Learning to Select In-Context Demonstration Preferred by Large Language Model

Anonymous ACL submission

Abstract

In-context learning (ICL) enables large language models (LLMs) to adapt to new tasks during inference using only a few demonstrations. However, ICL performance is highly dependent on the selection of these demonstrations. Recent work explores retrieval-based methods for selecting query-specific demonstrations, but these approaches often rely on surrogate objectives such as metric learning, failing to directly optimize ICL performance. Consequently, they struggle to identify truly beneficial demonstrations. Moreover, their discriminative retrieval paradigm is ineffective when the candidate pool lacks sufficient high-quality demonstrations. To address these challenges, we propose GENICL, a novel generative preference learning framework that leverages LLM feedback to directly optimize demonstration selection for ICL. Experiments on 19 datasets across 11 task categories demonstrate that GENICL achieves superior performance than existing methods in selecting the most effective demonstrations, leading to better ICL performance.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks. In-context learning (ICL), introduced by (Brown et al., 2020), enables LLMs to perform tasks with only a few examples as demonstration, without requiring parameter updates (Brown et al., 2020; Liu et al., 2021).

Although ICL is promising, the selection of in-context demonstrations is crucial, as it significantly impacts LLM performance (Lu et al., 2021; Min et al., 2022b). Many existing approaches rely on retrieval-based methods, either using off-the-shelf retrievers (Robertson et al., 2009) or training specialized ones (Reimers, 2019; Wang et al., 2022). Training-based methods typically optimize retrievers to approximate the relevance score of the candidate to the given query for downstream ICL tasks, often leveraging metric learning-based proxy tasks, such as contrastive loss (Rubin et al., 2021; Ye et al., 2023; Wang et al., 2023; Cheng et al., 2023; Luo et al., 2023; Liu et al., 2024).

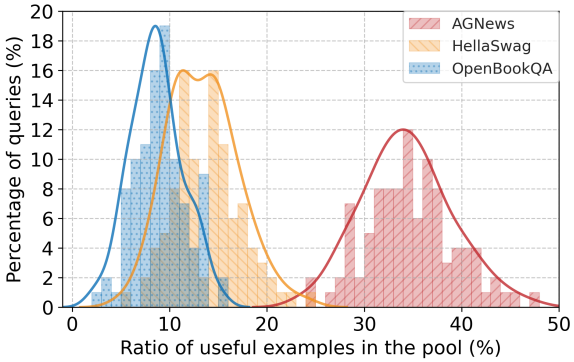

Figure 1: The distribution of the useful example ratio across test sets from different datasets. The x-axis represents the ratio $\frac{\text{\# useful examples}}{\text{\# total examples}}$, where a 'useful example' is one that helps the LLM generate an accurate output with in-context learning. The ratio remains low for most test queries, i.e., the majority of demonstration examples are ineffective for the LLM.

However, these methods face several challenges. The most critical issue is the misalignment between the surrogate learning objective of the retriever and the intrinsic optimization goal of ICL. Specifically, a discriminative model, trained with a metric learning objective to approximate the relevance score between a demonstration candidate and a given query, does not necessarily indicate the candidate's effectiveness as an in-context demonstration for an LLM. Furthermore, the scarcity of effective demonstration candidates in the retrieval pool poses another challenge for training a discriminative retriever. Figure 1 illustrates the distribution of test queries regarding the ratio of useful examples that help the LLM achieve accurate output using ICL. The figure shows that most candidates are ineffective for most queries, making retriever optimization particularly challenging. This issue is further exac-

erbated by the properties of hard labels for LLM feedback, as noted by (Wang et al., 2023).

To address these challenges, we propose GENICL, a novel generative preference learning framework that directly optimizes demonstration selection for ICL using LLM feedback. Specifically, we reformulate ICL as a generative Bayesian optimization problem, introducing a latent variable to bridge demonstration selection and LLM inference. GENICL then optimizes this objective through preference learning, which focuses more on the relative effectiveness between demonstration samples, allowing it to capture finer-grained information in scenarios where effective demonstrations are scarce. This optimization procedure on demonstration selection more closely aligns with the intrinsic objective of ICL. Through this approach, GENICL can better capture the LLM's demonstration preference and identify the truly effective demonstrations (preferred by the LLM). To validate the effectiveness of GENICL, we conducted experiments on a diverse set of datasets covering classification, multiple-choice, and generation tasks. The results demonstrate that GENICL achieves superior performance compared to existing methods.

Our contributions can be summarized as follows:

- We propose GENICL, a novel generative preference learning framework that directly optimizes demonstration selection for ICL using LLM feedback, overcoming the limitations of sub-optimality of surrogate learning objective used in traditional retrieval-based methods.
- The preference learning of paired effective and ineffective demonstrations leads to fine-grained distinguishness of demonstration preference of LLM.
- We conduct extensive experiments on 19 datasets across 11 task categories with a quantitative and qualitative analysis of the effectiveness of our proposed method.

The rest of the paper is organized as follows. Section 2 surveys the related work and points out the existing challenges in optimization ICL tasks. Section 3 details our GENICL with generative ICL modeling and the optimization approach of preference learning from LLM feedback. Section 4 introduces the experimental setup and the quantitative analysis with an analysis of how different methods work. Finally, Section 5 summarizes our paper, and Section 6 discusses the limitations.

2 Related work

2.1 In-context Demonstration Selection

In-context learning (Brown et al., 2020) enhances large language models by leveraging few-shot examples for task adaptation. While proficient at using provided examples, LLMs heavily depend on specific demonstrations (Min et al., 2022a; Luo et al., 2024), making demonstration selection crucial for downstream performance. Existing approaches to demonstration selection can be broadly categorized into two branches. The first employs off-the-shelf retrievers like BM25 (Robertson et al., 2009), SBERT (Reimers, 2019), and $E5_{base}$ (Wang et al., 2022), which prioritize semantic similarity but overlook downstream utility. The second category involves learning the retriever based on feedback from LLM performance on selected demonstrations (Li et al., 2023; Cheng et al., 2023; Luo et al., 2023; Chen et al., 2024). For instance, Rubin et al. (2021) trained a dense retriever using LLM-generated feedback, while Wang et al. (2023) distilled a retriever from a metric function estimating relevance scores between demonstration candidates and the query. However, these methods approximate the relevant score of the candidate given the query, which is not essentially aligned with the true optimization objective of ICL resulting in a sub-optimal solution. Moreover, only a minority of candidates truly enhance in-context learning performance (Figure 1), making it challenging to learn a reasonably dense retriever. Therefore, a more promising method is required for directly optimizing the intrinsic ICL objective.

2.2 Preference Learning

Preference learning plays a vital role in training LLMs to align generative models with human values and preferences (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022). It has also emerged as a powerful paradigm for enhancing LLM performance in complex tasks (Havrilla et al., 2024; Li et al., 2024). A key approach in this domain is reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), which involves training a reward model to capture human preferences and optimizing the model using reinforcement learning algorithms (Schulman et al., 2017). More recent methods, such as Direct Preference Optimization (DPO) (Rafailov et al., 2024) and Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024), bypass explicit reward mod-

eling and instead learn preferences directly from human feedback. While these methods have advanced preference learning performance, they primarily focus on human feedback without explicitly addressing the issue of the scarcity of effective demonstrations.

3 Methodology

3.1 Problem Setting

Consider a task sample x with ground-truth y, denoted as (x, y), and a set of demonstrations $\{(x_k, y_k)\}_{k=1}^{K} \subseteq \mathcal{P}$ where each demonstration (x_k, y_k) is an input-output tuple taken from the demonstration pool $\mathcal{P}$ containing all training samples from all tasks. K is the number of demonstrations used for the current query. Our goal is to provide the appropriate demonstrations for each x to improve the ICL performance of LLMs as

$$\arg\max_{(x_k,y_k)\in\mathcal{P}} P_{\mathcal{M}}(y \mid \{(x_k, y_k)\}_{k=1}^{K}, x) \quad (1)$$

3.2 Optimization Objective

Most methods based on LLM feedback used metric learning, such as contrastive learning, as a surrogate objective to train a discriminative model as the retriever. However, this optimization objective has a gap with in-context learning performance (details in Appendix C). Additionally, the relevance scores determined by the discriminative retriever come from a feature similarity perspective, overlooking optimizations at the semantic understanding level, which leads to the selected demonstrations that may not be truly useful for improving the LLM's in-context learning performance. To address these issues, we propose a generative framework to directly optimize demonstration selection for ICL performance, thereby avoiding the use of surrogate objectives that have a gap. We treat in-context learning as generative Bayesian optimization and introduce a latent variable z to establish the relationship between demonstration selection and ICL performance. The optimization problem is formulated as follows:

$$P_{\mathcal{M}}(Y \mid \{(X_k, Y_k)\}_{k=1}^{K}, X) = \int_z P_{\mathcal{M}}(Y \mid z, X)\, P_{\mathcal{M}}(z \mid \{(X_k, Y_k)\}_{k=1}^{K}, X)\, dz \quad (2)$$

This transformation is also supported by the perspectives of Xie et al. (2021) and Wang et al. (2024), where in-context learning can be viewed as a Bayesian inference process.

The latent variable z is viewed as the representation of the LLM's demonstration preference. Through z, we concretize the LLM's preference for demonstration and use it to identify effective demonstrations. Eq. (2) proves from a Bayesian perspective that, during the optimization process, both the ground truth and demonstrations should be considered simultaneously. However, in the surrogate optimization objectives of retriever-based methods, only the quality of the demonstrations is considered. This is why demonstrations selected by these methods do not necessarily enhance the LLM's ability to predict the ground truth within ICL.

To obtain the demonstration preference, we propose to fully optimize the latent variable z by jointly considering the ground truth and the preference of the demonstrations. We utilize the Evidence Lower Bound (ELBO) (Kingma, 2013) and Eq. (2) to obtain the following optimization objective, with the detailed derivation provided in Appendix D.

$$\mathcal{L}(z) = -\log P_{\mathcal{M}}(Y \mid z, X) - \log P_{\mathcal{M}}(z \mid \{(X_k, Y_k)\}_{k=1}^{K}, X) \quad (3)$$

Due to the combinatory search space of $(X_1, Y_1), (X_2, Y_2), ..., (X_K, Y_K)$ being immense, we simplify the optimization objective to the following formula.

$$\mathcal{L}(z) = -\log P_{\mathcal{M}}(Y \mid z, X) - \log P_{\mathcal{M}}(z \mid (X_k, Y_k), X) \quad (4)$$

3.3 Demonstration Preference Learning

The above optimization objective aims to find the variable z of the demonstration preference of the LLM. From an implementation perspective, z is a task-specific description, and its corresponding trainable parameters are θ. We optimize θ to refine the demonstration preference distribution corresponding to z.

To perform effective optimization in scenarios where effective demonstrations are scarce, we use preference learning, as it focuses more on the relative relationships between demonstrations, enabling it to capture finer-grained information. To learn the preference for effective demonstrations, we adopt the KTO algorithm (Ethayarajh et al.,

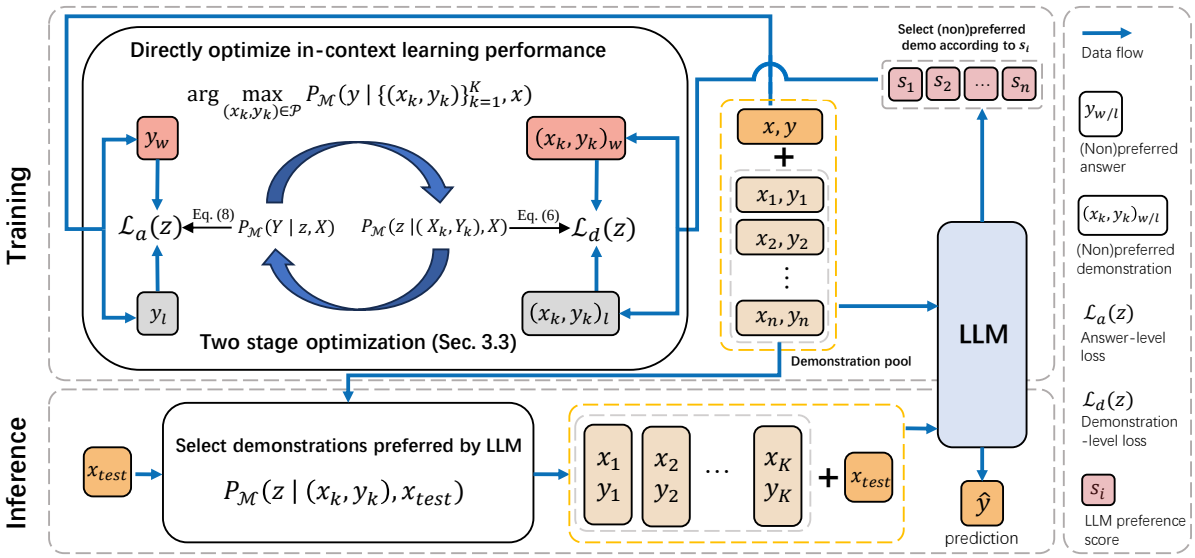

Figure 2: Training and inference pipeline of GENICL.

2024), which directly maximizes the utility of LLM generations based on context information, conditioned on either the effective demonstration $(x_k, y_k)_w$ or the relatively ineffective one $(x_k, y_k)_l$. This ensures that $(x_k, y_k)_w \succ (x_k, y_k)_l$ given the input x. To achieve this, we reformulate the optimization objectives, $-\log P_{\mathcal{M}}(Y \mid z, X)$ and $-\log P_{\mathcal{M}}(z \mid (X_k, Y_k), X)$ in Eq. (4), as a two-stage alternating optimization process.

The First Stage. In this stage, we optimize the demonstration selection part $-\log P_{\mathcal{M}}(z \mid (X_k, Y_k), X)$ in Eq. (4) with preference learning. Preference learning requires both preferred data and non-preferred data. To obtain the LLM preferences for demonstrations under the in-context learning paradigm, we utilize the ICL performance of LLM given each demonstration candidate as the preference score. The preference score s_k is the log-likelihood of LLM generation w.r.t. the ground-truth output y given the candidate (x_k, y_k) and the input x, reflecting the LLM's preference for the demonstration.

$$s_k = P(y|(x_k, y_k), x) \tag{5}$$

Since we need to score all the training data based on the above method to obtain the preference dataset, and the demonstration pool for each training sample is $\mathcal{P}$, the complexity of scoring all the data is quadratic in $|\mathcal{P}|$, which is computationally infeasible for an exhaustive search. Therefore, to reduce the search space, we use E5_{base} (Wang et al., 2022) to obtain a more relevant subset $\xi \subseteq \mathcal{P}$ as the new demonstration pool. For each training sample, we select the set of candidates with the highest preference scores as preferred demonstrations $(x_k, y_k)_w$, and the set of candidates with the lowest preference scores as non-preferred demonstrations $(x_k, y_k)_l$. The demonstration-level loss is formulated as follows.

$$\begin{aligned}\mathcal{L}_d(\theta) = -\mathbb{E}_{((x_k,y_k)_w,(x_k,y_k)_l,x,z)\sim\mathcal{D}} \\ \left[\lambda_w \sigma\left(\beta \log \frac{P_{\mathcal{M}}(z \mid (x_k,y_k)_w, x)}{P_{\mathcal{M}_{\text{ref}}}(z \mid (x_k,y_k)_w, x)} - s^d_{\text{ref}}\right)\right. \\ \left. + \lambda_l \sigma\left(s^d_{\text{ref}} - \beta \log \frac{P_{\mathcal{M}}(z \mid (x_k,y_k)_l, x)}{P_{\mathcal{M}_{\text{ref}}}(z \mid (x_k,y_k)_l, x)}\right)\right]\end{aligned} \tag{6}$$

$$\begin{aligned}s^d_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\beta\, \text{KL}\big(P_{\mathcal{M}}(z \mid (x_k,y_k), x) \\ \| \; P_{\mathcal{M}_{\text{ref}}}(z \mid (x_k,y_k), x)\big)\Big]\end{aligned} \tag{7}$$

where β is a hyperparameter, λ_w and λ_l are hyperparameters for preferred data and non-preferred data respectively. s_{ref} represents the use of KL divergence as a constraint to prevent the model from deviating too much from its original predictive ability, thereby enhancing the stability of the model.

The Second Stage. We then optimize the model generation part $-\log P_{\mathcal{M}}(Y \mid z, X)$ in Eq. (4) in a similar manner of preference learning. For all

tasks, we treat the ground truth as the correct answer y_w. For classification and multi-choice tasks, we consider the incorrect categories or options as wrong answers y_l, and for generation tasks, we randomly sample answers from other samples as y_l for the current sample. The answer-level loss is formulated as follows:

$$\mathcal{L}_a(\theta) = -\mathbb{E}_{(z,x,y_w,y_l)\sim\mathcal{D}} \left[\lambda_w \sigma\left(\beta \log \frac{P_{\mathcal{M}}(y_w \mid z, x)}{P_{\mathcal{M}_{\text{ref}}}(y_w \mid z, x)} - s^a_{\text{ref}}\right) + \lambda_l \sigma\left(s^a_{\text{ref}} - \beta \log \frac{P_{\mathcal{M}}(y_l \mid z, x)}{P_{\mathcal{M}_{\text{ref}}}(y_l \mid z, x)}\right)\right] \quad (8)$$

$$s^a_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\beta \,\text{KL}\big(P_{\mathcal{M}}(y \mid z, x) \parallel P_{\mathcal{M}_{\text{ref}}}(y \mid z, x)\big)\Big] \quad (9)$$

where the hyperparameters are consistent with those in the first stage.

In summary, we alternately optimize the losses of the two stages, and the complete optimization objective is as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_a(\theta) + \mathcal{L}_d(\theta) \quad (10)$$

And Algorithm 1 summarizes the whole training procedure.

Algorithm 1: Demonstration Preference Learning

Input: Task dataset $\mathcal{D} = \{(x, y)_i\}_{i=1}^{|\mathcal{D}|}$, latent variable z, demonstration pool $\mathcal{P}$, and learning rates η_1, η_2
Initialization: Parameters θ
for each (x, y) in task $\mathcal{D}$ **do**
  Calculate demonstration scores by Eq. (5), and identify preferred demonstrations $(x_k, y_k)_w$ and non-preferred ones $(x_k, y_k)_l$
  Compute demonstration-level loss $\mathcal{L}_d(\theta)$ by Eq. (6)
  Update $\theta \leftarrow \theta - \eta_1 \nabla_\theta \mathcal{L}_d(\theta)$
  Set the correct answer as $y_w = y$, and select an incorrect answer y_l
  Compute answer-level loss $\mathcal{L}_a(\theta)$ by Eq. (8)
  Update $\theta \leftarrow \theta - \eta_2 \nabla_\theta \mathcal{L}_a(\theta)$
end for

3.4 Demonstration Selection in Inference

In the training stage, we optimize the parameters θ to obtain the variable z of the demonstration preference of the LLM. In the inference stage, we first use E5_{base} (Wang et al., 2022) to reduce the search space and time cost of demonstration selection. Then we leverage the LLM with the latent variable z to select demonstrations for an input query x_{test}. In Eq. (2), for the same x, all demonstrations have the same $P_{\mathcal{M}}(y \mid z, x)$. Therefore, we select the top-K demonstrations based on the probability of generating the latent variable, as shown below:

$$P_{\mathcal{M}}(z \mid (x_k, y_k), x_{\text{test}}) \quad (11)$$

After obtaining a set of demonstrations $(X_k, Y_k)\}_{k=1}^{K}$, we use the frozen LLM to perform in-context learning. The entire training and inference pipeline is shown in Figure 2.

4 Experiments

4.1 Evaluations Setup

Datasets. We validate the effectiveness of GENICL across a range of language processing tasks including classification, multi-choice, and text generation. The classification tasks include **Natural Language Inference** (RTE (Wang, 2018), SNLI (Bowman et al., 2015)), **Paraphrase** (PAWS (Zhang et al., 2019), QQP (Wang, 2018)), **Reading Comprehension** (MultiRC (Khashabi et al., 2018), BoolQ (Clark et al., 2019))), **Sentiment Classification**: (SST2 (Socher et al., 2013), Sentiment140 (Go et al., 2009)), and **Topic Classification** (AGNews (Zhang et al., 2015)). The multi-choice tasks cover **Commonsense Reasoning** (COPA (Roemmele et al., 2011), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018)), **Coreference** (Winogrande (Sakaguchi et al., 2021)). The text generation tasks contain **Close QA** (NaturalQuestions (Kwiatkowski et al., 2019), SQuAD v1 (Rajpurkar, 2016)), **Commonsense Generation** (CommonGen (Lin et al., 2019)), **Data-to-Text** (E2E NLG (Dušek et al., 2019), DART (Nan et al., 2020)), and **Summarization** (AESLC (Zhang and Tetreault, 2019), Gigaword (Napoles et al., 2012)).

Baseline. Our evaluation compares GENICL against several established baseline approaches that leverage the ICL capabilities of LLMs. These baselines include zero-shot (which executes tasks without demonstrations), random (which selects demonstrations arbitrarily from the available

pool), and off-the-shelf retrieval models such as BM25 (Robertson et al., 2009), SBERT (Reimers, 2019), and $E5_{base}$ (Wang et al., 2022). Furthermore, we conduct comparative analyses against the following more advanced approaches.

- EPR (Rubin et al., 2021) is a prominent method using LLM feedback to label candidate examples as positive or negative, to train the retriever.
- LLM-R (Wang et al., 2023) trains the retriever using a reward model and knowledge distillation, which is one of the state-of-the-art retriever-based methods.
- CBDS (concept-based-demonstration-selection) (Wang et al., 2024) first learns the latent variable and then selects demonstrations with the highest probability of predicting the corresponding latent variable.

Implementation Details. To ensure a fair comparison with the baselines, we use the same template and evaluation metrics. For detailed information, please refer to the Appendix E. Our approach demonstrates enhanced efficiency through a reduced number of training parameters relative to retriever-based methodologies such as LLM-R (Wang et al., 2023) and $E5_{base}$ (Wang et al., 2022). For model implementation, we utilize Llama-7B (Touvron et al., 2023) as our primary LLM for both training and inference operations. Our inference protocol incorporates a set of 8 demonstrations within the context, with complete implementation details provided in Appendix A.

4.2 Main Results

Table 1 presents the performance comparison of the proposed GENICL and various baselines for classification, multi-choice, and text generation tasks. The results demonstrate that GENICL significantly outperforms all the baselines on most tasks. Zero-shot and Random consistently yield the worst performance across most tasks, highlighting the importance of using high-quality demonstrations in ICL. Retriever methods trained based on LLM feedback provide only limited improvements over off-the-shelf retrievers. Specifically, EPR performs worse than $E5_{base}$ on tasks such as MultiRC, RTE, and Sentiment140. This indicates that these discriminative methods, which use proxy optimization objectives, often fail to select truly useful demonstrations. Our generative Bayesian method GENICL, in contrast, significantly outperforms EPR and LLM-R, proving its effectiveness. Although GENICL uses $E5_{base}$ to narrow the search space, compared to directly using $E5_{base}$, we achieve significant improvements, especially on tasks such as PAWS, BoolQ, and SNLI, where our GENICL improves by more than 5 points. This indicates that directly using $E5_{base}$ struggles to identify demonstrations preferred by the LLM, whereas our method effectively addresses this issue. While CBDS employs a comparable latent concept variable methodology for demonstration selection, its performance is constrained due to adopting a misaligned optimization objective (details in Appendix A). Through the implementation of a more refined optimization objective, GENICL demonstrates superior performance.

For text generation tasks, the improvements of EPR and LLM-R over off-the-shelf retrievers are all relatively limited. This may be because metrics like ROUGE and EM typically exhibit a narrower range of variation compared to classification accuracy, which is consistent with the findings of Wang et al. (2023). Even so, GENICL still achieves significant improvements over LLM-R and off-the-shelf retrievers, particularly on the CommonGen and SQuAD v1 tasks. This indicates that GENICL is also highly effective for text generation tasks.

4.3 Ablation Study

To validate the effectiveness of the key design in GENICL, we conduct ablation studies on the CommonGen, BoolQ, and SQuAD v1 tasks, with results presented in Table 2. When non-preferred data is removed during training, the performance of GENICL drops significantly, indicating that using only useful demonstrations without considering relative preference relations makes it difficult to learn the distribution of LLM demonstration preference. This proves the necessity of optimization based on preference learning. Our complete optimization objective consists of two parts: answer-level loss and demonstration-level loss. Removing either leads to a performance drop, demonstrating the correctness of our proposed optimization framework and the importance of considering both the ground truth and the quality of demonstrations during optimization.

4.4 Validation on Other LLMs

The previous experimental results were obtained using the LLaMA-7B model. To comprehensively as-

	Classification Tasks								
	Topic	**Reading Comprehension**		**Paraphrase**		**NLI**		**Sentiment**	
	AGNews	BoolQ	MultiRC	PAWS	QQP	RTE	SNLI	Sentiment140	SST2
Zero-shot	31.4	64.7	57.0	53.0	57.9	59.9	39.6	49.3	54.2
Random	65.0	69.6	**60.4**	49.6	54.0	65.7	40.4	78.8	64.1
BM25	90.0	74.0	58.7	56.5	80.3	59.9	47.7	88.3	84.7
SBERT	89.8	73.6	53.3	58.3	81.7	60.2	56.2	94.1	87.8
$E5_{base}$	90.6	71.0	54.0	55.6	77.3	68.5	53.7	93.0	92.4
CBDS	67.3	77.6	49.3	57.6	64.2	56.3	43.5	92.5	69.2
EPR	91.8	74.8	50.4	57.7	81.7	66.8	68.4	91.4	88.7
LLM-R	92.4	74.9	50.2	57.5	80.9	61.7	80.0	91.6	93.4
GENICL (ours)	**92.6**	**78.1**	56.9	**63.9**	**82.0**	**72.9**	**84.6**	**94.7**	**95.0**

	Multi-Choice Tasks				Text Generation Tasks					
	Coreference	**Commonsense Reasoning**			**Summarize**		**CommonGen**	**Data-to-text**		**CloseQA**
	Winogrande	COPA	HellaSwag	OpenBookQA	AESLC	Gigaword	CommonGen	DART	E2ENLG	SQuADv1
Zero-shot	61.8	66.0	71.5	41.6	5.7	15.4	19.2	22.8	34.6	2.2
Random	62.1	74.0	72.1	43.0	6.5	27.2	36.3	34.5	51.1	47.3
BM25	66.4	77.0	74.6	47.8	23.9	32.5	38.3	55.4	53.8	53.7
SBERT	66.9	81.0	**75.2**	49.2	22.3	31.7	37.8	54.6	50.6	62.5
$E5_{base}$	66.7	84.0	75.0	51.0	23.4	31.9	37.6	54.9	52.3	61.9
CBDS	66.3	83.0	73.6	47.6	20.5	29.2	34.5	52.2	50.8	59.8
EPR	66.5	82.0	**75.2**	49.6	**26.0**	32.4	39.2	56.2	53.6	64.3
LLM-R	66.9	85.0	74.6	50.8	**26.0**	32.5	37.2	56.0	54.4	61.8
GENICL (ours)	**68.0**	**86.0**	74.6	**51.8**	24.4	**33.0**	**41.0**	**56.4**	**55.1**	**65.7**

Table 1: Main results on classification, multi-choice, and text generation tasks.

	CommonGen	BoolQ	SQuAD v1
GENICL	41.0	78.1	65.7
- w/o non-preferred data	38.7	71.2	62.5
- w/o answer-level loss	34.9	72.7	62.0
- w/o demo-level loss	37.9	65.8	61.7

Table 2: Ablation study.

	CommonGen	E2E NLG	PAWS	SST2
Zero-shot	19.2	34.6	53.0	54.2
Random	36.3	51.1	49.6	64.1
BM25	38.3	53.8	56.5	84.7
$E5_{base}$	37.6	52.3	55.6	92.4
GENICL (ours)				
- GPT-Neo 2.7B	35.3	50.1	60.5	92.7
- LLaMA-7B	41.0	55.1	63.9	95.0
- Vicuna-13B	39.7	56.3	71.7	94.0

Table 3: Performance on other LLMs.

sess the efficacy of GENICL, we expanded our evaluation to encompass various LLMs, specifically GPT-Neo 2.7B (Black et al., 2021) and Vicuna-13B (Zheng et al., 2023). The comparison results are presented in Table 3.

The GENICL method using language models of different scales shows performance improvements on most tasks, demonstrating the generalizability and effectiveness of GENICL. Specifically, the GENICL method with GPT-Neo 2.7B outperforms $E5_{base}$ on classification tasks like PAWS and SST2, but performs worse than the baseline on text generation tasks such as CommonGen and E2E NLG. When using LLaMA-7B and Vicuna-13B, GENICL shows significant improvements in both classification and text generation tasks. We speculate that text generation tasks require more complex logical reasoning steps and smaller models like GPT-Neo 2.7B face difficulties in understanding complex semantics and performing reasoning.

4.5 The Superiority of Our Optimization Objective

Our optimization objective is directly derived from the ICL paradigm, making it more aligned with ICL compared to retriever-based methods. We compare the probability of predicting the ground truth using demonstrations selected by different methods, as described in Eq. (1). The results, shown in Figure 3, indicate that compared to retriever-based methods such as LLM-R and SBERT, GENICL achieves higher predicted probabilities for the ground truth, with a more concentrated probability distribution. This demonstrates that our optimization objective better enhances ICL performance, further validating our previous conclusions.

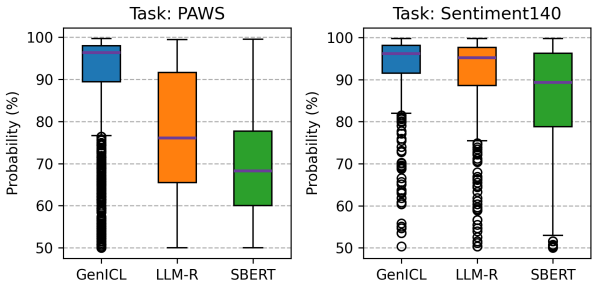

Figure 3: The predicted probability of the ground truth using demonstrations selected by different methods.

4.6 The Impact of the Number and Order of Demonstrations

In Figure 4, we examine how the performance of GENICL changes as the number of demonstrations varies on the following four datasets: RTE, SST2, CommonGen, and Gigaword. The results show that increasing the number of demonstrations does not always lead to better performance. For SST2 and CommonGen, a significant drop in performance is observed when the number of demonstrations is too large, while Gigaword experiences a slight decrease.

To investigate the impact of the order of demonstrations on downstream task performance, we compare three different order settings:

- Shuffle: the top-K selected demonstrations are randomly shuffled.
- Descending: the top-K selected demonstrations are ordered in descending order of Eq. (11).
- Ascending: the top-K selected demonstrations are ordered in ascending order of Eq. (11).

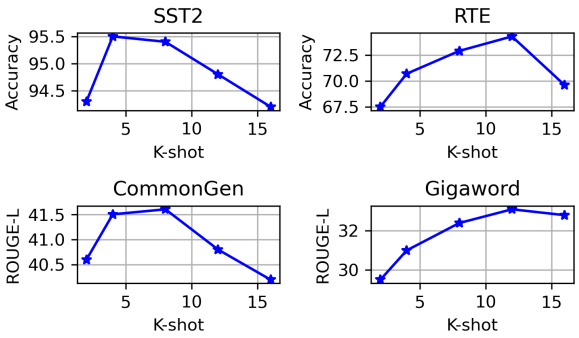

Figure 4: The effect of demonstration number K.

The results are shown in Table 4. Lu et al. (2021) have revealed that ICL is sensitive to the order of demonstrations when using random examples. However, we found that the order of the demonstrations selected by our method has little impact on the final in-context learning performance. This supports the conclusion that high-quality demonstrations are less sensitive to the ordering (Li et al., 2023; Chen et al., 2022; Li and Qiu, 2023). This further illustrates that the demonstrations selected by our method, based on preference learning, are of high quality.

Demo. order	Agnews	QQP	Gigaword	E2E NLG
Shuffle	92.8	81.9	32.6	55.2
Descending	92.6	82.0	33.0	55.1
Ascending	92.8	82.5	32.6	54.9

Table 4: Comparison of different demonstration orders.

5 Conclusion

In this paper, we introduced GENICL, a novel generative preference learning framework for optimizing demonstration selection in in-context learning (ICL). By reformulating ICL as a generative Bayesian optimization problem, GENICL bridges demonstration selection and LLM inference through a latent demonstration variable, aligning the intrinsic goal of ICL. GENICL also leverages preference learning on relatively effective and ineffective demonstrations from LLM feedback, achieving fine-grained demonstration selection. A wide range of experiments have illustrated the superiority of our proposed method GENICL. We believe that our findings contribute to advancing the field of ICL and hope that GENICL serves as a foundation for future research in directly optimizing demonstration selection for LLMs.

6 Limitations

During the optimization phase, we treat each demonstration independently, ignoring the interactions between demonstrations (such as order and combination). In the inference phase, we select a set of demonstrations based on their scores, but the combination of individually optimal demonstrations does not necessarily result in the overall best combination.

Our method requires scoring each candidate in the demonstration pool during the demonstration selection stage. However, the search space and computational cost of this process are prohibitive. To address this, we employ $E5_{base}$ to reduce the search space by obtaining a subset of the demonstration pool as a new, smaller pool. However, this approach may filter out some valuable demonstrations. Additionally, improving the efficiency of demonstration scoring remains a promising direction for further exploration.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58(2).

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Yunmo Chen, Tongfei Chen, Harsh Jhamtani, Patrick Xia, Richard Shin, Jason Eisner, and Benjamin Van Durme. 2024. Learning to retrieve iteratively for in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7156–7168.

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

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Ondřej Dušek, David M Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. *arXiv preprint arXiv:1911.03905*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. 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.

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jialian Li, Yipin Zhang, Wei Shen, Yuzi Yan, Jian Xie, and Dong Yan. 2024. Boosting deductive reasoning with step signals in rlhf. *arXiv preprint arXiv:2410.09528*.

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

Xiaonan Li and Xipeng Qiu. 2023. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts. *arXiv preprint arXiv:2305.05181*, 16.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Wenhan Liu, Yutao Zhu, and Zhicheng Dou. 2024. Demorank: Selecting effective demonstrations for large language models in ranking task. *arXiv preprint arXiv:2406.16332*.

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

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

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

Courtney Napoles, Matthew R Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the joint workshop on automatic knowledge base construction and web-scale knowledge extraction (AKBC-WEKEX)*, pages 95–100.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI spring symposium series*.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. *arXiv preprint arXiv:1906.03497*.

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

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Implementation Details

We use LoRA (Hu et al., 2021) to train the parameters corresponding to the latent variable, and the total number of trainable parameters is approximately 79M. The hyperparameters of GENICL are summarized in Table 5. We use 8 NVIDIA A100-80GB GPUs for both training and inference. For a single task, the entire pipeline takes approximately two days. For a fair comparison, our method and all baselines use the same template, as shown in Appendix E.

	GENICL
Optimizer	AdamW
Warmup steps	3000
Training steps	20000
Learning rate	5e-6
Batch size	32
Maximum input length	1024
Maximum output length	64
β	0.1
λ_w	1.0
λ_l	1.0

Table 5: Hyper-parameters.

Implementation Details of CBDS (Wang et al., 2024). For the CBDS baseline, we used the authors' publicly available code and followed the training methodology described in the original paper. The original CBDS setup involves concurrent training across multiple tasks, with each task optimizing its own set of concept tokens θ. Specifically, the optimization objective during training is to minimize $-\log P(Y|\theta, X)$. During testing, demonstrations (X_k, Y_k) are selected based on $P(\theta|X_k, Y_k)$. Training consisted of 10,000 steps with a learning rate of 1e-2 and a batch size of 16. A crucial difference between CBDS and both our method and other retriever-based approaches is that CBDS uses a shared set of demonstrations for all queries within a specific task. This design, while computationally efficient, often leads to performance inferior to that of retriever-based methods, which dynamically select demonstrations for each query. This performance difference was also observed in our reproduction experiments.

Implementation Details of LLM-R (Wang et al., 2023). Following the LLM-R baseline procedure, we first trained a reward model and then used it for distillation to train the LLM-R model, which uses the pretrained E5_{base} model as the initialization model. We utilized the code provided by the authors, retraining both the reward model and the LLM-R model while maintaining the hyperparameters from the original publication. For EPR (Rubin et al., 2021), we directly adopt the results presented in the LLM-R paper (Wang et al., 2023).

B Details about useful samples statistics

Figure 1 illustrates the distribution of the ratio of useful examples within demonstration pools across the AGNews, HellaSwag, and OpenBookQA datasets. We first identified "hard queries" defined as those queries that remained answered incorrectly by the LLM even when provided with the top-8 demonstrations initially retrieved by the E5_{base} (Wang et al., 2022). Subsequently, for each identified hard query, a set of 100 demonstrations was randomly sampled without replacement from the respective dataset's full demonstration pool.

Each sampled demonstration was then individually assessed for its usefulness. A demonstration was considered "useful" if its concatenation with the hard query resulted in the LLM producing the correct answer. The *Ratio of Useful Examples* was calculated for each query as $(\text{Number of Useful Demonstrations}/100) \times 100\%$. To visualize the distribution, these ratios were binned, and the *Percentage of Queries* falling within each bin was calculated as (Number of Queries in Bin / Total Number of Hard Queries) $\times$ 100%, where "Number of Queries in Bin" is the count of hard queries with a ratio within that bin's range, and "Total Number of Hard Queries" represents the total number of hard queries for that dataset.

C The Limitations of Contrastive Learning

In in-context learning, it is necessary to find demonstrations that are truly useful for the LLM to predict the ground truth. The goal of in-context learning is as follows:

$$\arg\max_{(x_k,y_k)\in\mathcal{P}} P_{\mathcal{M}}(y \mid \{(x_k,y_k)\}_{k=1}^{K}, x)$$

However, the objective of contrastive learning is to minimize the following loss.

$$\mathcal{L} = -\log\left(\frac{e^{s(x,x^+,y^+)}}{e^{s(x,(x^+,y^+))} + \sum_{i=1}^{N_{\text{neg}}} e^{s(x,(x_i^-,y_i^-))}}\right)$$

In the optimization process of contrastive learning, the objective shifts to optimizing the retriever by comparing the similarity between different samples. Specifically, the model is trained to distinguish between positive samples (x^+, y^+) and negative samples (x^-, y^-), optimizing the model by maximizing the separation between positive and negative samples. This method enlarges the distance between positive and negative samples in the embedding space, which is not aligned with in-context learning. Furthermore, it completely ignores the information about the ground truth during optimization, leading to demonstrations selected by this method that do not improve the LLM's ability to predict the ground truth.

D Derivation of the Optimization Objective

Starting from the marginal likelihood,

$$\log P_{\mathcal{M}}(Y \mid \{(X_k,Y_k)\}_{k=1}^{K}, X) = \log \int_z P_{\mathcal{M}}(Y \mid z, X) \times P_{\mathcal{M}}(z \mid \{(X_k,Y_k)\}_{k=1}^{K}, X)\, dz$$

we introduce an arbitrary auxiliary distribution $q(z)$ (with $\int q(z)\, dz = 1$) and apply Jensen's inequality:

$$\begin{aligned}
&\log P_{\mathcal{M}}(Y \mid \{(X_k,Y_k)\}_{k=1}^{K}, X) \\
&= \log \int_z q(z) \times \\
&\frac{P_{\mathcal{M}}(Y \mid z, X)\, P_{\mathcal{M}}(z \mid \{(X_k,Y_k)\}_{k=1}^{K}, X)}{q(z)}\, dz \\
&\geq \int_z q(z) \times \\
&\log \frac{P_{\mathcal{M}}(Y \mid z, X)\, P_{\mathcal{M}}(z \mid \{(X_k,Y_k)\}_{k=1}^{K}, X)}{q(z)}\, dz \\
&\triangleq \mathcal{L}(q).
\end{aligned}$$

The quantity $\mathcal{L}(q)$ is the evidence lower bound (ELBO) (Kingma, 2013). We approximate $q(z)$ by a Dirac δ distribution: $q(z) = \delta(z - z^*)$ which essentially assumes that the posterior $P_{\mathcal{M}}(z \mid$

$\{(X_k, Y_k)\}_{k=1}^{K}, X)$ is highly concentrated at the optimal latent variable z^*. By the sifting property of the Dirac δ,

$$\int_z \delta(z - z^*) f(z)\, dz = f(z^*),$$

for any well-behaved function $f(z)$.

Substituting this into the ELBO yields

$$\mathcal{L}(q) = \int_z \delta(z - z^*) \times \log \frac{P_{\mathcal{M}}(Y \mid z, X)\, P_{\mathcal{M}}(z \mid \{(X_k, Y_k)\}_{k=1}^{K}, X)}{\delta(z - z^*)}\, dz$$

$$= \log \frac{P_{\mathcal{M}}(Y \mid z^*, X)\, P_{\mathcal{M}}(z^* \mid \{(X_k, Y_k)\}_{k=1}^{K}, X)}{\delta(0)}.$$

Although $\delta(0)$ is formally an infinite constant, it does not depend on z^* and can therefore be ignored during optimization. Maximizing the ELBO then amounts to optimizing

$$\mathcal{L}(z) = -\log P_{\mathcal{M}}(Y \mid z^*, X) - \log P_{\mathcal{M}}(z^* \mid \{(X_k, Y_k)\}_{k=1}^{K}, X)$$

E Dataset Details

Dataset Name	Template	Answer	Train	Test	Metric
AGNews	""{Sentence}" What is this text about? World, Sports, Business, or Technology?", "{Answer}"	'World', 'Sports', 'Business', 'Technology'	120,000	7,600	Accuracy
BoolQ	"{Sentence1} Can we conclude that {Sentence2}?", "{Answer}"	'No', 'Yes'	9,427	3,270	Accuracy
MultiRC	"{Sentence1} Question: "{Sentence2}" Response: "{Sentence3}" Does the response correctly answer the question?", "{Answer}"	'No', 'Yes'	27,243	4,848	F1
RTE	"{Sentence1} Based on the paragraph above can we conclude that "{Sentence2}"? Yes or No?","Answer"	'Yes', 'No'	2,490	277	Accuracy
SNLI	"If "{Sentence1}", does this mean that "{Sentence2}"? Yes, No, or Maybe?", "{Answer}"	'Yes', 'Maybe', 'No'	549,367	9,824	Accuracy
PAWS	"{Sentence1} {Sentence2} Do these sentences mean the same thing?", "{Answer}"	'No', 'Yes'	49,401	8,000	Accuracy
QQP	""{Sentence1}" "{Sentence2}" Would you say that these questions are the same?", "{Answer}"	'No', 'Yes'	363,846	40,430	Accuracy
Sentiment140	"{Sentence} What is the sentiment of this tweet?", "{Answer}"	'Negative', 'Positive'	1,600,000	359	Accuracy
SST2	"Review: "{Sentence}" Is this movie review sentence negative or positive?", "answer"	'Negative', 'Positive'	67,349	872	Accuracy

Table 6: Classification Task Dataset Details.

Dataset Name	Template	Answer	Train	Test	Metric
Winogrande	"How does the sentence end? {Sentence}", "{Answer}"	'A', 'B'	40,398	1,267	Accuracy
OpenBookQA	"{Sentence1} {Sentence2}", "{Answer}"	'A', 'B', 'C', 'D'	4,957	500	Accuracy
COPA	""{Sentence1}" What is the {Sentence2}?", "{Answer}"	'A', 'B'	400	100	Accuracy
HellaSwag	"What happens next in this paragraph? {Sentence}", "{Answer}"	'A', 'B', 'C', 'D'	39,905	10,042	Accuracy

Table 7: Multi-choice Task Dataset Details.

Dataset Name	Template	Train	Test	Metric
AESLC	"What is the subject line for this email? {Sentence}", "{Label}"	13,181	1,750	ROUGE-L
Gigaword	"Write a short summary for this text: {Sentence}", "{Answer}"	2,044,465	730	ROUGE-L
SQuAD v1	"Please answer a question about the following article about {Sentence}: {Sentence1} {Sentence2}", "{Answer}"	87,599	10,570	Exact Match
CommonGen	"Concepts: {Sentence}. Write a sentence that includes all these words.", "{Answer}"	67,389	4,018	ROUGE-L
DART	"Triple: {Sentence} What is a sentence that describes this triple?", "{Answer}"	62,659	2,768	ROUGE-L
E2E NLG	"Attributes: {Sentence}. Produce a detailed sentence about this restaurant.", "{Answer}"	33,525	1,847	ROUGE-L

Table 8: Generation Task Dataset Details.