Learning to Rank for In-Context Example Retrieval

Yuwen Ji,[1,2*] **Luodan Zhang,**[2*] **Ambyer Han,**[3*]
Haoran Que,[4] **Lei Shi,**[5] **Wang Chao,**[3] **Yue Zhang**[2†]

[1]Zhejiang University [2]Westlake University
[3]Amap, Alibaba Group [4]Peking University [5]Beihang University
{zhangluodan, zhangyue, jiyuwen}@westlake.edu.cn

Abstract

Recent advances in retrieval-based in-context learning (ICL) train the retriever using a classification objective, which categorizes in-context examples (ICEs) into the most useful and the rest based on absolute scores. However, during inference, ICEs are *retrieved by score ranking* rather than classification — The classification training objective deviates from this test scenario. Hence, in this paper, we propose a novel algorithm that trains a retrieval model by ranking formulation, where the preference rankings between ICEs are given by comparing the likelihood of the LLM generating the correct answer conditioned on each exemplar. By learning to rank, we motivate the retriever to automatically learn diverse rationales why specific examples are more useful for ICL decisions. This addresses the issue that classification models poorly capture broader utility. Experimental results demonstrate the top-1 performance of our proposal across 9 NLP tasks, with ablation studies and case studies further validating the effectiveness of our design. *The code can be found in:* `https://github.com/2022neo/SeDPO_NIPS25`

1 Introduction

Large Language Models (LLMs) [1, 2, 3] have shown their versatility in addressing diverse problems through in-context learning (ICL), which can be viewed as few-shot learning. ICL [4, 5, 6, 7] allows providing a few in-context examples (ICEs) to guide LLMs in generating predictions for test inputs without parameter updates. When there is a large set of labeled data, selecting the most useful few examples can improve ICL performance. To this end, existing methods [8, 9] fine-tune LMs as dense retrievers, typically in two steps: (1) Scoring a set of examples one by one or group by group using the ICL LLM; (2) Training the retriever model on scored data to align with the ICL LLM.

Dominant approaches train the retriever using a classification objective [10], categorizing ICEs into the most useful and the rest based on absolute scores. However, during inference, ICEs are *retrieved by score ranking* rather than classification — The classification training objective deviates from this test scenario, which leads to limitations. Taking math problems as an example, when there are no reference answers for a test query, recommending other useful ICEs, e.g., relevant formulas, will be more valuable. What we care about ultimately is not the absolute classification of examples, but the relative orders conditioned by the test input — which sets of examples are more useful.

We formulate in-context example retrieval as an information retrieval (IR) task by adopting a learning-to-rank (LTR) objective. LTR has been widely used in information retrieval [11], naturally capturing the broader utility across examples with ranking formulation. However, it remains underexplored in

[*]Equal contribution.
[†]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

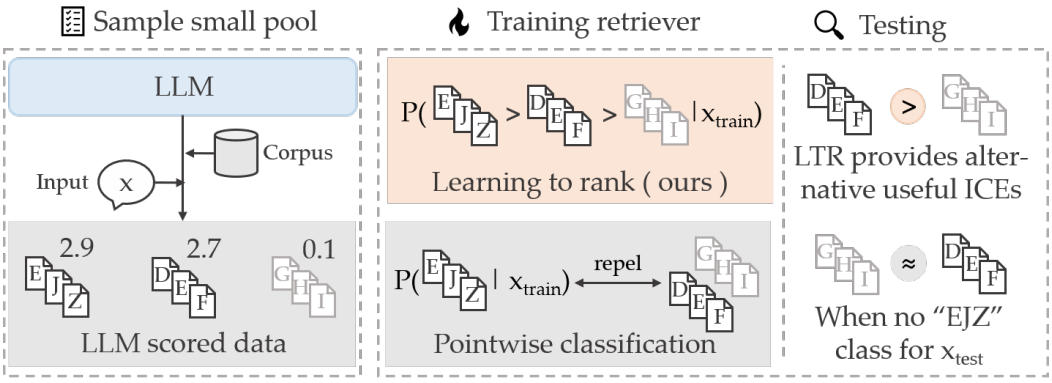

Figure 1: Motivating framework of existing work and our proposal (in red). E.g., when there are no reference answers (EJZ) for test input, LTR recommends alternative concepts or formulas (DEF).

ICL. As depicted in Figure 1, using partial order knowledge between scored examples, a retriever can retrieve a broader range of useful examples with ranking formulation, improving ICL performance.

To train the retriever, we propose a novel algorithm that aligns the preference ranking of ICEs, by comparing the likelihood of the LLM generating the correct answer conditioned on each exemplar. By LTR, we motivate the retriever to automatically learn various rationales why ICEs are more useful for ICL decisions. This addresses the issue that classification models poorly capture broader utility. To learn how to rank training examples, we adapt direct preference optimization [12] (**DPO**), integrating the **S**equential **E**xample relaxation [13], so as to derive a trainable pairwise ranking objective, while implicitly conforming to global preference order constraints. We thus name this algorithm **SeDPO**.

Experimentally, our method constantly ranks in top-1 across 9 NLP tasks, with an improvement of up to 18% over the best-performing classification-based baseline, achieving the SOTA results. Ablation study further confirms the usefulness of our ranking formulation and its complementary strength to existing paradigms. We summarize our key contributions as follows:

- We propose to learn preference ranking orders for ICL example retrieval.
- We introduce an extension of DPO for training retriever with pairwise ranking formulation.
- We demonstrate that SeDPO significantly outperforms existing state-of-the-art retrieval methods for in-context example retrieval.

2 Related work

Retrieval-based ICL. Two primary types of retrievers are commonly used for sample selection in ICL. The first type consists of off-the-shelf retrievers derived from heuristic criteria [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. However, the assumption that model performance is always correlated with heuristic criteria is not reliable. Consequently, other approaches train a retriever on a corpus using LLM's feedback so as to select ICEs that the LLM truly prefers. EPR [26] encodes and selects few-shot examples independently, with both queries and examples being encoded as vectors, enabling rapid retrieval through DPR (dense passage retrieval) [27] during inference. Following EPR's paradigm, CEIL [28] and UPRISE [29] are proposed to tackle various aspects of NLP tasks.

Observing that the above methods can overlook the interaction of examples that are used together as one few-shot prompt, the latest advancements achieve the SOTA performance by modeling the sequential order between examples in each prompt, such as RetICL [30], Active Example Selection [31], and Se^2 [13]. Among these, RetICL and Active Example Selection formalize the problem as Markov Decision Processes (MDP) and optimize models through reinforcement learning (RL). However, they suffer from training instability. In contrast, Se^2 models sequential order among examples in each prompt directly on the input text, achieving closed-loop optimization and the SOTA performance, making it the most suitable baseline in this class. Neither of these methods considers ranking orders of examples across prompts, which we address in this paper.

Recent ICL papers have considered aspects that are complementary to our work: (1) BESC [32] focuses on the internal ordering of ICEs within each prompt, using a contrastive loss to incrementally construct optimal sequences with dynamic lengths step-by-step. By contrast, our work emphasizes the ranking of different prompts. (2) CASE [33] prioritizes the efficiency of the selection process, framing it as a "top-m arm identification" problem with absolute training rewards. We, however, prioritize the quality of retrieved examples, formulating selection as a "learning to rank" problem with pairwise training rewards. (3) CLG [34] is a task-level selection method for few-shot scenarios, where scalability is critical—it selects a fixed set of examples for all test queries via gradient matching. Our work, by contrast, is optimized for few-shot scenarios where query-specific utility of each example is essential. We leave the exploration of these complementary aspects for future work.

Preference-aware ICL. Example retrieval methods considering the preference ranking orders of ICL examples are limited. UDR [35] additionally fits mini-batch rank indices with LambdaRank loss [36], which may conflict with global rank indices, as inserting a new scored example between two originally scored examples alters the local rank indices. This discrepancy amplifies as the example pool scales. Another line is RL-ICL [37], which develops a self-retrievable LLM with PPO [38] by modeling ICL performance as a reward signal. Due to differences in experimental and retrieval setups, we focus our comparison on mainstream retriever-based methods. A few ICL methods [39, 40] consider the preference order as an evaluation metric. In contrast, our method can efficiently learn the ranking orders of scored examples over the entire corpus.

3 Method

We first formalize the problem of ICL for LLM (Section 3.1), and subsequently use it to label ranking data (Section 3.2). Finally, we use the ranked data to train one different retriever (Section 3.3).

3.1 In-context learning with sequential example retrieval

In-context learning [41] is a key capability of LLMs where the model learns from a few examples provided in the input prompt to perform a task without parameter updates. Following the previous definition [13, 30], given a test sample (x, y), the LLM predicts $\hat{y}$ based on examples and input x as:

$$\hat{y} = \text{LLM}(e_K \oplus e_{K-1} \oplus ... \oplus e_1 \oplus x), \tag{1}$$

where e_k=$(x_k, y_k)_{k=1}^K$ is an example drawn from a corpus $\mathcal{C}$, consisting of an input-output pair. K is the shot number and $\oplus$ is the concatenation. The retrieval objective is to seek a set of examples in $\mathcal{C}$ for test input x, putting them into a sequential order $(e_1, ..., e_{K-1}, e_K) = e_{[1:K]}$ following [42], aiming to make $\hat{y}$ match the label y. All candidate K-shot sequences are denoted as $\mathcal{E} = \{e_{[1:K]}\}$. Notably, we specify in Eq. (1) that ordered examples are input right-to-left. As long as the ICL templates remain consistent during training and inference, the left-to-right input order is also supported.

Retriever model. We consider a class of methods based on dense passage retrieval (DPR) as our retriever model. DPR consists of a query encoder $\text{E}_{\text{input}}(\cdot)$ and an example encoder $\text{E}_{\text{example}}(\cdot)$, often initialized with a pretrained text-encoder such as BERT-base-uncased [43]. The retrieval score $\phi_{\text{retriever}}$ of example e for test input x conditioned on c is computed as $\text{sim}(e, x|c) = \text{E}_{\text{example}}(e)^\top \text{E}_{\text{input}}(c \oplus x)$.

Retrieval process. Following [13], we use the same beam search during inference to retrieve $e_{[1:K]}$ for fair comparison. We first encode and index all examples $e_k \in \mathcal{C}$ using the trained $\text{E}_{\text{example}}(\cdot)$. Given a test input x, we encode x with $\text{E}_{\text{input}}(\cdot)$ and retrieve w examples with the highest retrieval scores $\text{E}_{\text{example}}(\cdot)^\top \text{E}_{\text{input}}(\cdot)$. We set default beam size $w = 3$, the same as in [13]. We also draw $w = 1$ for brief illustration in Figure 2(c). These examples are then concatenated with the current inputs as new context sequences. The retriever scores are accumulated into the sequences' scores. This process is repeated, encoding inputs and seeking examples to maintain the w highest scoring candidate sequences until each sequence contains K examples. The sequence with highest accumulated scores is chosen as $e_{[1:K]}$. Motivated by left-to-right generation in autoregressive models, this retrieval process lets later retrieved ICEs observe previous retrieved ones, being aware of sequential relationships.

Training task. To convert LTR for ICL into a formal goal, we design our training task as follows: (1) A scoring function ϕ_{LLM} takes $(e_{[1:K]}, x, y)$ as input and evaluates the ICL performance for each $e_{[1:K]} \in \mathcal{E}$ through LLM. In this way, we obtain an ICL performance ranking conditioned on x. (2) Since the true answer y of a test input x is unavailable during inference, another scoring function

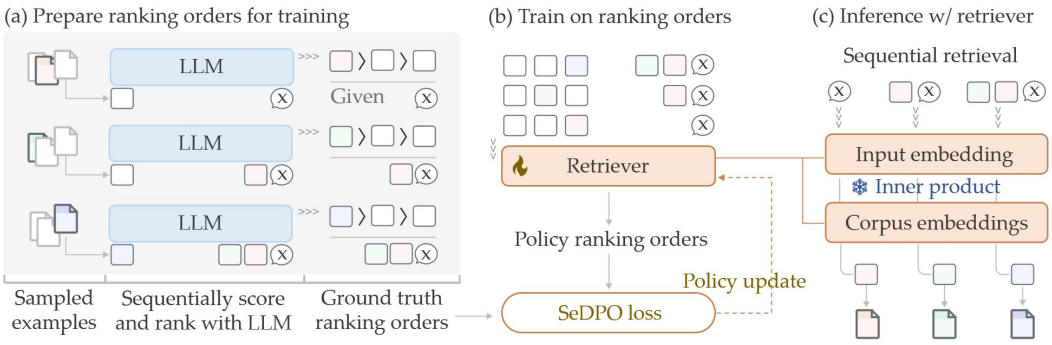

Figure 2: (a) Prepare preference data $\tilde{\mathcal{D}}$. (b) Train the retriever on $\tilde{\mathcal{D}}$ with SeDPO loss to align with partial order instead of the top-scored. (c) Inference with dense passage retrieval (DPR).

$\phi_{\text{retriever}}$ is introduced, which takes $(e_{[1:K]}, x)$ as input to score each $e_{[1:K]} \in \mathcal{E}$. This produces a retriever-based ranking conditioned on x. (3) To ensure that higher-scored retrieved $e_{[1:K]}$ can lead to better ICL performance, our goal is to train the retriever by aligning the $\phi_{\text{retriever}}$-based ranking with the ϕ_{LLM}-based ranking. Notably, previous methods widely use classification objective, categorizing $e_{[1:K]}$ with the best ICL performance as positive, and the rest as negative, which is unlike our goal.

3.2 Labeling data with ranking orders

Definition of ranking orders. We focus on Multiple-Choice Question (MCQ) tasks [44] by default. MCQ is a question format in which respondents (i.e., LLM) are asked to select only the correct answers from the choices offered as a list. How much the LLM's prediction $\hat{y}$ matches the ground truth $y \in \mathcal{Y}_{\text{gt}}$ can be measured by a task-specific scoring function $\mathrm{S}(\cdot,\cdot)$, which can be formulated:

$$\phi_{\text{LLM}}(e_{[1:K]}, x, y) = \mathrm{S}_{\text{MCQ}}(e_{[1:K]} \oplus x, y) = \frac{\mathrm{LH}(y|e_{[1:K]} \oplus x)}{\sum_{y' \in \mathcal{Y}} \mathrm{LH}(y'|e_{[1:K]} \oplus x)}, \tag{2}$$

where $\mathcal{Y}$ or $\mathcal{Y}_{\text{gt}}$ is output/ground-truth label space, LH is per token conditional likelihood of the LLM. For brevity, we denote $e_{[1:K]} \oplus x = (e_K \oplus ... \oplus e_1 \oplus x)$ w.r.t Eq. (1). As higher scores indicate that LLMs are more likely to output ground-truth answers, we can define partial order for ranking as $e^w_{[1:K]} \succ e^l_{[1:K]} | x$ if $\mathrm{S}_{\text{MCQ}}(e^w_{[1:K]} \oplus x, y) > \mathrm{S}_{\text{MCQ}}(e^l_{[1:K]} \oplus x, y)$ for all $y \in \mathcal{Y}_{\text{gt}}$ given x. The partial order can be a total order by weighted aggregating S_{MCQ} for all $y \in \mathcal{Y}_{\text{gt}}$; but we focus on the partial order as domain knowledge isn't required. In scored MCQ data, there may be multiple or no examples that guide LLMs in predicting correctly, requiring exploration beyond top-scored examples.

Scored data construction. Since the corpus $\mathcal{C}$ can form $(|\mathcal{C}|!)/(|\mathcal{C}| - K)!$ possible $e_{[1:K]}$, scoring all $e_{[1:K]}$ would be computationally prohibitive. Following [13], we employ a greedy algorithm to selectively score the data for fair comparison, ultimately obtaining sequentially scored data $\mathcal{D}$. The constructed queries in $\mathcal{D}$ are marked by tilde. For each $(x, y) \in \mathcal{C}$, we denote the test input x without ICE as $\tilde{x}_0$, and sample L examples from corpus $\mathcal{C}$ as $\mathcal{B}(\tilde{x}_0)$. We score the examples in $\mathcal{B}(\tilde{x}_0)$ with frozen LLM and Eq. (2), and repeatedly resample the examples in $\mathcal{B}(\tilde{x}_0)$ that cannot be ranked. Finally, we select an example from scored set $\mathcal{B}(\tilde{x}_0)$ as e_c based on its rank:

$$p(\text{rank}) = \frac{\exp(-\text{rank})}{\sum_{\text{rank}'=1}^{L} \exp(-\text{rank}')} \tag{3}$$

from which the higher-scored example is more likely to be selected and the diversity is preserved. $\tilde{x}_{k+1} = e_c \oplus \tilde{x}_k$ is iteratively updated for next round scoring, until the K-shot data are all constructed. The L scored examples in each $\mathcal{B}(\tilde{x}_k)$ are gathered with corresponding $\tilde{x}_k$ to form $\mathcal{D}$ (Fig. 2 (a)).

Ranked data construction. Given $\tilde{x}_k$, we have L scored examples, forming $\binom{L}{2}$ pairs of partial order. Empirically, we consider that two types of preference play a dominant role: (a) Given (x, y), examples in different pairs are non-overlapping, so as to enhance the diversity of training data. (b) e^w and e^l have a larger score margin, where their discrepancy is easier to learn. Therefore, we select T examples with the highest scores as the preferred ones and randomly match them with bottom-T dispreferred candidates, to construct sequential preference data $\tilde{\mathcal{D}}$ out of $\mathcal{D}$. We discuss it in ablation.

3.3 Algorithm for learning orders

We devise a novel RL algorithm for training a retriever, as directly applying existing RL algorithms such as DPO [12] on ranked data (Sec. 3.2) is prohibitive: (1) DPO requires knowing the probability distribution of specific retrieval actions over the entire corpus (i.e., policy), but the retriever model only outputs retrieval scores for specific ICEs; (2) optimizing retrieval actions over the entire corpus is very expensive and hard to scale. We innovatively address these challenges with (a) reparameterization of retriever score into policy model and (b) sequential relaxation of $e_k \in e_{[1:K]}$ in this section.

Policy of retriever model. To train a retriever with ranked data using DPO, we must convert the retriever score into a policy, to model the probability that retrieved ICEs are ranked in the order of ICL performance. We denote $\mathcal{C} \setminus \{e_i\}_{i<k}$ as the corpus excluding the selected ICEs. Based on retrieval scores $\mathrm{sim}(\cdot,\cdot|\cdot)$, the policy π of selecting $e_{[1:K]}$ from all candidates without replacement is:

$$\pi(e_{[1:K]}|x) = \prod_{k=1}^{K} \frac{\exp(\mathrm{sim}(e_k, x|\{e_i\}_{i<k}))}{\sum_{e_* \in \mathcal{C}\setminus\{e_i\}_{i<k}} \exp(\mathrm{sim}(e_*, x|\{e_i\}_{i<k}))} \tag{4}$$

Direct preference optimization (DPO) [12] is one of the most popular RL methods for preference ranking orders modeling. Instead of learning an explicit reward model, DPO reparameterizes the reward function r using a closed-form expression with the optimal policy. By initializing policy model π_θ and reference model π_{ref} with Eq. (4), the reward function r is as follows

$$r(x, e_{[1:K]}) = \beta \log \frac{\pi_\theta(e_{[1:K]}|x)}{\pi_{\mathrm{ref}}(e_{[1:K]}|x)} + \beta \log Z(x) \tag{5}$$

where β controls the KL-divergence constraint on policy/reference models, and $Z(x)$ is the partition function [12]. By incorporating this reward formulation into the BradleyTerry ranking objective, $P(e^w_{[1:K]} > e^l_{[1:K]}|x) = \sigma(r(x, e^w_{[1:K]}) - r(x, e^l_{[1:K]}))$, DPO expresses the probability of preference data with the policy model, yielding the following loss for triple $(x, e^w_{[1:K]}, e^l_{[1:K]})$:

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x, e^w_{[1:K]}, e^l_{[1:K]})} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(e^w_{[1:K]}|x)}{\pi_{\mathrm{ref}}(e^w_{[1:K]}|x)} - \beta \log \frac{\pi_\theta(e^l_{[1:K]}|x)}{\pi_{\mathrm{ref}}(e^l_{[1:K]}|x)} \right), \right] \tag{6}$$

where $e^w_{[1:K]}$ and $e^l_{[1:K]}$ denote the preferred and dispreferred prompts conditioned by input x. During training, we update the π_θ while freezing π_{ref}. By plugging Eq. (4) into Eq. (6), we have:

$$\begin{aligned}
\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{ref}) &= -\mathbb{E}_{(x, e^w_{[1:k]}, e^l_{[1:k]}) \sim \mathcal{D}} \left[\log \sigma \left(\beta \cdot f_\theta(x, e^w_{[1:K]}) - \beta \cdot f_\theta(x, e^l_{[1:K]}) - \beta \cdot (\gamma_w - \gamma_l) \right) \right] \\
f_\theta(x, e^j_{[1:K]}) &= \sum_{k=1}^{K} \left[\mathrm{sim}_\theta(e^j_k, x|\{e^j_i\}_{i<k}) - \mathrm{sim}_{\mathrm{ref}}(e^j_k, x|\{e^j_i\}_{i<k}) \right], \qquad j \in \{w, l\} \\
\gamma_j &= \sum_{k=1}^{K} \log \frac{\sum_{e_* \in \mathcal{C}\setminus\{e^j_i\}_{i<k}} \exp(\mathrm{sim}_\theta(e_*, x|\{e^j_i\}_{i<k}))}{\sum_{e_* \in \mathcal{C}\setminus\{e^j_i\}_{i<k}} \exp(\mathrm{sim}_{\mathrm{ref}}(e_*, x|\{e^j_i\}_{i<k}))}, \qquad j \in \{w, l\}
\end{aligned} \tag{7}$$

However, the estimation of the policy denominator (i.e., γ) necessitates repeated re-embedding of the entire corpus for each sampling step. This representation results in a prohibitively high computational overhead. To tackle this, we consider a relaxation by modeling the sequential relation among e_k within the same prompt $e_{[1:K]}$, motivated by left-to-right generation in autoregressive models.

Sequential LLM preference alignment. Under the sequential assumption, the ICL performance of the prompt given retrieved e_k is solely influenced by the performance of the retrieved ICEs, i.e., $e_{i<k}$. Therefore, the optimal $e_k \in e_{[1:K]}$ can be explored sequentially:

$$\max_{\{e_k\}_{k=1}^K} \mathrm{S}_{\mathrm{MCQ}}(e_K \oplus ... \oplus e_1 \oplus x, y) \Rightarrow \{e_k | \max_{e_k} \mathrm{S}_{\mathrm{MCQ}}(e_k \oplus ... \oplus e_1 \oplus x, y)\}_{k=1}^K \tag{8}$$

from which sequential preference data $\tilde{\mathcal{D}}$ can be constructed from $\mathcal{D}$ as:

$$\{(e^w_k \succ e^l_k)|e_{k-1} \oplus ... \oplus x\}_{k=2}^K \cup \{(e^w_1 \succ e^l_1)|x\} = \{(e^w_k \succ e^l_k)|\tilde{x}_{k-1}\}_{k=1}^K \tag{9}$$

where the conditional term is abbreviated as $\tilde{x}_{k-1}$. Based on this, the policy can be factorized sequentially into $\pi(e_{[1:K]}|x) = \prod_{k=1}^{K} \pi(e_k|\tilde{x}_{k-1})$. Here, $\pi(e_k|\tilde{x}_{k-1})$ is as follows:

$$\pi(e_k|\tilde{x}_{k-1}) = \frac{\exp(\text{sim}(e_k, \tilde{x}_{k-1}))}{\sum_{e_* \in \mathcal{C} \setminus \{e_i\}_{i<k}} \exp(\text{sim}(e_*, \tilde{x}_{k-1}))} \tag{10}$$

by plugging Eq. (10) into Eq. (6), we obtain same form as Eq. (7) with $\gamma_w - \gamma_l = 0$:

$$\begin{gathered}\mathcal{L}_{\text{SE-DPO}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(\tilde{x}_{k-1}, e_k^w, e_k^l) \sim \tilde{\mathcal{D}}} \left[\log \sigma \left(\beta \cdot f_\theta(\tilde{x}_{k-1}, e_k^w) - \beta \cdot f_\theta(\tilde{x}_{k-1}, e_k^l) - \beta \cdot 0\right)\right], \\ f_\theta(\tilde{x}_{k-1}, e_k^j) = \text{sim}_\theta(e_k^j, \tilde{x}_{k-1}) - \text{sim}_{\text{ref}}(e_k^j, \tilde{x}_{k-1}), \qquad j \in \{w, l\}\end{gathered} \tag{11}$$

The new objective function is tractable by implicitly considering the global partition function (i.e., the denominator in Eq. (10)). Training signals (e.g., e_k^w,e_k^l) can be readily generated from scored data. We train the retriever on $\tilde{\mathcal{D}}$ with Eq. (11). The resulting retriever can discern the preferred ICEs for varying context input. Note that the partial order learned by our bi-encoder satisfies the transitivity: if $e^a \succ e^b|x$ and $e^b \succ e^c|x$, then $e^a \succ e^c|x$, ensuring our retriever is ranking-informative. The proof of the transitivity can be found in Appendices.

4 Experiments

We take the SOTA method Se^2 [13] as our base, which is a classification-based method, and implement SeDPO on top of it. We first compare the ICL performance with SOTA retrievers (main results); validate key components in SeDPO (ablation); then provide extra experiments for deeper analysis.

4.1 Experimental settings

Task and dataset. We use a total of 9 tasks across 4 distinct categories, including Paraphrase: **MRPC** [45], **PAWS** [46], **QQP** [47]; Coreference: **WSC** [48]; Reading: **MultiRC** [49], **BoolQ** [50], **AGNews** [51]; NLI: **MNLI-m/mm** [52]. The preprocessing and evaluation for all datasets are the same as Se^2. The description of each dataset is provided in the appendices.

Implementation details. For fair comparison, we follow the hyperparameter setting of Se^2 and use GPT-Neo-2.7B [53] as the scoring and inference LLM for most of the experiments. The encoders in retriever model are initialized with "BERT-base-uncased" [43]. The shot number $K = 3$. We set $T = 20$, ensuring that $\tilde{\mathcal{D}}$ is sourced from the same training data for fairness; retriever is fine-tuned for 6 epochs on each category, the best checkpoint is chosen based on retrieval accuracy of validation set, and evaluated using task-specific metric on the test set. Refer to appendices for more details.

Additionally, Se^2 incorporates two data augmentation techniques: (1) positive chosen — Se^2 reserves only the data where the selected representative examples can guide LLM in correctly predicting; (2) in-batch rejection — in each training batch, the negative sample set for each input is extended with examples from other inputs for diversity. Following Se^2, we reserve the data where e^w guides LLM in correctly predicting; for each input, the rejected instance is uniformly sampled from combined set of original e^l and examples from other inputs. We analyse the techniques in ablation study.

Baselines. All model comparisons are fair using the same size, as detailed in the appendices. We repeat each algorithm 10 times and report the average performance. The off-the-shelf baselines include: **Random** involves sampling K demonstrations randomly; **BM25** [21] is the commonly used sparse retriever that finds exemplars based on textual similarity; **SBERT** [23] is a dense retriever by computing sentence embedding, we take "paraphrase-mpnet-base-v2" as its encoder.

For fair comparison, we re-trained the learning-based baselines aligning our task settings (e.g., with the same scored dataset): **UPRISE** [29] estimate the usefulness of each example separately; **Se^2** [13] improves upon UPRISE by sequentially retrieving representative examples, making it the most competitive and accessible alternative for comparison; **UDR** [35] uses LambdaRank loss to inject ranking information, while estimating the usefulness of each example separately. We also study the integration of UDR's LambdaRank loss with Se^2 in the ablation experiments.

Table 1: Main results on various tasks. The **best results** and the second-best are highlighted. The *Avg.* of all metrics for tasks within the same category with significant improvements is marked by ↑.

	Paraphrase						***Coreference***
	MRPC		**PAWS**	**QQP**		*Avg.*	**WSC**
	acc	f1	acc	acc	f1		acc/*Avg.*
Zeroshot	46.1±0.0	45.3±0.0	51.8±0.0	48.4±0.0	42.1±0.0	46.7±0.0	59.6±0.0
Random	66.8±3.0	79.5±4.1	50.1±3.8	40.6±4.8	50.9±7.8	57.6±3.8	48.3±8.2
BM25	57.8±0.0	69.1±0.0	48.9±0.0	54.8±0.0	55.4±0.0	57.2±0.0	52.4±0.0
SBERT	56.4±0.0	66.9±0.0	49.4±0.0	51.2±0.0	56.2±0.0	56.0±0.0	46.2±0.0
UDR	65.9±4.6	75.4±3.5	51.8±1.2	74.1±1.9	67.9±2.4	67.0±1.2	52.0±4.7
UPRISE	74.0±0.8	83.3±0.1	49.1±0.0	71.0±1.0	69.8±0.1	69.4±0.2	46.5±2.2
Se^2	77.6±0.4	85.4±0.3	54.7±0.1	75.5±0.1	72.8±0.0	73.2±0.2	55.1±0.9
SeDPO	**77.9±0.9**	**85.6±0.2**	**73.0±2.9**	**77.6±0.6**	**75.0±0.2**	**77.9±0.6**↑	**62.5±0.2**↑

	Reading				***Natural Language Inference (NLI)***		
	MultiRC	**BoolQ**	**AGNews**	*Avg.*	**MNLI-m**	**MNLI-mm**	*Avg.*
	f1	acc	acc		acc	acc	
Zeroshot	57.1±0.0	54.6±0.0	38.4±0.0	50.0±0.0	35.2±0.0	36.4±0.0	35.8±0.0
Random	57.7±2.5	54.8±6.7	25.8±1.1	46.1±1.2	34.2±3.0	34.9±3.9	34.6±1.6
BM25	46.5±0.0	60.3±0.0	81.7±0.0	62.8±0.0	35.3±0.0	35.6±0.0	35.5±0.0
SBERT	49.3±0.0	58.1±0.0	84.7±0.0	64.0±0.0	37.3±0.0	37.3±0.0	37.3±0.0
UDR	55.3±3.1	54.6±1.9	88.5±1.0	66.1±0.9	62.7±1.5	65.0±1.3	63.8±1.4
UPRISE	55.4±0.2	61.5±0.1	90.6±0.8	69.2±0.1	68.5±0.1	70.3±0.3	69.4±0.2
Se^2	52.1±2.3	63.6±0.2	90.8±0.3	68.8±0.7	69.4±0.2	70.4±0.1	69.9±0.2
SeDPO	**60.3±0.4**	**64.6±1.7**	**91.0±0.2**	**72.0±0.6**↑	**70.6±0.1**	**72.0±0.3**	**71.3±0.2**↑

4.2 Main results

Table 1 details the experimental results on MCQ tasks, where generative LLMs are known to need improvement [29]. On each task, we mark the best results in bold and underline the second-best. The *Avg.* column represents the mean performance for each category, with significant improvements marked by ↑ (confidence level is 99%) over the best alternative. SeDPO outperforms all other methods in all categories with an avg improvement of up to 4.7%, and we also have several findings.

First, random sampling does not lead to sizable gains compared to zeroshot. In contrast, BM25 and SBERT have a significant gain over random sampling and zeroshot. This demonstrates the necessity of providing related examples for LLMs in downstream tasks. In addition, finetuning-based retrievers perform better than off-the-shelf retrievers, highlighting the effectiveness of using LLM feedback.

Second, among the finetuning-based retrievers, Se^2 outperforms UPRISE by using sequential relationships between examples, showing the importance of capturing sequential relationships between examples. Though UDR considers local ranking regularization, the performance gain is limited on average compared to Se^2. In particular, our proposed SeDPO achieves the best performance in all categories, significantly exceeding Se^2 by at least 1.4% to a maximum of 7.4%. This demonstrates that learning preference orders rather than the representative examples, significantly enhances the ICL performance on MCQ tasks and complements previous advancements.

Third, for downstream tasks such as PAWS, WSC, and MultiRC, SeDPO achieves a substantial improvement of up to 18.3%. We speculate that retrievers trained to learn representative patterns fail to capture the true preferences of LLMs on challenging data, as they show no significant improvement compared to zero-shot. Particularly, Se^2 underperforms random selection in both MultiRC and WSC tasks, indicating that these tasks require the capture of more diverse query-dependent ICE patterns. Se^2 learns from top-scored ICEs, thus insufficient to generalize to test input. In contrast, SeDPO consistently outperforms random selection by 2.9% to 3.9% and surpasses Se^2 by 7.4% to 8.2%. This demonstrates the superiority of our proposal in challenging tasks.

4.3 Ablation studies

We compare our original framework with its variants altering each time a different component.

Table 2: Ablation results on *Paraphrase* task using GPT-Neo-2.7B, trained on 3-shot.

	Paraphrase
SeDPO ($\beta = 0.02$)	77.9
w/o positive chosen	77.9
w/ top-1 chosen	70.8
w/o in-batch rejection	73.9
w/ random preference	57.6
w/ LambdaRank (UDR)	72.9
w/ RoBERTa	85.7
SeDPO ($\beta = 0.02$) $\circ$ Se^2	79.0
Se^2 $\circ$ SeDPO ($\beta = 0.02$)	74.8

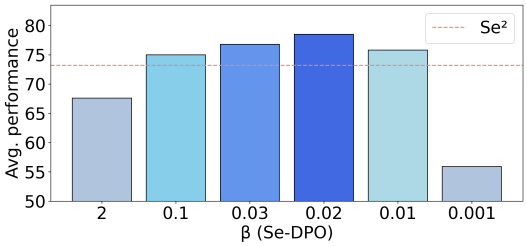

Figure 3: Performance of SeDPO on *Paraphrase* category using GPT-Neo-2.7B with different β. The dashed line represents the results of **Se^2**.

Table 3: The average textual/semantic diversity of selected ICEs, as well as the average performance when the input order of ICEs is randomized. We take the main results on *Paraphrase* as our base.

	SeDPO	**Se^2**	**UPRISE**	**UDR**	**SBERT**	**BM25**	**Random**
Textual Diversity	53.3%	49.0%	46.7%	54.3%	49.7%	46.0%	**61.4%**
Semantic Diversity	40.7%	39.0%	37.3%	40.3%	25.3%	29.0%	**46.0%**
Random order (Best of 5)	78.2%	73.5%	70.9%	68.4%	57.3%	58.1%	-
Random order (Worst of 5)	77.1%	72.5%	68.6%	66.3%	55.1%	57.0%	-

Complementary strength. We test the complementary strength of SeDPO and Se^2. In Table 2, SeDPO (β=0.02) $\circ$ Se^2 denotes initializing the retriever's weights of Se^2 with weights of trained SeDPO, and vice versa. Training Se^2 model using SeDPO's trained weights can enhance ICL performance. We surmise that the greedy data construction may not be globally optimal for fully learning preference ranking order, there is still room for improvement. Conversely, initializing SeDPO with Se^2's weights leads to suboptimal results. This asymmetry likely arises from SeDPO's inherent KL-divergence constraint, which preserves the retriever's pretrained knowledge base. Se^2 may overfit to top-scored, hindering SeDPO's ability to use the retriever's pretrained knowledge.

Effect of components. As shown in Table 2, replacing our original preference data with randomly selected pairs of the same size led to poor results (*w/ random preference*). Since $T \ll \binom{L}{2}$, random local observations struggle to learn reliable partial orderings. Furthermore, using local representative examples (*w/ top-1 chosen*) as the only chosen in preference data and pairing them randomly with bottom-T examples, while performing better than random selection, still falls short of our original design. This demonstrates the effectiveness of reserving diverse (e^w, e^l) with a larger score margin.

Notably, we also investigate the impact of the two data augmentation techniques used in Se^2 (detailed in Sec. 4.1). The *positive chosen* does not enhance our results, indicating that learning orders rely more on the discrepancy between chosen/rejected samples rather than the absolute quality of the chosen ones. In contrast, *in-batch rejection* increases data diversity and improves our results by 4%.

Influence of embeddings. This paper uses "BERT-base-uncased" as the encoder model for fair comparison. Nowadays there are many better text embedding models available. To further show the influence of the embeddings, we replaced the retriever backbone of SeDPO with "RoBERTa-base" [54], which is known to outperform "BERT-base-uncased" [43] across a range of NLP tasks. The results are shown as *w/ RoBERTa* in Table 2, SeDPO benefits from stronger embeddings as expected.

Diversity of retrieved examples. Table 3 presents the diversity metrics corresponding to the main results in Table 1, where higher values indicate lower query-similarity and a richer context of retrieved ICEs. Both SeDPO and UDR achieve high diversity in selected ICEs, exceeding other retrievers (except random) by at least 3.6%/1.3% in textual/semantic diversity. This demonstrates that incorporating ranking signals enhances diversity. The poor ICL performance of randomly selected ICEs highlights the importance of selecting relevant ICEs. In addition, though UDR's diversity is comparable to SeDPO's, its ICL performance lags behind Se^2 and UPRISE. This indicates that SeDPO better trades off diversity with ICL utility and successfully retrieves *diverse yet useful* ICEs.

Table 4: Transferability on shot number and model size. The average performance of *Paraphrase*.

Inference Model	Method	1-shot	3-shot	6-shot	9-shot	12-shot	15-shot	Average
GPT-2-XL-1.5B (0-shot=39.6)	BM25	57.6	58.5	58.8	58.7	59.3	60.1	58.8
	SBERT	57.9	57.5	59.0	59.6	58.6	58.3	58.5
	UPRISE	69.2	69.4	69 .8	69.8	70.0	70.2	69.7
	Se^2	73.9	72.9	72.9	72.8	72.8	72.7	73.0
	SeDPO	75.0	78.9	79.5	79.2	79.0	79.2	**78.5**
GPT-Neo-2.7B (0-shot=46.7)	BM25	57.1	57.2	58.9	59.5	59.0	59.4	58.5
	SBERT	56.6	56.0	59.4	58.9	59.8	58.4	58.2
	UPRISE	69.4	69.7	69.5	69.2	69.2	69.3	69.4
	Se^2	73.5	73.2	73.1	73.0	72.8	72.6	73.0
	SeDPO	77.6	77.9	78.0	77.9	78.2	78.1	**78.0**
Llama3-8B-Instruct (0-shot=56.4)	BM25	68.6	73.2	74.7	75.1	75.6	76.7	74.0
	SBERT	68.3	73.0	73.4	75.1	75.4	76.1	73.5
	UPRISE	70.9	75.3	76.4	76.6	76.9	77.0	75.5
	Se^2	71.9	76.7	78.0	78.0	77.9	77.9	76.7
	SeDPO	71.9	77.4	78.5	79.3	80.2	80.3	**77.9**
Llama3.3-70B (0-shot=67.6)	BM25	78.4	80.7	82.2	81.7	81.8	82.2	81.2
	SBERT	78.7	80.3	81.2	81.7	81.7	82.7	81.1
	UPRISE	77.3	80.2	81.3	80.5	80.8	81.3	80.2
	Se^2	77.9	81.0	82.0	82.2	81.9	81.9	81.1
	SeDPO	78.6	81.0	82.3	82.9	83.2	83.2	**81.9**

Impact of ICE ordering. We empirically analyze the impact of randomizing the input order of ICE on paraphrase performance. The results in Table 3 suggest that randomizing the ICE order exhibits limited potential for enhancing sequential approaches. We attribute this to that SeDPO and Se^2 already optimize the input order of ICEs to an extent by sequential example selection.

Impact of β. Table 3 shows the *Paraphrase* performance of SeDPO using GPT-Neo-2.7B with different β. Too large or too small β can lead to a dominant or negligible constraint of KL divergence, resulting in performance degradation. SeDPO shows promising improvements when β ranges from 0.01 to 0.1, so we tune β between 0.001 and 2 across different categories.

Impact of T. Table 12 in **Appendices** shows the extra ablation results for different T, following the setup of Table 2. As T increases, the performance of SEDPO improves because a broader preference ranking is learned. However, when T is too large, the performance gain decreases due to more low-confidence LLM rankings, leading to the same conclusion as in Table 2 (*w/ random preference*).

4.4 Analysis

Transferability. As LLM scales, aligning the LLM preference under different shot numbers K of ICEs is time-consuming and resource-intensive. We thus explore the effectiveness of retrievers, as LLMs and example numbers vary. Specifically, in our main experiments (Table 1), we trained retrievers for each category using 3-shot data and GPT-Neo-2.7B. We then evaluated these retrievers in unseen inference settings, where the K varied from 1 to 15, and the inference LLMs included GPT-2-XL [55], GPT-Neo-2.7B, Llama3-8B-Instruct and Llama3.3-70B [2]. Table 4 illustrates that SeDPO consistently outperforms baseline retrievers in all settings. Notably, though SeDPO is trained on 3-shot, its performance improves as the K increases and significantly outperforms 0-shot by at least 10% on all LLMs. We also find that as the model size of LLMs increases, the gap between different ICL methods decreases. This indicates that larger LLMs are smarter and can reason using suboptimal examples; on the other hand, smaller LLMs (not that smart) rely more on high-quality examples, as is also mentioned in [56]. However, the in-context examples provided by SeDPO are useful for both small and large LLMs, demonstrating strong transferability.

Math inference. We compare SeDPO and Se^2 with several reasoning-focused retrievers [57, 58, 28] on AUQA [59], a MCQ task requiring math inference. Table 9 in **Appendices** shows that SeDPO outperforms Se^2, aligning with our main discovery. Though RGER [57] is built for reasoning-focused tasks, SeDPO surpasses RGER [57] without being geared toward reasoning capabilities.

More in-depth analysis. To further study the performance boundaries of SeDPO, we provide extra results in **Appendices**, including 0-shot performance on human-labeled data, impact of preference

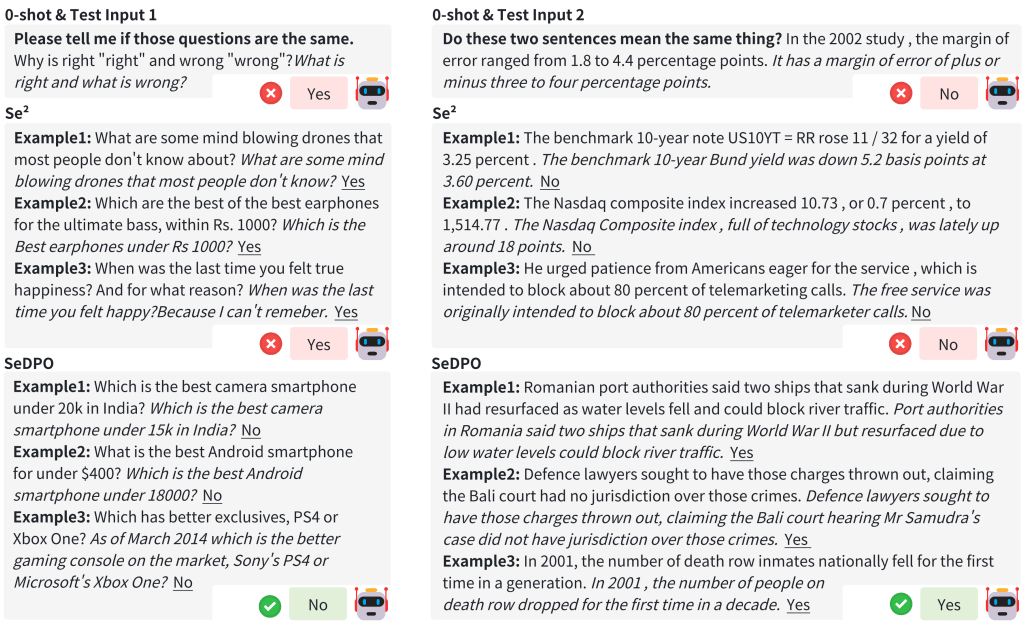

Figure 4: Two cases on *Paraphrase* where SeDPO helps LLM correctly infer, but Se^2 does not.

dataset construction using various LLMs, detailed math inference, analysis of full list order, proof of transitivity, cost of constructing training data, discussion of Eq. (2) in open-ended QA setting.

Case study. In Figure 4, we analyze two cases from the *Paraphrase*, to intuitively compare the effectiveness of SeDPO and Se^2. The answers of the examples are marked with underscores. Specifically, the task focuses on whether two sentences in the test input are synonymous. Se^2's examples emphasize surface patterns: in the left case, the two sentences have similar word compositions; in the right case, there are changes in percentage numbers. In contrast, SeDPO captures the task-related nuances preferred by LLMs: it selects examples based on the extent of difference between the two sentences and marginalizes causal information using broader contexts such as historical, legal, and social. This demonstrates how learning preference ranking orders gives a broader causal relationship between examples, which improves the ICL performance. More cases can be found in the appendices.

5 Conclusion

We considered learning to rank for in-context example retrieval, introducing SeDPO, a simple yet effective method. Unlike dominant methods that focus on representative examples, SeDPO captures the global preference orders through a pairwise ranking formulation. We effectively address the issue that classification-based retrievers poorly capture broader utility. Extensive experiments demonstrate our superiority. **Additional experiments, discussions, and proofs** appear in the Appendices.

Limitations. Our research focuses on improving retriever training but shares existing frameworks' structural limitations. First, we mainly use GPT-Neo-2.7B for fair comparison, where shot number analysis is constrained by sequence length; this can be improved by recent input/prompt compression [60]. Second, results are affected by inherent biases [61] in retriever models and LLMs, requiring fair and interpretable strategies (a promising direction). Third, permutation-based example retrieval is underexplored. Notably, SeDPO needs dominant preference identification to reduce computational costs and redundant interference; while clear for MCQ tasks, open-ended question challenges remain.

Broader Impacts. Learning-to-rank for ICE retrieval boosts LLM's ICL performance but carries negative societal risks. Without safeguards, it may retrieve/prioritize biased, misleading, or harmful examples—reinforcing unfair decisions (e.g., employment/legal consultation) or spreading disinformation—and could be misused to get adversarial examples manipulating LLMs. Mitigation requires rigorous fairness audits, retrieved example filters, and controlled access for high-stakes deployments.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) Key Program under Grant Number 62336006, the National Key R&D Program of China (2021YFB3500700), and the Beijing Science and Technology Plan Project.

References

[1] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[3] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[5] Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*, 2024.

[6] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024.

[7] Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2025.

[8] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*, 2024.

[9] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[10] Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*, 2024.

[11] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2023.

[12] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. se^2: Sequential example selection for in-context learning. *arXiv preprint arXiv:2402.13874*, 2024.

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

[15] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*, 2022.

[16] Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*, 2022.

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

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

[19] Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. On the relation between sensitivity and accuracy in in-context learning. *arXiv preprint arXiv:2209.07661*, 2022.

[20] Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*, 2023.

[21] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

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

[23] N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[24] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

[25] Young-Jun Lee, Chae-Gyun Lim, and Ho-Jin Choi. Does gpt-3 generate empathetic dialogues? a novel in-context example selection method and automatic evaluation metric for empathetic dialogue generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 669–683, 2022.

[26] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

[27] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[28] Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR, 2023.

[29] Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Denvy Deng, and Qi Zhang. Uprise: Universal prompt retrieval for improving zero-shot evaluation. *arXiv preprint arXiv:2303.08518*, 2023.

[30] Alexander Scarlatos and Andrew Lan. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*, 2023.

[31] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*, 2022.

[32] Xiang Gao, Ankita Sinha, and Kamalika Das. Learning to search effective example sequences for in-context learning. *arXiv preprint arXiv:2503.08030*, 2025.

[33] Kiran Purohit, V Venktesh, Sourangshu Bhattacharya, and Avishek Anand. Sample efficient demonstration selection for in-context learning. *arXiv preprint arXiv:2506.08607*, 2025.

[34] Jianfei Zhang, Bei Li, Jun Bai, Rumei Li, Yanmeng Wang, Chenghua Lin, and Wenge Rong. Selecting demonstrations for many-shot in-context learning via gradient matching. *arXiv preprint arXiv:2506.04579*, 2025.

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

[36] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[37] Quanyu Long, Jianda Chen, Wenya Wang, and Sinno Jialin Pan. Large language models know what makes exemplary contexts. *arXiv preprint arXiv:2408.07505*, 2024.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[39] Marie Al Ghossein, Emile Contal, and Alexandre Robicquet. Iclerb: In-context learning embedding and reranker benchmark. *arXiv preprint arXiv:2411.18947*, 2024.

[40] Feifan Song, Yuxuan Fan, Xin Zhang, Peiyi Wang, and Houfeng Wang. Icdpo: Effectively borrowing alignment capability of others via in-context direct preference optimization. *arXiv preprint arXiv:2402.09320*, 2024.

[41] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[42] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[43] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[44] Nikitas Karanikolas, Eirini Manga, Nikoletta Samaridi, Eleni Tousidou, and Michael Vassilakopoulos. Large language models versus natural language understanding and generation. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, PCI '23, page 278–290, New York, NY, USA, 2024. Association for Computing Machinery.

[45] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[46] Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[47] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[48] Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. A hybrid neural network model for commonsense reasoning. In Simon Ostermann, Sheng Zhang, Michael Roth, and Peter Clark, editors, *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 13–21, Hong Kong, China, November 2019. Association for Computational Linguistics.

[49] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, 2018.

[50] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

[51] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[52] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[53] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[54] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

[56] Liang Wang, Nan Yang, and Furu Wei. Learning to retrieve in-context examples for large language models. In *EACL 2024*, July 2023.

[57] Yukang Lin, Bingchen Zhong, Shuoran Jiang, Joanna Siebert, and Qingcai Chen. Reasoning graph enhanced exemplars retrieval for in-context learning. *arXiv preprint arXiv:2409.11147*, 2024.

[58] Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, et al. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. *arXiv preprint arXiv:2310.02954*, 2023.

[59] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

[60] Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024.

[61] Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. *arXiv preprint arXiv:2305.19148*, 2023.

[62] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[63] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024.

[64] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention the limitations of the work in "Conclusion" section and provide more details in the appendices.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Proofs related to our theoretical results can be found in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce our main results can be found in the "Experiments" section or the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Data, code, and instructions are all available in the supplemental material. We will open-source all materials when the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all the training and test details in the "Experiments" section; more details can be found in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the statistical significance of the main comparison in the "Experiments" section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Full information on the computer resources is available in the supplemental materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully followed the Ethics Guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mention the impacts of the work in "Conclusion" section and provide more details in the appendices.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We conduct experiments using compliant released data/models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer to the supplemental materials.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the assets introduced in the paper in the supplemental materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We describe the usage of LLMs in experiments and provide more details in appendices.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendices

.1 Datasets details

- **MRPC**: A paraphrase task with 3,668 training examples and 408 test examples, evaluated using Accuracy and F1.
- **PAWS**: A paraphrase task with 49,401 training examples and 8,000 test examples, evaluated using Accuracy.
- **QQP**: A paraphrase task with 363,846 training examples and 40,430 test examples, evaluated using Accuracy and F1.
- **WSC**: A coreference task with 554 training examples and 104 test examples, evaluated using Accuracy.
- **MultiRC**: A reading comprehension task with 27,243 training examples and 4,848 test examples, evaluated using F1.
- **BoolQ**: A reading comprehension task with 9,427 training examples and 3,270 test examples, evaluated using Accuracy.
- **AGNews**: A reading comprehension task with 120,000 training examples and 7,600 test examples, evaluated using Accuracy.
- **MNLI-m/mm**: A natural language inference task with 392,702 training examples and 9,815/9,832 test examples for m/mm, evaluated using Accuracy.

We list the detailed datasets' statistical information above. To convert datasets into natural language instructions, we follow previous practice [13] using the instruction template of FLAN [62]. Each task dataset corresponds to approximately seven templates. All datasets are publicly available under open licenses (e.g., CC-BY, CC-BY-SA, or research-only terms). Datasets are all for MCQ tasks and are widely used in relevant work without offensive content, which is in line with our purpose of use.

.2 Model details

- **BERT-base-uncased** has approximately 110 million parameters and is released under the Apache License 2.0.
- **GPT-Neo-2.7B** consists of 2.7 billion parameters and is distributed under the MIT License.
- **Llama3-8B-Instruct** features 8 billion parameters and is licensed under the Meta Llama 3 Community License.
- **Llama3.3-70B** contains 70 billion parameters and is also governed by the Meta Llama 3 Community License.

The documentation for the artifacts is publicly available, refer to their citations in main paper.

.3 Implementation details

Table 5 lists the overall hyperparameters. For fair comparison, we follow the hyperparameter setting of Se^2: we use GPT-Neo-2.7B [53] as the scoring and inference LLM for most experiments; both encoders were initialized with "BERT-base-uncased" [43]; up to 10k data points are randomly selected for each task to construct the training data and example pool for each category while maintaining class balance in classification tasks; sample size L is set to 50 by default, depending on the configuration of Se^2; the shot number K=3; retriever is fine-tuned for 6 epochs on each category, the best checkpoint is chosen based on retrieval accuracy using the validation set, and evaluated using task-specific metric on the test set. For the hyperparameters of SeDPO, we set T=20 and ensure that our preference data are sourced from the training data of Se^2 for data fairness. β takes values between 0.001 and 2.

Number of samples per batch. Note that in mini-batch training, for each top-scored positive example considered by Se^2, it is necessary to simultaneously consider T low-scored negative examples and T negative examples randomly sampled from the corpus, as shown in parentheses of # of samples per batch for Se^2 in Table 5. SeDPO generates T positive and negative example pairs for each input, but each positive example only needs to consider one negative example, as shown in parentheses of the number of samples per batch for SeDPO in Table 5. We trained the retriever with 8 threads in a

Table 5: Hyperparameter settings.

Hyperparameter	Assignment	Hyperparameter	Assignment
shot-number	3	Max sequence length	512 for retriever
Optimizer	Adam		2048 for LLMs
Number of epochs	6	per GPU Max learning rate	1e-5
Preference β	> 0.001	# of samples per batch for SeDPO	8*32*(1+1)
	< 2	# of samples per batch for Se^2	1*32*(1+2*T)
Adam epsilon	1e-8	Warmup steps	1000
Adam beta weights	0.9, 0.999	Learning rate decay	linear
Weight decay	0.0	Learning rate scheduler	warmup linear

data-distributed manner on 8*A100-80GB. To speed up the training process of Se^2, we considered 32 positive examples in each mini-batch, along with their dependent negative examples, resulting in a total of 32*(1+2*T) examples used per mini-batch. Since SeDPO needs to allocate additional memory to the reference model, we only consider using 8*32*(1+1) examples per mini-batch in SeDPO. The number of samples per batch can be adjusted according to the experimental environment.

Since the training algorithm does not alter the model architecture, the total number of parameters remains 220M, consistent with Se^2. Briefly put, our design is compatible with existing fine-tuning-based retrievers using DPR[27] or its sequential version, no further inference or data load is introduced. SeDPO leaves the original framework's token flux unchanged [13], each task takes about 7/9 hours in the scoring/training phase. To allay concerns that the improved ICL performance might stem from differences in backbone models, we detail settings of related methods in Table 6 for reference only.

Table 6: The setting of the related methods.

	Finetuned	Retriever (Number of Parameters)	Scoring LLM
BM25	×	×	×
SBERT	×	paraphrase-mpnet-base-v2 (1*109)	×
UPRISE	✓	2*BERT-based-uncased (2*110M)	GPT-Neo-2.7B
Se^2	✓	2*BERT-based-uncased (2*110M)	GPT-Neo-2.7B
SeDPO	✓	2*BERT-based-uncased (2*110M)	GPT-Neo-2.7B

.4 Diversity calculation

Textual diversity in Table 3 is measured by Levenshtein Edit distance:

$$div_{\text{textual}}(s_1, s_2) = \frac{dist(s_1, s_2)}{sum(len(s_1), len(s_2))} \tag{12}$$

where s_1 and s_2 are two sentences, $dist(\cdot, \cdot)$ is Levenshtein Edit distance, $len(\cdot)$ denotes the number of characters in sentence. Semantic diversity in Table 3 is measured by SBERT [23]:

$$div_{\text{semantic}}(s_1, s_2) = \frac{1 - cos(E(s_1), E(s_2))}{2} \tag{13}$$

where $E(\cdot)$ is the sentence embedding encoded by SBERT and $cos(\cdot, \cdot)$ is the cosine similarity between two embeddings. We take "paraphrase-MiniLM-L6-v2" as the encoder.

Table 7: The co-reference performance using preference data generated by different LLMs.

	GPT-Neo-2.7B	Llama3-8B-Instruct	DeepSeek-R1
SeDPO	62.5%	59.6%	53.8%
Se^2	55.1%	54.8%	53.8%

.5 Analysis of preference labeling

It will be valuable to study the reliability of the preferences generated by artificial intelligence and their impact on performance. To this end, we conduct comparative experiments on dataset

construction using various LLMs in two settings: (a) To study the reliability of the likelihood-based scoring method, we construct training data through Llama3-8B-Instruct [2] using Eq. (2). (b) To compare with the prompt-engineer-based scoring method, we use DeepSeek-R1 as an agent to simulate a human annotation pipeline following Table 8. The overall results on the co-reference task are shown in Table 7. Specifically, to align with the results in Table 1, the test time LLM is still GPT-Neo-2.7B. The generalization ability of the retriever on different test LLMs has been analyzed in Table 4. The results show that imitating human sorting cannot super-enhance ICL performance. We recommend using the output probability of the LLM for the desired answer to rank ICEs.

Table 8: Prompt used for R1 ranking.

Prompt
You have a question inside <question> tags, and you have a correct answer inside <answer> tags. Your task is to determine which examples inside <demonstrations> tags are more conducive to obtaining the correct answer to the question. <question>{#question}<\question> <answer>{#answer}<\answer> <demonstrations>#demonstrations<\demonstrations> In the tags, the examples are arranged in the form of "ID: example". You can sort all the examples based on their usefulness and return their sorted IDs in the form of a number list that can be parsed by JSON in the <output> tags. More useful examples should be at the front of the list.

Discussion on full list order. Recent advances [63] in Reinforcement Learning from Human Feedback show the potential of modeling full list order. The policy in this kind of method requires estimating the partition function by knowing all actions. For LLMs, the partition function means being able to observe the entire vocabulary to model the probability of the next word, and this is achievable. However, applying such a formulation to retrieval is substantially more challenging — it involves re-embedding the entire corpus at each sampling step, which leads to prohibitive computational costs. In contrast, SeDPO, the pairwise approach, circumvents this by implicitly estimating the partition function, i.e., the denominator of retrieval policy in Eq. (4). This makes pairwise algorithm feasible.

Table 9: The performance on mathematical inference tasks.

	Base LLM	**Base Shot Number**	**AUQA-3shot**	**AUQA-8shot**
CEIL [28]	Llama2-7B-chat	8	-	22.83
DQ-LoRe [58]	Llama2-7B-chat	8	-	25.20
RGER [57]	Llama2-7B-chat	8	-	25.59
Se^2	GPT-Neo-2.7B	3	24.80	22.44
SeDPO	GPT-Neo-2.7B	3	25.59	27.56

.6 Mathematical inference

The results on inference tasks are shown in Table 9. Due to variations in the language models, tasks, instruction templates, training and testing datasets, as well as evaluation metrics used by different methods, and due to limitations in computational resources, it is hard to include all related work in the comparison. So we also collected the settings and reported performance of [57] for reference only. Note that only the AUQA [59] task adopted in RGER [57] belongs to MCQ and is compatible with our scoring framework. The results show SeDPO still outperforms Se^2 on inference tasks, aligning with our main discovery. Notably, though RGER is explicitly built for reasoning-focused tasks, SeDPO achieves comparable effectiveness without being geared toward reasoning capabilities.

.7 0-shot retrieval performance

BEIR [64] is a robust and heterogeneous evaluation benchmark for information retrieval, aiming to assess the 0-shot retrieval capabilities on human-labeled document of retrieval models. We

Table 10: The zero-shot retrieval performance on tasks of BEIR.

	webis-touche2020	**fiqa**	**scidocs**	**arguana**	**nq**	**Avg.**
DPR-BERT	0.0000	0.0002	0.0025	0.0601	0.0012	0.0128
Se^2	0.0000	0.0000	0.0032	0.0287	0.0010	0.0066
SeDPO	0.0000	0.0000	0.0032	0.0701	0.0013	0.0149

evaluate nDCG@10 following BEIR in Table 10. While both Se^2 and SeDPO are fine-tuned on DPR-BERT, they show no significant 0-shot retrieval gains, aligning with BEIR's observation about dense retrievers' generalization limitations on human-labeled documents. Furthermore, as Table 3 analyzed, SeDPO and Se^2's task-preference specialization can compromise semantic similarity modeling, which may be detrimental to content-similarity tasks in BEIR. Addressing this requires strategies such as scaling and task-aware prompting, which presents a promising research direction.

.8 Extra analyse

Discussion of ϕ_{LLM} in open-ended QA setting. Our paper focused on MCQ setting, with theoretical soundness. We also empirically analyze open-ended QA as follows. For open-ended QA, the set $\mathcal{Y}$ becomes impractically large, making Eq. (2) intractable to compute. We have experimented with sampling y values for $\mathcal{Y}$ directly from the model; however, the results were unsatisfactory. Unlike MCQs, open-ended QA lacks a clear way to quantify and normalize the quality gap between good and bad answers, making reliable supervision difficult.

Proof of transitivity. By the definition of $\succ$ (Section 3.2):

- If $e^a \succ e^b \mid x$, then for all $y \in \mathcal{Y}_{\text{gt}}$, $S_{\text{MCQ}}(e^a \oplus x, y) > S_{\text{MCQ}}(e^b \oplus x, y)$. (1)
- If $e^b \succ e^c \mid x$, then for all $y \in \mathcal{Y}_{\text{gt}}$, $S_{\text{MCQ}}(e^b \oplus x, y) > S_{\text{MCQ}}(e^c \oplus x, y)$. (2)

Note that $S_{\text{MCQ}}(\cdot)$ is a scalar-valued function, and its outputs are real numbers. The ">" relation on the real numbers is transitive: for any real numbers a, b, c, if $a > b$ and $b > c$, then $a > c$.

Applying this transitivity to (1) and (2) for each $y \in \mathcal{Y}_{\text{gt}}$:

For all $y \in \mathcal{Y}_{\text{gt}}$, $S_{\text{MCQ}}(e^a \oplus x, y) > S_{\text{MCQ}}(e^b \oplus x, y)$ and $S_{\text{MCQ}}(e^b \oplus x, y) > S_{\text{MCQ}}(e^c \oplus x, y)$ implies $S_{\text{MCQ}}(e^a \oplus x, y) > S_{\text{MCQ}}(e^c \oplus x, y)$. (3)

By the definition of $\succ$ again, (3) implies $e^a \succ e^c \mid x$. Thus, the relation $\succ$ is transitive.

Q.E.D.

Cost of constructing training data. We provide in Table 11 the average cost of constructing preference data for all tasks. The scoring batch size is 10, using GPT-Neo-2.7B as the ICL model. For each x, we only sample $T = 20$ preference pairs. This means the number of processed preference pairs is less than the number of scored entries. For instance, on NLI, the total time for constructing scored data is 794× greater than that for preference data:

Table 11: Cost of constructing training data.

Steps to construct trainset	**Speed**
Score data (Se2)	0.0241 s/entry
Rank scored data (SeDPO)	0.0007 s/pair

Table 12: Impact of T

Method	**T=1**	**T=10**	**T=20**	**T=30**
SeDPO (β=0.02)	66.1	74.5	77.9	79.9

.9 More case studies

In addition, we provide more interesting cases in Table 13. Various finetuning-based baselines of our experiments are considered.

Table 13: More case studies on different tasks.

Task: Paraphrase (MRPC)

Test Input: "And it's going to be a wild ride," said Allan Hoffenblum, a Republican consultant. "Now the rest is just mechanical," said Allan Hoffenblum, a Republican consultant. Please tell me if the sentences above mean the same.

UPRISE:
Example1: His wife, who he married in a first ever space wedding by a space phone during his lengthy mission, waited in Moscow. His wife Yekaterina Dmitriyeva, whom he married in a first ever space wedding by a space phone during his daunting mission, was waiting for him in Moscow. Please tell me if the sentences above mean the same. Yes.

Example2: The main change, said Jim Walton, CNN's president, is a fundamental shift in the way CNN collects its news. The main change, said CNN president Jim Walton, was a fundamental shift in the way the network collected its news. Please tell me if the sentences above mean the same. Yes.

Example3: "There were," said board member and Nobel-prize winning Stanford physicist Douglas Osheroff, "some extremely bad decisions." Board member Douglas Osheroff, a Nobel-prize winning Stanford physicist, said: "There were some extremely bad decisions." Please tell me if the sentences above mean the same. Yes.

Test Input Answer: Yes. ×

Se^2:
Example1: The Nets and the Spurs are crossing new frontiers of offensive ineptitude while causing their high-scoring American Basketball Association forefathers to cringe. The Nets and the San Antonio Spurs are crossing new frontiers of offensive ineptitude while embarrassing their high-scoring ABA forefathers. Please tell me if the sentences above mean the same. Yes.

Example2: The girl turned up late Sunday at a convenience store in East Palo Alto, about 30 miles from her home. The girl turned up late Sunday night at an East Palo Alto convenience store about 30 miles from her home. Please tell me if the sentences above mean the same. Yes.

Example3: "There were," said board member and Nobel-prize winning Stanford physicist Douglas Osheroff, "some extremely bad decisions." Board member Douglas Osheroff, a Nobel-prize winning Stanford physicist, said: "There were some extremely bad decisions." Please tell me if the sentences above mean the same. Yes.

Test Input Answer: Yes. ×

SeDPO:
Example1: He was sentenced in June to more than seven years in prison for securities fraud, perjury and other crimes. He was sentenced to more than seven years in prison after pleading guilty to charges including securities fraud. If the first sentence is true, is the second one also true? No.

Example2: He really left us with a smile on his face and no last words, daughter Linda Hope said. "He really left us with a smile on his face and no last words...He gave us each a kiss and that was it." she said. If the first sentence is true, is the second one also true? No.

Example3: This Palm OS smart phone is the last product the company will release before it becomes a part of palmOne. This was almost certainly its last full quarter before the company becomes a part of Palm. If the first sentence is true, is the second one also true? No.

Test Input Answer: No. ✓

Task: Paraphrase (QQP)

Test Input: What is teeth scaling? How is teeth scaling done? Do those questions have the same meaning?

UPRISE:
Example1: How much time is needed to master Quantum Mechanics from scratch? How much time is required to master Quantum mechanics from scratch? Do those questions have the same meaning? Yes.

Example2: Why was the holocaust so slow? Why did the holocaust happen so slowly? Do those questions have the same meaning? Yes.

Example3: Why can't we make artificial blood? Why we can't make artificial blood? Do those questions have the same meaning? Yes.

Test Input Answer: Yes. ×

Se^2:
Example1: The best of 2016: Which are the best Bollywood movies in 2016? Which Bollywood movie you like the most in 2016? Do those questions have the same meaning? Yes.

Example2: How does starving help to lose weight? How can starving yourself to lose weight adversely affect your health? Do those questions have the same meaning? Yes.

Example3: Why are so few drugs with promising animal trials tested in humans? How much do clinical trials for drugs cost? Do those questions have the same meaning? No.

Test Input Answer: Yes. ×

SeDPO:
Example1: Is PC gaming better than console gaming? Is PC gaming better? Please tell me if those questions are the same. Yes.

Example2: Should people over 95 not be allowed to vote? Should people over 93 not be allowed to vote? Please tell me if those questions are the same. Yes.

Example3: Which hp laptop is best for a graphic design/gamer? Which is best HP or Dell laptop? Would you say that these questions are the same? No.

Test Input Answer: No. ✓

Task: Paraphrase (PAWS)

Test Input: Do these mean the same? Wilbur was born on 1 March 1921 in North Caldwell, New Jersey and grew up in New York City. Wilbur was born in North Caldwell, New Jersey March 1, 1921, and grew up in New York City.

UPRISE:
Example1: Are these paraphrases? Born in Gosforth, Northumberland, he moved to the south as a boy to Wiseton Estate, near Retford, Nottinghamshire, when his father found jobs there. Born in Retford, Nottinghamshire, he moved as a boy to Wiseton Estate, near Gosforth, Northumberland, when his father found jobs there. No.

Example2: Do these mean the same? J.David Spurlock was born on November 18, 1959 in Memphis, Tennessee. He moved to Dallas, Texas in 1973. David Spurlock was born on 18 November 1959 in Dallas, Texas, and moved to Memphis, Tennessee in 1973. No.

Example3: Do these mean the same? Joe was born in Somerville, Massachusetts on March 27, 1929 and grew up in Quincy, Massachusetts. Joe was born on March 27, 1929 in Quincy, Massachusetts, where he grew up in Somerville, Massachusetts. No.

Test Input Answer: NO. ×

Se^2:
Example1: Are these paraphrases? The following sound changes from Proto-Celtic to Welsh, Cornish and Breton are summarised in the regular consonant table. The regular consonantal sound changes from Proto-Celtic to Welsh, Cornish and Breton are summarised in the following table. No.

Example2: Do these two sentences from wikipedia have the same meaning? Kennell was born in Colorado Springs, Colorado, and spent her early years between the Rockies and Dunedin, Florida. Kennell was born in Dunedin, Florida, and spent her early years between the Rockies and Colorado Springs, Colorado. No.

Example3: Do these two sentences from wikipedia have the same meaning? Robert Maass was born in East Orange, New Jersey, to study German immigrants Hedwig and Clara Maass. Robert Maass was born in East Orange, New Jersey, to German immigrants Hedwig and Clara Maass. Yes.

Test Input Answer: No. ×

SeDPO:
Example1: Do these two sentences from wikipedia have the same meaning? Several of these names were chosen to correspond to their international equivalents in rough chess, and not as literal translations of the Japanese names. These names were chosen to correspond to their international counterparts in the rough chess and not as literal translations of Japanese names. Yes.

Example2: Do these two sentences from wikipedia have the same meaning? Due to the results obtained in the previous round, Kevin Gleason received + 30kg , Gianni Morbidelli + 20kg and Pepe Oriola + 10kg. Due to the results of the previous round Kevin Gleason received + 30kg, Gianni Morbidelli + 20kg and Pepe Oriola + 10kg. Yes.

Example3: Do these two sentences from wikipedia have the same meaning? When combined for joint or coalition operations, it was known as a common or employed air operations centre for coalition operations. When combined for joint or coalition operations, it was known as a joint or employed air operations center for coalition operations. Yes.

Test Input Answer: Yes. ✓