BM25	61.1	67.9	70.9	<u>69.8</u>	68.8
CR	40.2	58.3	59.5	54.5	55.4
CD	39.2	57.6	60.7	53.3	56.7
ICCL	<u>64.6</u>	<u>69.7</u>	70.9	68.1	71.0
Kate	63.6	69.4	<u>71.1</u>	67.4	<u>72.2</u>
PPL	38.4	59.3	58.1	53.4	50.5
DKNN	47.1	58.5	59.0	61.6	53.3
CEIL	42.6	61.3	61.8	57.1	54.9
MoD	35.8	57.1	58.9	56.1	54.4
PDDS	**66.8**	**77.9**	**79.2**	**73.4**	**75.6**

Table 6: Experimental results on additional Large Language Models for Emotion. The best results are highlighted in bold, and the second-best results are underlined.

6 Discussion

Evaluation across target LLM scales and families. We further evaluate PDDS on five additional target LLMs (1B–70B). As shown in Table 6, PDDS remains competitive across scales and families, achieving up to 79.2% accuracy. Overall, PDDS transfers well across LLM families and scales, suggesting the scorer captures performance-relevant signals.

Cost considerations. We evaluate PDDS overhead on SST-5 in terms of training time, storage, token cost, and latency. Training the scoring model takes 1.2 hours and 472 MB. At inference, PDDS uses 189.5 tokens and 0.31 seconds per query on average (Table 4). Overall, the one-time training cost is amortized over repeated use, making PDDS practically acceptable.

Training data efficiency (few-shot supervision). We study whether the PDDS scoring model can be effectively trained with only a small amount of labeled data. Specifically, we limit the number of training queries to 30. To enrich supervision in this setting, we construct additional single- and k-demonstration in-contexts, $C^{(1)}(x)$ and $C^{(k)}(x)$ (Equ. 4 and 5). These in-contexts are generated not only via similarity-based retrieval, but also by leveraging retrieval strategies from several ICL methods, including BM25, CD, and PPL. This yields a diverse set of query–in-context–performance triples despite the limited supervision, enabling stable training of the scoring model. As shown in Table 7, PDDS still achieves up to 52.0% accuracy with 30-shot supervision.

Dataset & Metric	SST5 & Accuracy (%) ↑		
Shots	GLM	Llama	Qwen
30	45.7	38.4	52.0
Dataset & Metric	WMT19 En-Zh & BLEU ↑		
30	0.215	0.156	0.138

Table 7: The results of few-shot (30) training data. GLM, Llama, and Qwen indicate GLM4-9B, LLAMA3.1-8B, and Qwen2.5-7B.

7 Conclusion

We revisit demonstration selection for ICL from a performance-driven perspective. Specifically, we formulate it as predicting downstream LLM performance for an input query–in-context pair, rather than optimizing heuristic proxies. Based on this formulation, we propose a lightweight and end-to-end framework that learns a scoring model to directly score and construct effective in-contexts.

Extensive experiments across six task categories, eight datasets, and eight target LLMs show that our method consistently improves ICL performance by aligning context construction with predicted LLM performance. Further analyses demonstrate that the scoring model generalizes across datasets, tasks, and target LLMs, and can be effectively trained with few-shot data, making the approach practical and scalable. Meanwhile, our method can be applied as a plug-and-play approach to improve the performance of other ICL methods. Overall, our results suggest that aligning context construction with predicted LLM performance offers a principled alternative to proxy-based heuristics, and reframes context selection as an explicit, performance-oriented decision problem in ICL.

Acknowledgements

Supported by the Shenzhen Science and Technology Program (CJGJZD20240729141505007), the National Natural Science Foundation of China (Nos. U2541229, 62441619, and 62411540034), and the Ningbo Science and Technology Innovation 2025 Major Project (2025Z027), and the National Natural Science Foundation of China under Grant 62502550, Shenzhen Science and Technology Program (KJZD20240903095700001).

Contribution Statement

† Wenqiang Wang and Mingbo Yang contributed equally to this work as co-first authors. * Xiaochun Cao is the corresponding author. Wenqiang Wang made the primary contribution to this work, including developing the main idea, formulating the performance-driven demonstration selection framework, designing the scoring model and the two-stage in-context construction strategy, conducting the main analysis, and writing the manuscript. Mingbo Yang contributed to code implementation, experiment execution, and result organization. Aiping Zhang contributed to methodological discussions, experimental analysis, and manuscript revision. Yan Xiao and Peng Chen contributed to dataset preparation, baseline comparison, and experimental evaluation. Jianjie Huang contributed to technical discussions and manuscript refinement. Xiaochun Cao supervised the project, provided overall research guidance, and contributed to the final manuscript revision.

References

[Agirre *et al.*, 2014] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91, 2014.

[Barrault *et al.*, 2019] Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. Findings of the 2019 conference on machine translation (wmt19). ACL, 2019.

[Cai *et al.*, 2025] Yuyang Cai, Peng Chen, Feng Qian, Gao Chen, and Jingwen Yan. Botwavenet: semantic segmentation network for remote sensing images based on wavelet transform and attention mechanisms. *Journal of Electronic Imaging*, 34(2):023053–023053, 2025.

[Cer *et al.*, 2017] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, 2017.

[Chen *et al.*, 2025] Peng Chen, Wenxuan He, Feng Qian, Guangyao Shi, and Jingwen Yan. A synergistic cnn-transformer network with pooling attention fusion for hyperspectral image classification. *Digital Signal Processing*, 160:105070, 2025.

[Chen *et al.*, 2026] Peng Chen, Fangjun Huang, and Chao Huang. Dyc-clip: Dynamic context-aware multi-modal prompt learning for zero-shot anomaly detection. *Pattern Recognition*, page 113215, 2026.

[Ding *et al.*, 2026] Yu Ding, Xudong Han, Junjie Yang, Tianyang Wang, Ziqian Bi, Xinyuan Song, Junfeng Hao, Junhao Song, Enze Ge, Benji Peng, et al. Few-shot and zero-shot learning with large language models: Prompting strategies and in-context learning. *Authorea Preprints*, 2026.

[Ganesan, 2018] Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*, 2018.

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

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[Hyndman and Koehler, 2006] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[Jiang *et al.*, 2026] Zhonglin Jiang, Shenghan Gao, Qian Tang, Zequn Wang, Yu Liu, and Hong-Zhong Huang. Generative reliability-based design optimization using in-context learning capabilities of large language models. *Journal of Mechanical Design*, 148(3):031705, 2026.

[Li and Qiu, 2023] Xiaonan Li and Xipeng Qiu. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts. *CoRR*, 2023.

[Li *et al.*, 2025] Chuyuan Li, Raymond Li, Thalia S Field, and Giuseppe Carenini. Delta-knn: Improving demonstration selection in in-context learning for alzheimer's disease detection. *arXiv preprint arXiv:2506.03476*, 2025.

[Liu *et al.*, 2022] Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 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, 2022.

[Liu *et al.*, 2024] Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024.

[Liu *et al.*, 2025] Hui Liu, Wenya Wang, Hao Sun, Chris Xing Tian, Chenqi Kong, Xin Dong, and Haoliang Li. Unraveling the mechanics of learning-based demonstration selection for in-context learning. In *Association for Computational Linguistics 2025*, 2025.

[Ma *et al.*, 2019] Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, 2019.

[Naik *et al.*, 2023] Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi.

Diversity of thought improves reasoning abilities of large language models. 2023.

[Otero *et al.*, 2026] Abraham Otero, Daniel García, Luciano Sánchez[1], and Nahuel Costal. Can in-context learning enable large vision language models to detect ecg. In *Artificial Intelligence in Biomedicine: First Conference of the Spanish Society of Artificial Intelligence in Biomedicine, CIABiomed 2025, Seville, Spain, October 23–24, 2025, Proceedings*, page 200. Springer Nature, 2026.

[Peng *et al.*, 2024] Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. *arXiv preprint arXiv:2401.12087*, 2024.

[Post, 2018] Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[Rush *et al.*, 2015] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[Saravia *et al.*, 2018] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Hsuan Chen. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, 2018.

[Shi *et al.*, 2026] Lida Shi, Fausto Giunchiglia, Ran Luo, Daqian Shi, Rui Song, Xiaolei Diao, and Hao Xu. An empirical study of llms via in-context learning for stance classification. *Information Processing & Management*, 63(1):104322, 2026.

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

[Wang *et al.*, 2023] Wenqiang Wang, Chongyang Du, Tao Wang, Kaihao Zhang, Wenhan Luo, Lin Ma, Wei Liu, and Xiaochun Cao. Punctuation-level attack: single-shot and single punctuation attack can fool text models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 49312–49324, 2023.

[Wang *et al.*, 2024] Song Wang, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. Mixture of demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 37:88091–88116, 2024.

[Wang *et al.*, 2025] Wenqiang Wang, Yan Xiao, Hao Lin, Yangshijie Zhang, and Xiaochun Cao. Multi-task adversarial attacks against black-box model with few-shot queries. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13994–14014, 2025.

[Yang *et al.*, 2025] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[Ye *et al.*, 2023] Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. In *ICML*, pages 39818–39833. PMLR, 2023.

[Zuo *et al.*, 2025] Xinyue Zuo, Yan Xiao, Xiaochun Cao, Wenya Wang, and Jin Song Dong. Dt4lm: Differential testing for reliable language model updates in classification tasks. *IEEE Transactions on Software Engineering*, 2025.