More is not always better? Enhancing Many-Shot In-Context Learning with Differentiated and Reweighting Objectives

Xiaoqing Zhang[1,2*] **Ang Lv**[1] **Yuhan Liu**[1] **Flood Sung**[2]
Wei Liu[3] **Jian Luan**[3] **Shuo Shang**[4] **Xiuying Chen**[5†] **Rui Yan**[1,6,7†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China [2]MoonshotAI [3]MiLM Plus, Xiaomi Inc.
[4]University of Electronic Science and Technology of China [5]Mohamed bin Zayed University of Artificial Intelligence
[6]School of Artificial Intelligence, Wuhan University
[7]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MoE
{xiaoqingz, anglv, yuhan.liu, ruiyan}@ruc.edu.cn floodsung@moonshot.cn
{liuwei40,luanjian}@xiaomi.com jedi.shang@gmail.com xy-chen@pku.edu.cn

Abstract

Large language models (LLMs) excel at few-shot in-context learning (ICL) without requiring parameter updates. However, as ICL demonstrations increase from a few to many, performance tends to plateau and eventually decline. We identify two primary causes for this trend: the suboptimal negative log-likelihood (NLL) optimization objective and the incremental data noise. To address these issues, we introduce *DrICL*, a novel optimization method that enhances model performance through *Differentiated* and *Reweighting* objectives. Globally, DrICL utilizes differentiated learning to optimize the NLL objective, ensuring that many-shot performance surpasses zero-shot levels. Locally, it dynamically adjusts the weighting of many-shot demonstrations by leveraging cumulative advantages inspired by reinforcement learning, thereby mitigating the impact of noisy data. Recognizing the lack of multi-task datasets with diverse many-shot distributions, we develop the *Many-Shot ICL Benchmark* (ICL-50)-a large-scale benchmark of 50 tasks that cover shot numbers from 1 to 350 within sequences of up to 8,000 tokens-for both fine-tuning and evaluation purposes. Experimental results demonstrate that LLMs enhanced with DrICL achieve significant improvements in many-shot setups across various tasks, including both in-domain and out-of-domain scenarios. We release the code and dataset hoping to facilitate further research in many-shot ICL[1].

1 Introduction

In-context learning (Brown et al., 2020) enables models to quickly adapt and address specific issues by utilizing contextual cues, improving adaptability and generalization. With the expansion of the context length in advanced LLMs, the ability to process text lengths up to 1 million tokens allows LLMs to accept increasingly more demonstrations. The ICL scenarios with hundreds or thousands of shots are called many-shot learning (Agarwal et al., 2024). However, many-shot does not always result in better performance than few-shot. Some models exhibit a linear decrease in ICL capabilities with the increase in ultra-long text lengths (Liu et al., 2024a). As shown in Figure 1, we present the accuracy variations of Mistral-7B-Instruct-v0.2 on the CLSClusteringS2S dataset (Li et al., 2022). As the number of ICL examples increases, models' performance exhibits a trend of rising and then falling.

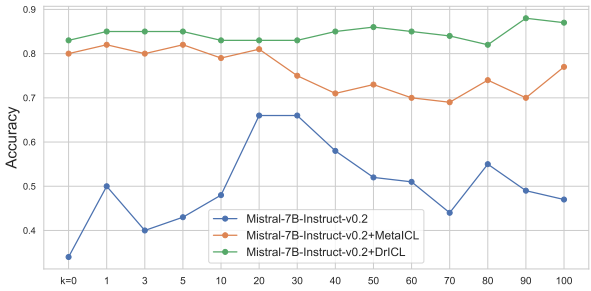

Figure 1: The performance trend of LLMs across different k-shots scenarios. k refers to the number of demonstration examples provided to LLMs, "+MetaICL" uses MetaICL for fine-tuning, while "+DrICL" uses our DrICL strategy.

We summarize two possible factors based on our preliminary study and previous works. The first factor is the training objective. As Agarwal et al. highlights, while the straightforward NLL decreases during testing with ICL, performance on many downstream tasks also deteriorates. The second factor is the increasing noise with the large number of demonstrations. Long et al.; Gao et al. demonstrate that the effectiveness of ICL heavily depends on the quality of the demonstrations. While in many-shot scenarios, utilizing a large number of high-quality demonstrations presents significant

*This work was done during the internship at Moonshot AI.
†Corresponding authors.
[1]https://github.com/xiaoqzhwhu/DrICL

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30539–30552

challenges, such as the huge workload of creating them and the difficulty of domain adaptation. Existing few-shot ICL methods do not address these issues, making them unsuitable for many-shot scenarios (Zhang et al., 2024; Li et al., 2023; Agarwal et al., 2024; Bertsch et al., 2024).

To address the above factors, we propose DrICL, enhancing many-shot in-context learning with a refined fine-tuning objective. For the first factor, we propose differentiated learning to deal with the trade-off between many-shot and zero-shot scenarios from a *global* perspective. During differentiated learning, we ensure that the performance on many-shot demonstrations surpasses that on zero-shot demonstrations. This approach promotes the model's deeper understanding of contextual cues, encouraging it to leverage contextual information effectively. For the second factor, we find the noise in the demonstrations is reflected in the sharp increase in loss observed for certain samples, which disrupts the training process. Inspired by reinforcement learning, we propose an advantage-based reweighting method to reduce focus on noise in many-shot demonstrations from a *local* perspective. In reinforcement learning, the "advantage function" is essential for assessing the value of actions beyond the average expected return, effectively directing the policy to choose actions that are predicted to yield the highest future rewards (Baird, 1994). Similarly, just as the advantage function helps identify actions that yield higher returns than the average, we extend this concept by using cumulative advantage to adjust the weights of demonstrations. This adjustment ensures that demonstrations with extreme loss fluctuations do not disproportionately influence the model or disrupt stable learning. The cumulative advantage for each demonstration is calculated based on the loss of the current demonstration and the losses of sampled demonstrations from preceding ones. We introduce a sampling window to ensure a balanced consideration of previous demonstrations. Each sequence is divided into multiple reweighting windows, and for each demonstration in a reweighting window w, the preceding window $w-1$ serves as the sampling window. Finally, we incorporate the cumulative advantage into the negative log-likelihood (NLL) computation, resulting in an advantage-based training objective.

In addition to the two challenges mentioned above, another significant obstacle is the lack of sufficient multi-task training data that spans a wide range of task numbers. Such data is crucial for studying the effects of ICL across many-shot scenarios. We present ICL-50, the largest dataset to study many-shot ICL, encompassing 50 tasks across 7 task types, and a total of over three million samples. The maximum number of shots achievable varies depending on the model, allowing for comprehensive investigations into many-shot ICL, including research on fine-tuning and inference. We categorize ICL-50 into different subsets based on task types, including in-domain and out-of-domain tasks, to fully assess the model's ICL capabilities in many-shot scenarios.

In summary, this paper identified two challenges in avoiding ICL's decreasing performance under Many-shot scenarios: suboptimal training objective, and incremental data noise. We propose "differentiated learning" and "advantaged-based reweighting" to address these challenges, respectively. We further propose the largest ICL-50 to support our training, evaluation as well as further studies. We experiment DrICL on the ICL-50 with open-source LLMs, showing stable performance both in-domain and out-of-domain under many-shots. All these points form the major contribution of this paper.

2 Related Work

In-context Learning. In-context learning allows models to execute downstream tasks without the need for parameter updates, enabling language models to serve as a universal tool for a variety of tasks. As the number of examples supplied to LLMs grows, supplementary strategies become essential to bolster the model's ICL capabilities. For instance, Anil et al. (2024) employ multi-example prompts, which can accommodate up to 256 demonstrations, to overcome the inherent limitations of language models. Hao et al. (2022) propose the structured prompting method to overcome length restrictions and extend in-context learning to thousands of examples. Li et al. (2023) use a customized model architecture to support the expansion of contextual examples to 2,000, and (Agarwal et al., 2024) utilize reinforced ICL and unsupervised ICL to extend the scope of contextual examples to 8,192. Unlike their work, we enhance the ICL capability of LLMs by improving the model's parameters rather than the form of contextual examples.

Instruction Tuning of LLMs. Instruction tuning has become an effective technique for enhanc-

ing the capabilities and controllability of LLMs (Zhang et al., 2023). In the domain of ICL, studies like MetaICL (Min et al., 2022) , IAD (Liu et al., 2024b), and PEFT (Bertsch et al., 2024) have demonstrated that fine-tuning LLMs with both small and large demonstration sizes, denoted as k, lead to improved ICL performance. Despite their studies being confined to a modest quantity of tuning data—capped at 10,000 entries—there is an evident necessity for deeper research into how ICL performs when scaled up with more extensive datasets. Consequently, we introduce ICL-50, a significantly larger dataset, designed to delve into the strategies for amplifying ICL's potential.

LLM Data Reweighting. As LLMs rapidly advance, the application of data reweighting in training has become increasingly prevalent. In the pre-training stage, SoftDedup significantly improves training efficiency by selectively reducing the sampling weight of data with high commonness through a soft deduplication method, rather than removing them to increase the integrity of the dataset (He et al., 2024). ScaleBiO reweights the data of LLMs by filtering irrelevant data samples and selecting informative samples, demonstrating its effectiveness and scalability across models of different sizes on tasks such as data denoising, multilingual training, and instruction tuning (Pan et al., 2024). In the ICL scenario, Yang et al. (2023) propose WICL to enhance the performance of ICL by assigning optimal weights to demonstration samples in the inference. Unlike other works, we set the weights during the training process based on the positions of multiple examples in a sequence.

3 DrICL

In this work, we propose the DrICL learning framework, which adjusts the weights of demonstrations and integrates reweighting within differentiated objectives, as illustrated in Figure 2. In DrICL, we organize training data through many-shot and zero-shot demonstrations. By simultaneously training the sequence of many-shot and zero-shot with a differentiated objective, we strengthen the model's overall ICL capability. At the same time, to further reduce the noise of demonstrations in many-shot scenarios, we introduce a weighted training objective towards different samples in the many-shot demonstrations. By sampling the model's performance under different demonstrations, we calculate the cumulative advantage gained as the number of demonstrations increases and use this cumulative advantage to adjust the learning process. Below, we show the components of the DrICL framework from both a global and local perspective, as well as the learning strategy.

3.1 Global Perspective: Differentiated Learning

We apply differentiated learning for the trade-off of many-shot and zero-shot sequences due to differing sample lengths, where longer sequences might introduce more noise. We expect that after refining the learning objectives, the model can still perform well in scenarios with numerous demonstrations, longer samples, and potentially noisy backgrounds. In each iteration, we sample K pairs of examples (x_k, y_k) from the training dataset, where k ranges from 1 to K. Then, we concatenate the examples x_k and their corresponding labels y_k, and the instruction I generated by GPT-3.5-turbo for the current task as the input sequence $S_K = \{I; x_1 y_1 x_2 y_2 \ldots x_K y_K\}$. We train the model to predict the label y_k of the k-th example based on the instruction and the features and labels of the previous $k-1$ examples. The training objective of the model is to minimize the NLL loss $\mathcal{L}_{\text{NLL}}$, with the previous $k-1$ examples as the training examples and the k-th example as the test example. This training method helps the model learn in context during the inference stage. We organize the number of demonstration examples according to k. When $k > 0$, we perform many-shot instruction-tuning, and when $k = 0$, we perform zero-shot instruction-tuning. During our training process, we expect that different examples of the same training sequence S_k can serve as helpful contexts C_h for each other. In the absence of context, we define C_{none}, so we update the original NLL loss combined with the additional objective for many-shot and zero-shot as follows:

$$\mathcal{L}_{\text{many-shot}} = \mathcal{L}_{\text{NLL}}(\text{LLM}(C_h, Q; \theta), A_{gt}),$$
$$\mathcal{L}_{\text{zero-shot}} = \mathcal{L}_{\text{NLL}}(\text{LLM}(C_{\text{none}}, Q; \theta), A_{gt}),$$

where Q is the input question to the model, and A_{gt} is the corresponding ground truth answer. We utilized many-shot data as Q and transformed it into a degraded zero-shot format using the Parallel Context Windows (PCW) method (Ratner et al., 2022). PCW works by masking many-shot sequences to generate a zero-shot sequence, effectively enabling us to leverage both formats. In our implementation,

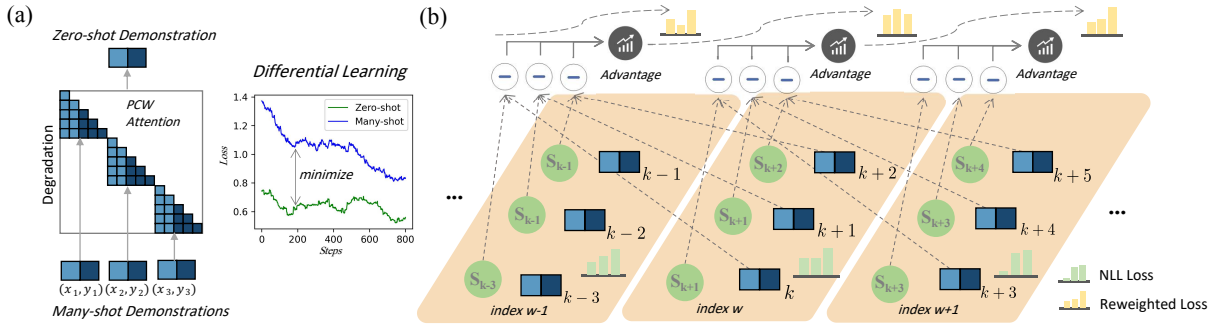

Figure 2: The DrICL Training Framework. (a) The global differentiated learning for many-shot and zero-shot demonstrations. (b) The local advantage-based reweighting method assigns differential weights to demonstrations in window w with window size $|W| = 3$ and sampling size $|S| = 1$, utilizing the cumulative advantage from the preceding window $w - 1$.

PCW was used solely to simplify the input for coding purposes. We aim to simultaneously optimize these two losses, such that $\mathcal{L}_{\text{many-shot}} < \mathcal{L}_{\text{zero-shot}}$. A lower many-shot loss signifies that the model has more effectively mastered in-context learning, thus enhancing its ability to accurately predict A_{gt}.

We have the following differentiated objectives:

$$\mathcal{L}_{\text{diff}} = (1 + \alpha) * \mathcal{L}_{\text{many-shot}} + (1 - \alpha) * \mathcal{L}_{\text{zero-shot}},$$

where α is the hyperparameter that controls the trade-off between many-shot and zero-shot.

3.2 Local Perspective: Advantage-based Reweighting

After global Differential Learning, we noticed that loss fluctuates at certain k-shot points instead of decreasing consistently, suggesting some samples affect the model's context significantly, possibly introducing noise. To address this, we introduced a reweight mechanism that adjusts weights based on performance differences between adjacent windows, giving higher weights to samples with larger differences, and helping the model adapt to dynamic contexts. The model optimizes the weights of demonstration data, continuously balancing exploration and data utilization to achieve a rapid and stable ICL performance. Below, we describe the overall process from three aspects: importance sampling, advantage functions, and reweighting.

3.2.1 Importance Sampling

Importance sampling adjusts weights to reduce bias and imbalance from noisy data. Here, we use the training model's loss on samples to calculate their importance weights. For each training sequence $S_K = \{x_1 y_1 x_2 y_2 \ldots x_K y_K\}$, we calculate the loss $\mathcal{L}_{\text{many-shot}_k}$ generated by the sequence $\{x_1 y_1 x_2 y_2 \ldots x_k\}$ at the current k-th position to represent the features of S_k. Our goal is to weight examples by their significance, emphasizing critical instances and reducing focus on less important ones.

To prevent an undue focus on specific parts of the data, we introduce a reweighting window, designed to segment the sequence into multiple parts. Each window is intended to handle a portion of the sequence with a total length of K. The sequence is segmented into $\lfloor \frac{K}{W} \rfloor$ equal windows, each with a size of W. For the k-th demonstration we have the reweighting window index w as follows:

$$w = \left[\lfloor \frac{k}{W} \rfloor \times W : \left(\lfloor \frac{k}{W} \rfloor + 1 \right) \times W \right].$$

We designate the preceding window $w - 1$ as the sampling window, to select $|S|$ demonstrations for those in the reweighting window w, compiling these into a set S. The demonstrations within set S are then utilized for further training, leveraging accumulated benefits to enhance learning.

$$w - 1 = \left[\left(\lfloor \frac{k}{W} \rfloor - 1 \right) \times W : \lfloor \frac{k}{W} \rfloor \times W \right].$$

We define a target distribution $p(x)$ and an importance distribution $q(x)$ with their probability density functions to achieve set S. Specifically, for each training sample S_k and feature vector $\mathcal{L}_k$ of the k-th demonstrations, we use the ratio of the values of the target distribution $p(x)$ and the importance distribution $q(x)$ to calculate the weight $weight_k$ for the k-th demonstration in S_k and select the top $|S|$ samples with the highest weights:

$$weight_k = \frac{p(\mathcal{L}_{\text{many-shot}_k})}{q(\mathcal{L}_{\text{many-shot}_k})}.$$

Through these steps, we calculate the weights of important samples and select the top $|S|$ representative samples from the given sample distribution.

3.2.2 Advantage Functions

To assess the model's cumulative advantages as the k-shots grow, we select the sample set S for the k-th instance within the weighting window w. Subsequently, we determine the average loss of the sampling window $w-1$ using the formula:

$$\mathcal{L}_{\text{sampling}_{w-1}} = \frac{1}{|S|} \sum_{\text{instance}_i \in S} \mathcal{L}_{\text{instance}_i}.$$

The reward is defined as the difference between the loss of the instance at the current position k and the average loss of the instances in window $w-1$:

$$\mathcal{R}_k = \mathcal{L}_{\text{many-shot}_k} - \mathcal{L}_{\text{sampling}_{w-1}}$$

Here, $\mathcal{L}_{\text{sampling}_{w-1}}$ represents the performance of the model on all sampled instances before window w. It denotes the model's performance with fewer than k shots, whereas $\mathcal{L}_{\text{many-shot}_k}$ signifies the model's performance with k shots. Next, we define the accumulated advantages to measure the strategy's performance in different positions k:

$$\mathcal{A}_k = \exp(\mathcal{R}_k/\gamma),$$

where γ is a temperature parameter used to adjust the sensitivity of the rewards. The exponential increase in the advantages metric strengthens positive rewards while suppressing negative rewards, guiding the model to select strategies that bring significant performance improvement.

3.2.3 Reweighting

In the DrICL framework, we select important samples in the previous window and calculate the reward that measures the model's performance in different positions to update the NLL loss for many-shot scenarios. We adjust the overall training objective of the many-shot sequence as follows:

$$\mathcal{L}_{\text{many-shot}} = \frac{1}{k} \sum_{k} \mathcal{L}_{\text{many-shot}_k} * \mathcal{A}_k.$$

By introducing the reweighting mechanism, we can not only maintain the performance of the current demonstration but also further optimize the model through gradient descent, leading to improved long-term performance.

Algorithm 1 Differentiated and Reweighting In-Context Learning (DrICL)

Parameter: α, γ, S, W

Initialize training data D, total number of iterations T, set current iteration $t = 0$.
for t in T **do**
  for d=$x_1, y_1, x_2, y_2, ..., x_K, y_K$ in D **do**
    Let the zero-shot loss $\mathcal{L}_{\text{zero-shot}} = 0$, many-shot loss $\mathcal{L}_{\text{many-shot}} = 0$.
    for k in K **do**
      Calculate the many-shot loss $\mathcal{L}_{\text{many-shot}_k}$.
      Mask the context of x_k by PCW attention to get the sequence zero-shot$_k$.
      Calculate the zero-shot loss $\mathcal{L}_{\text{zero-shot}_k}$.
      Set the window index $w = \lfloor k/W \rfloor$.
      Sample $|S|$ demonstrations from the window $w-1$ based on importance to form a validation set S.
      Calculate the sampling loss $\mathcal{L}_{\text{sampling}_{w-1}}$ for the demonstrations in S.
      Set the $\mathcal{R}_k = \mathcal{L}_{\text{many-shot}_k} - \mathcal{L}_{\text{sampling}_{w-1}}$.
      Update the cumulative advantage: $\mathcal{A}_k = exp\,(\mathcal{R}_k/\gamma)$.
      Assign the weighted loss: $\mathcal{L}_{\text{many-shot}_k} = \mathcal{L}_{\text{many-shot}_k} \times \mathcal{A}_k$.
      $\mathcal{L}_{\text{many-shot}} \mathrel{+}= \mathcal{L}_{\text{many-shot}_k}$.
      $\mathcal{L}_{\text{zero-shot}} \mathrel{+}= \mathcal{L}_{\text{zero-shot}_k}$.
    end for
    $\mathcal{L}_{\text{many-shot}} = \mathcal{L}_{\text{many-shot}}/K$.
    $\mathcal{L}_{\text{zero-shot}} = \mathcal{L}_{\text{zero-shot}}/K$.
    Update $\mathcal{L}_{\text{diff}}$ with hyperparameter α.
  end for
end for

3.2.4 Learning Strategy

The detailed process of the DrICL is presented in Algorithm 1. It enables the model to build upon prior knowledge at each iteration, avoiding uniform learning, thereby achieving progressive performance enhancement over long-term training.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

To delve into the exploration of many-shot ICL in LLMs, we need plenty of data across a wide range of k-shots. The datasets employed in MetaICL, like CROSSFIT (Ye et al., 2021) and UNIFIEDQA (Khashabi et al., 2020), have a notable constraint: their task lengths are generally centered around 100 tokens. This focus restricts the wide range of k-shot distributions, especially when the training sequence length is constrained. On the other hand, the LongICLBench dataset introduced by Li et al. (2024) significantly extends the length ranging from 1,000 to 50,000 tokens. Nonetheless, the dataset's limitation to a few hundred task instances renders it more suitable for inference rather than

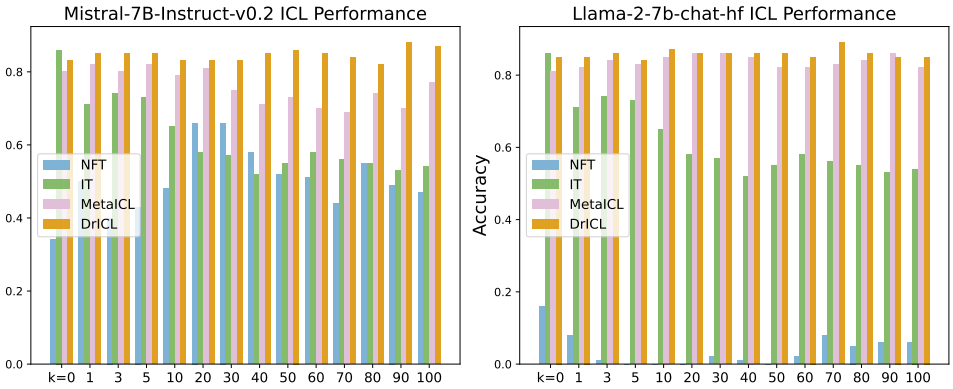

Figure 3: The performance with incremental k-shots for Mistral-7B-Instruct-v0.2 and Llama-2-7b-chat-hf on CLSClusteringS2S under different strategies. We focus on CLSClusteringS2S for its high k-shot count, enabling a broader evaluation of DrICL. Our DrICL consistently shows better performance with a diverse range of k.

Dataset	Models	k=0			k=1			k=3			k=5		
		D3	R1	B1	D3	R1	B1	D3	R1	B1	D3	R1	B1
XSUM^{id}	NFT	0.08	0.14	0.17	0.08	0.17	0.14	0.07	0.09	0.11	0.04	0.12	0.07
	IT	0.19	0.22	0.31	0.18	0.18	0.29	0.17	0.18	0.27	0.16	0.18	0.28
	MetaICL	0.19	0.23	0.30	0.15	0.22	0.31	0.18	0.23	0.30	0.18	0.22	0.29
	DrICL	**0.20**	0.22	**0.33**	**0.19**	0.21	**0.32**	**0.20**	**0.23**	**0.34**	**0.20**	**0.22**	**0.33**
CNN^{ood}	NFT	0.07	0.29	0.20	0.06	0.21	0.18	0.04	0.15	0.09	0.05	0.07	0.05
	IT	0.18	0.34	0.51	0.18	0.32	0.51	0.15	0.27	0.39	0.20	0.11	0.14
	MetaICL	0.19	0.34	0.51	0.19	0.35	0.51	0.18	0.34	0.51	0.18	0.33	0.47
	DrICL	**0.19**	**0.34**	**0.51**	**0.19**	0.34	**0.51**	**0.19**	0.31	**0.52**	0.19	0.31	**0.47**

Table 1: Summarization results on Llama-2-7b-chat-hf, where "id" denotes in-domain datasets and "ood" signifies out-of-domain datasets. **Bold** indicates that our model performs the best.

Dataset	Models	k=0	k=1	k=3	k=5	AVG	MAX
EcomRetrieval^{id}	NFT	0.19	0.10	0.09	0.01	0.10	0.19
	IT	0.93	0.06	0.12	0.19	0.33	0.93
	MetaICL	0.89	0.85	0.92	0.91	0.89	0.92
	DrICL	**0.93**	**0.87**	**0.94**	**0.93**	**0.92**	**0.94**
VideoRetrieval^{ood}	NFT	0.32	0.39	0.14	0.04	0.22	0.39
	IT	1.00	0.18	0.21	0.33	0.43	1.00
	MetaICL	0.89	0.96	1.00	1.00	0.96	1.00
	DrICL	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 2: Retrieval performance on Llama-2-7b-chat-hf. **Bold** indicates that our model performs the best.

extensive training. In light of these limitations, we have developed the ICL-50 dataset. It encompasses 7 tasks of diverse difficulty levels and includes 50 datasets with average sample lengths per task that vary from 10 to 14,000 tokens. With the number of samples extending from the hundreds into the hundreds of thousands, the ICL-50 dataset ensures a substantial volume of data suitable for training and inference. More details can be found in the Appendix.

4.1.2 Base Models

We perform our experiments using two foundational models, namely Llama-2-7b-chat-hf and Mistral-7B-Instruct-v0.2. The base models are trained by different paradigms: • **NFT** (Touvron et al., 2023; Jiang et al., 2023): The foundational models with No Fine-tuning. • **IT** (Wei et al., 2021): Instruction Tuning foundational models with zero-shot examples. • **MetaICL** (Min et al., 2022): Fine-tuning foundational models with many-shot examples.

4.1.3 Evaluation Metrics

In our evaluation, we employ accuracy for assessing the performance of question answering, clustering, logical reasoning, classification, and retrieval tasks. For the summarization task, we utilize Distinct of trigram tokens (D3), ROUGE for unigrams (R1), and BLEU for unigrams (B1) as our metrics. In the case of reranking tasks, we apply standard ranking metrics, including Precision at k $P@k$, Recall at k $R@k$, and Normalized Discounted Cumulative Gain $G@k$.

4.1.4 Implementation Details

For the Llama-2-7b-chat-hf model, we configured the hyperparameter α to 0.2, while for Mistral-7B-Instruct-v0.2, we set α to 0.4. We set the parameter γ to counteract the effects of weight explosion, and our experiments identified 11 as the optimal value

Dataset	Models	k=0	k=1	k=3	k=5	k=10	k=20	k=30	k=40	k=50	k=60	k=70	AVG	MAX
OpenbookQA^{id}	NFT	0.27	0.28	0.30	0.28	0.26	0.22	0.21	0.23	0.19	0.21	0.13	0.23	0.28
	IT	0.70	0.50	0.57	0.56	0.57	0.59	0.56	0.54	0.54	0.50	0.44	0.55	0.70
	MetaICL	0.59	0.59	0.64	0.63	0.72	0.75	0.77	0.78	0.78	0.79	0.70	0.70	0.79
	DrICL	0.69	**0.72**	**0.77**	**0.77**	**0.78**	**0.76**	0.76	0.76	**0.80**	0.76	**0.76**	**0.76**	**0.80**
ARC^{ood}	NFT	0.65	0.31	0.23	0.16	0.29	0.22	0.25	0.19	0.11	0.10	0.03	0.23	0.65
	IT	0.71	0.60	0.62	0.62	0.60	0.60	0.60	0.57	0.60	0.30	0.21	0.55	0.71
	MetaICL	0.67	0.71	0.76	0.78	0.76	0.78	0.78	0.82	0.82	0.78	0.71	0.76	0.82
	DrICL	**0.78**	**0.78**	**0.76**	**0.81**	**0.80**	**0.80**	**0.81**	0.78	0.79	0.75	0.67	**0.78**	0.81
CLSClusteringS2S^{id}	NFT	0.16	0.08	0.01	0.00	0.00	0.00	0.02	0.01	0.00	0.02	0.08	0.03	0.16
	IT	0.86	0.71	0.74	0.73	0.65	0.58	0.57	0.52	0.55	0.58	0.56	0.64	0.86
	MetaICL	0.81	0.82	0.84	0.83	0.85	0.86	0.86	0.85	0.82	0.82	0.83	0.84	0.86
	DrICL	0.85	**0.85**	**0.86**	**0.84**	**0.87**	**0.86**	**0.86**	**0.86**	**0.86**	**0.85**	**0.89**	**0.86**	**0.89**
ArxivClusteringS2S^{ood}	NFT	0.04	0.05	0.09	0.11	0.08	0.05	0.01	0.03	0.04	0.06	0.00	0.05	0.11
	IT	0.39	0.32	0.25	0.29	0.20	0.23	0.22	0.20	0.18	0.21	0.21	0.25	0.39
	MetaICL	0.35	0.41	0.36	0.33	0.42	0.36	0.37	0.39	0.33	0.37	0.39	0.37	0.42
	DrICL	0.34	0.35	**0.38**	**0.41**	0.36	**0.40**	**0.43**	0.36	**0.34**	**0.41**	0.35	**0.38**	**0.43**
TenkgnadClusteringS2S^{ood}	NFT	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.29
	IT	0.29	0.20	0.16	0.19	0.19	0.12	0.15	0.13	0.18	0.17	0.15	0.18	0.29
	MetaICL	0.24	0.19	0.23	0.21	0.24	0.23	0.23	0.26	0.27	0.27	0.25	0.24	0.27
	DrICL	0.23	**0.23**	**0.23**	**0.25**	**0.30**	**0.26**	**0.27**	0.23	**0.27**	0.26	**0.26**	**0.25**	**0.30**

Table 3: The performance of question answering, clustering, and classification tasks across various datasets on the Llama-2-7b-chat-hf model. **Bold** indicates that our model performs the best.

Dataset	Models	k=0	k=1	k=3	k=5	k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100	k=200	AVG	MAX
CLSClusteringS2S	NFT	0.34	0.50	0.40	0.43	0.48	0.66	0.66	0.58	0.52	0.51	0.44	0.55	0.49	0.47	0.00	0.47	0.66
	IT	0.86	0.71	0.74	0.73	0.65	0.58	0.57	0.52	0.55	0.58	0.56	0.55	0.53	0.54	0.07	0.58	0.86
	MetaICL	0.80	0.82	0.80	0.82	0.79	0.81	0.75	0.71	0.73	0.70	0.69	0.74	0.70	0.77	0.73	0.76	0.82
	DrICL	0.83	**0.85**	**0.85**	**0.85**	**0.83**	**0.83**	**0.83**	**0.85**	**0.86**	**0.85**	**0.84**	**0.82**	**0.88**	**0.87**	0.71	**0.84**	**0.88**
TweetSentimentExtraction	NFT	0.43	0.30	0.31	0.36	0.33	0.35	0.34	0.40	0.27	0.27	0.20	0.35	0.39	0.38	0.30	0.33	0.43
	IT	0.74	0.56	0.67	0.52	0.66	0.65	0.69	0.56	0.64	0.65	0.70	0.68	0.70	0.68	0.69	0.65	0.74
	MetaICL	0.75	0.77	0.80	0.77	0.78	0.79	0.78	0.81	0.73	0.75	0.79	0.76	0.73	0.70	0.51	0.75	0.81
	DrICL	**0.82**	**0.81**	**0.81**	**0.80**	**0.83**	**0.80**	**0.80**	0.78	**0.83**	**0.78**	**0.79**	**0.76**	**0.77**	**0.81**	**0.76**	**0.80**	**0.83**

Table 4: The performance variation of datasets with the highest k-shots on Mistral-7B-Instruct-v0.2. **Bold** indicates that our model performs the best.

for this parameter. We determined that the optimal sampling size for S is 1, with the reweighted window size W set at 10. For details on the experimental hyperparameter settings, please refer to Appendix B.1. For all training and evaluation tasks, we utilized 8 A100 GPUs.

Dataset	Models	k=0	k=1	k=3	k=5	k=10	k=20	AVG	MAX
GSM8K	NFT	0.28	0.16	0.11	0.07	0.01	0.02	0.11	0.28
	IT	0.31	0.26	0.26	0.21	0.26	0.16	0.24	0.31
	MetaICL	0.24	0.26	0.26	0.24	0.24	0.26	0.25	0.26
	DrICL	0.30	**0.28**	**0.31**	**0.27**	**0.32**	**0.26**	**0.29**	**0.32**

Table 5: The reasoning performance on GSM8K for the Llama-2-7b-chat-hf model. **Bold** indicates that our model performs the best.

4.2 Results of Tasks

We validate our method on 12 datasets with both in-domain and out-of-domain tasks. Figure 3 compares baseline models on the CLSClusteringS2S dataset across different k-shots of Llama-2-7b-chat-hf and Mistral-7B-Instruct-v0.2. Table 1 shows summarization performance, while Table 2 details retrieval metrics. Results for question answering, clustering, and classification are summarized in Table 3, and Table 5 presents reasoning task performance. Our reranking experiments are shown in Table 6. Given Mistral-7B-Instruct-v0.2's superior performance on sequences over 4,000 tokens, we compare baseline variations across k-shots for the tasks with the highest k, as detailed in Table 4.

As shown in Tables 1, 2, 3, 5, and 6, DrICL significantly improves performance across various tasks. While MetaICL shows substantial fluctuations in k-shot performance on datasets like OpenbookQA and ARC, DrICL maintains more stable results. The slight advantage of our method over Meta-ICL is due to its focus on optimizing many-shot loss. IT's performance declines with increasing context length, as it relies solely on zero-shot, which is less effective in many-shot scenarios. Additionally, Llama-2-7b-chat-hf's 4,000-token limit causes performance on the ARC dataset to drop from 0.82 to 0.78 when k exceeds 50. Under the DrICL framework, among the 12 datasets tested, $k = 0$ led to a performance decrease on 5 datasets, no change on 2, and improvement on 5. Overall,

performance improved by 0.5% with $k = 0$, remaining stable. For $k > 0$, datasets like CLSClusteringS2S showed continuous improvement, while DrICL effectively maintained performance stability as k increased across most datasets.

cMedQA	k=0			k=1		
	P@10	R@10	G@10	P@10	R@10	G@10
MetaICL	0.33	0.51	0.53	0.30	0.46	0.52
DrICL	0.31	0.48	**0.55**	**0.33**	**0.51**	**0.54**

Table 6: The comparison of ranking performance on the cMedQA dataset for the Llama-2-7b-chat-hf model, with a focus on zero-shot and one-shot settings due to its handling of examples with an extensive number of tokens. **Bold** indicates that our model performs the best.

4.3 In-Context Learning Analysis

4.3.1 Performance Tradeoff

We observe that the foundation models underperform on both datasets. After fine-tuning, the IT strategy achieves its best in the few-shot. MetaICL, benefiting from many-shot training data, performs well at larger k-shot levels but still shows significant fluctuations. In contrast, DrICL delivers more stable results, with accuracy steadily improving as k increases. DrICL not only outperforms MetaICL in many-shot scenarios but also demonstrates faster loss convergence, as shown in Figure 6(a) in the Appendix B.1, indicating its tradeoff of many-shot and zero-shot demonstrations.

Dataset	NFT	IT	MetaICL	DrICL
OpenbookQA	2.20E-03	3.90E-03	5.60E-03	8.00E-04
ARC	2.41E-02	2.10E-02	2.00E-03	1.40E-03
CLSClusteringS2S	2.40E-03	1.00E-02	3.00E-04	2.00E-04
ArxivClusteringS2S	1.00E-03	3.70E-03	8.00E-04	9.00E-04
TengkgnadClusteringS2S	7.00E-03	1.90E-03	5.00E-04	5.00E-04
TweetSentimentExtraction	3.30E-03	3.40E-03	4.90E-03	5.00E-04
GSM8K	8.50E-03	2.20E-03	1.00E-04	5.00E-04
XSUM	1.40E-03	2.20E-03	5.00E-05	5.00E-05
CNN	3.90E-03	2.30E-02	3.00E-03	3.70E-03
EcomRetrieval	4.10E-03	1.20E-01	7.00E-04	8.00E-04
VideoRetrieval	2.00E-02	1.10E-01	2.00E-03	0.00E+00
cMedQA	0.00E+00	2.40E-02	8.60E-03	9.40E-03
Average	**6.49E-03**	**2.71E-02**	**2.38E-03**	**1.56E-03**

Table 7: The performance variation of various datasets.

4.3.2 Performance Variance

Table 7 is the performance variance across the NFT, IT, MetaICL, and DrICL methods. We track the performance variance across all datasets with NFT(6.49E-03), IT(2.71E-02), MetaICL(2.38E-03), and DrICL(1.56E-03) as k varied. We prove that our method demonstrates the smallest deviation, indicating greater stability in performance as k-shot changes.

4.3.3 Data Noise Sensitivity

We compare DrICL with and without local reweighting by examining how loss variance trends for each k-shot demonstration during training. The reweighting window in DrICL reduces loss variance and effectively balances the impact of noisy data by appropriately weighting demonstrations. This reduction in sensitivity to data noise helps maintain stable performance as the number of demonstrations increases. For details of the noise variation please refer to Table 11 in Appendix B.2.

4.4 Ablation Studies

4.4.1 Hyperparameters

Figure 7, 8, and 6(b) illustrate the impact of varying the hyperparameters α, γ, and S on the training of Llama-2-7b-chat-hf and Mistral-7B-Instruct-v0.2. For details of the study of hyperparameters please refer to Appendix B.1.

WinoWhy	k=0	k=1	k=3	k=5	k=10	k=20	k=30	k=40	k=50	AVG	MAX
DrICL	0.30	**0.51**	**0.5**	**0.53**	**0.55**	**0.57**	0.48	**0.63**	**0.52**	**0.51**	**0.63**
DrICL w/ W=1	0.47	0.46	0.47	0.51	0.51	0.49	0.45	0.58	0.43	0.49	0.58
DrICL w/o global	0.45	0.43	0.46	0.41	0.52	0.51	0.52	0.50	0.43	0.47	0.52
DrICL w/o local	0.43	0.52	0.44	0.47	0.51	0.53	0.33	0.42	0.33	0.44	0.52

Table 8: The ablation results with different settings.

4.4.2 Global and Local Contribution

Table 8 displays the outcomes of DrICL when applying only global strategies or local strategies exclusively to the WinoWhy dataset. The results show that refining learning objectives via a global hyperparameter to trade off the performance and the local reweighting of demonstration examples can boost the LLMs' ICL capabilities.

4.4.3 Analysis of Window Size

Table 8 shows that increasing the window size improves performance. As the sequence length grows, the number of k-shots also increases. Relying only on data from position $k-1$ based on previous $k-1$-shot demonstrations can cause significant variability, amplifying the impact of data fluctuations. Expanding the sampling range helps mitigate this effect. We also tested sampling from positions 0 to $k-1$, but found the model preferentially selected certain data points, which didn't fully reflect the model's overall performance. As a result, we selected a window size of 10.

5 Conclusions

To enhance the ICL capacity as context lengths grow and demonstration k-shots rise, we introduce the DrICL algorithm to tackle the inaccurate objective and noise. This innovative method strategically calibrates global training goals to prioritize many-shot examples over zero-shot ones and employs local reweighting of many-shot instances using cumulative advantages as dynamic rewards, steering the model toward effective learning trajectories. To substantiate the effectiveness of our approach, we have curated and released the ICL-50 dataset, characterized by its diverse tasks and a broad spectrum of text lengths and quantities. Our method demonstrates notable enhancements in both in-domain and out-of-domain tasks. We anticipate that our research will stimulate further exploration into ICL's potential and contribute to the advancement of LLM performance.

Limitations

In this work, we balance the samples in the training set, but we have not fully analyzed the algorithm's robustness across datasets of varying sizes. As a result, DrICL's performance may vary when applied to datasets of different scales, which we plan to explore in future work. Regarding window size design, we used a uniform size for all samples. However, tasks with varying sample lengths may result in oversampling for short-text tasks and undersampling for long-text tasks. To address this, we plan to implement dynamic window sizes based on sample lengths to ensure balanced representation for both short and long samples.

Acknowledgments

This work is also supported by the Public Computing Cloud, Renmin University of China and by fund for building worldclass universities (disciplines) of Renmin University of China.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-shot jailbreaking. *Anthropic, April*.

Leemon C Baird. 1994. Reinforcement learning in continuous time: Advantage updating. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 4, pages 2448–2453. IEEE.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. 2024. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*.

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.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. 2024. On the noise robustness of in-context learning for text generation. *arXiv preprint arXiv:2405.17264*.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*.

Nan He, Weichen Xiong, Hanwen Liu, Yi Liao, Lei Ding, Kai Zhang, Guohua Tang, Xiao Han, and Wei Yang. 2024. Softdedup: an efficient data reweighting method for speeding up language model pre-training. *arXiv preprint arXiv:2407.06654*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. 2023. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. 2022. Csl: A large-scale chinese scientific literature dataset. *arXiv preprint arXiv:2209.05034*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yuhan Liu, Xiuying Chen, Gao Xing, Ji Zhang, and Rui Yan. 2024b. Iad: In-context learning ability decoupler of large language models in meta-training. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8535–8545.

Quanyu Long, Yin Wu, Wenya Wang, and Sinno Jialin Pan. 2024. Decomposing label space, format and discrimination: Rethinking how llms respond and solve tasks via in-context learning. *arXiv preprint arXiv:2404.07546*.

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. 2022. 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.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Rui Pan, Jipeng Zhang, Xingyuan Pan, Renjie Pi, Xiaoyu Wang, and Tong Zhang. 2024. Scalebio: Scalable bilevel optimization for llm data reweighting. *arXiv preprint arXiv:2406.19976*.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Parallel context windows for large language models. *arXiv preprint arXiv:2212.10947*.

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*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Zhe Yang, Damai Dai, Peiyi Wang, and Zhifang Sui. 2023. Not all demonstration examples are equally beneficial: Reweighting demonstration examples for in-context learning. *arXiv preprint arXiv:2310.08309*.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.

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*.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. Winowhy: A deep diagnosis of essential commonsense knowledge for answering winograd schema challenge. *arXiv preprint arXiv:2005.05763*.

Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. 2024. Batch-icl: Effective, efficient, and order-agnostic in-context learning. *arXiv preprint arXiv:2401.06469*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *arXiv preprint arXiv:2104.06598*.

A Dataset

A.1 Overview

ICLB dataset includes the following tasks:

•**QA**: MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), BoolQ (Clark et al., 2019), NarrativeQA (Kočiskỳ et al., 2018), TruthfulQA (Lin et al., 2021), OpenbookQA (Mihaylov et al., 2018), ARC (Clark et al., 2018), and QUAC (Choi et al., 2018).

•**Reasoning**: GSM8K (Cobbe et al., 2021), APPS (Hendrycks et al., 2021), MATH (Hendrycks et al., 2021), BABI (Weston et al., 2015), and AR-LSAT (Zhong et al., 2021).

•**Summarization**: XSUM (Narayan et al., 2018) and CNN/DailyMail (Hermann et al., 2015).

We refer to the MTEB (Muennighoff et al., 2022) and C-MTEB (Xiao et al., 2023) that contains the description of the following dataset.

•**Clustering**: ArxivClusteringS2S, ArxivClusteringP2P, BiorxivClusteringS2S, BiorxivClusteringP2P, MedrxivClusteringP2P, RedditClustering, RedditClusteringP2P, StackExchangeClustering, StackExchangeClusteringP2P, CLSClusteringS2S, CLSClusteringP2P, ThuNewsClusteringS2S, BlurbsClusteringS2S, BlurbsClusteringP2P, TenkgnadClusteringS2S, TenkgnadClusteringP2P.

•**Classification**: AmazonPolarity, AmazonReviews, Emotion, ToxicConversations, TweetSentimentExtraction, JDReview, MultilingualSentiment, OnlineShopping, Waimai and WinoWhy (Zhang et al., 2020).

•**Retrieval**: cMedQA, TREC-COVID, DuReaderRetrieval, EcomRetrieval, MMarco, MedicalRetrieval, T2R, and VideoRetrieval.

•**Reranking**: cMedQA and AskUbuntuDupQuestions.

A.2 Data Analysis

The statistics of data volume for each task can be referred to in Table 9, where the data volume for 50 tasks ranges from several hundred to hundreds of thousands of entries, providing ample data for the model's training and inference. The distribution of the number of tokens for tasks is between 10 and 14,000, as shown in Figure 4. When the many-shot fine-tuning sequence length is fixed, the k-shot number varies significantly. Figure 5 illustrates the k-shot distribution with a fixed training sequence length of 8,000, ranging between 0 and 350. When the training sequence length is increased to 32,000, the range of k-shot variation will exceed 1,000. This provides a solid data foundation for the performance study of ICL in many-shot scenarios.

Task Type	Task Name	Train	Test
QA	MMLU	99834	13985
	HellaSwag	39905	10042
	BoolQ	9427	100
	NarrativeQA	36208	0
	TruthfulQA	22434	0
	OpenbookQA	4957	500
	QUAC	83568	7354
	ARC	1096	106
Reasoning	APPS	5000	0
	MATH	7500	5000
	BABI	200000	20000
	GSM8K	7473	1319
	AR-LSAT	1630	230
Summarization	CNN	83321	9258
	DailyMail	197555	21951
	XSUM	204045	11334
Clustering	CLSClusteringP2P	81499	8501
	CLSClusteringS2S	83359	6641
	ThuNewsClusteringS2S	83816	6184
	ArxivClusteringP2P	135171	14829
	ArxivClusteringS2S	133190	16810
	BiorxivClusteringP2P	58185	6815
	BiorxivClusteringS2S	57893	7107
	BlurbsClusteringP2P	135018	14982
	BlurbsClusteringS2S	132972	17028
	MedrxivClusteringP2P	29337	3163
	RedditClustering	134749	15251
	RedditClusteringP2P	134391	15609
	StackExchangeClustering	132996	17004
	StackExchangeClusteringP2P	58584	6416
	TenkgnadClusteringP2P	38953	4110
	TenkgnadClusteringS2S	39293	3770
Classification	JDReview	3468	261
	MultilingualSentiment	90761	9239
	OnlineShopping	7675	325
	AmazonPolarity	89979	10021
	AmazonReviews	90145	9855
	Emotion	12124	3876
	ToxicConversations	45823	4177
	TweetSentimentExtraction	24189	3292
	Waimai	7697	303
	WinoWhy	1160	443
Retrieval	cMedQARetrieval	5898	804
	TREC-COVID	803	79
	DuReaderRetrieval	7996	865
	EcomRetrieval	757	139
	MMarco	5944	741
	MedicalRetrieval	829	78
	T2R	8958	1042
	VideoRetrieval	859	28
Reranking	cMedQAReranking	806	99
	AskUbuntuDupQuestions	295	45

Table 9: The statistics of each task dataset.

A.3 Data Deploy

For each task, we leverage GPT-3.5-Turbo to generate instructions. We segment our datasets into meta-train and meta-test sets, holding back data from one task per category for evaluating our method's ability to generalize to new data. For overall evaluation, we possess both in-domain and out-of-domain test sets in comparison to the meta-train data. For Classification, meta-train domains include online shopping, multilingual, sentiment analysis, and tox-

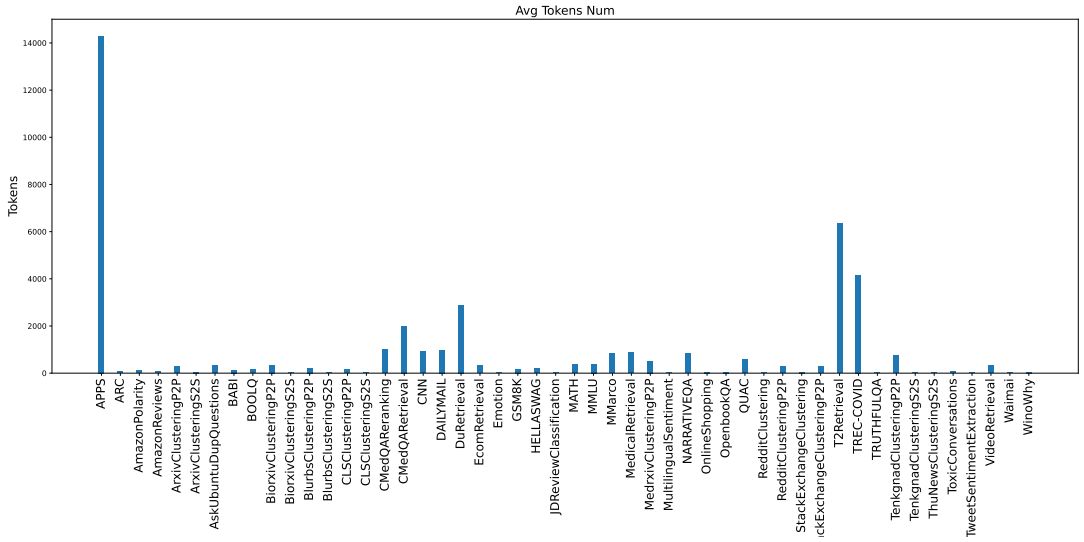

Figure 4: The token distributions of each task dataset.

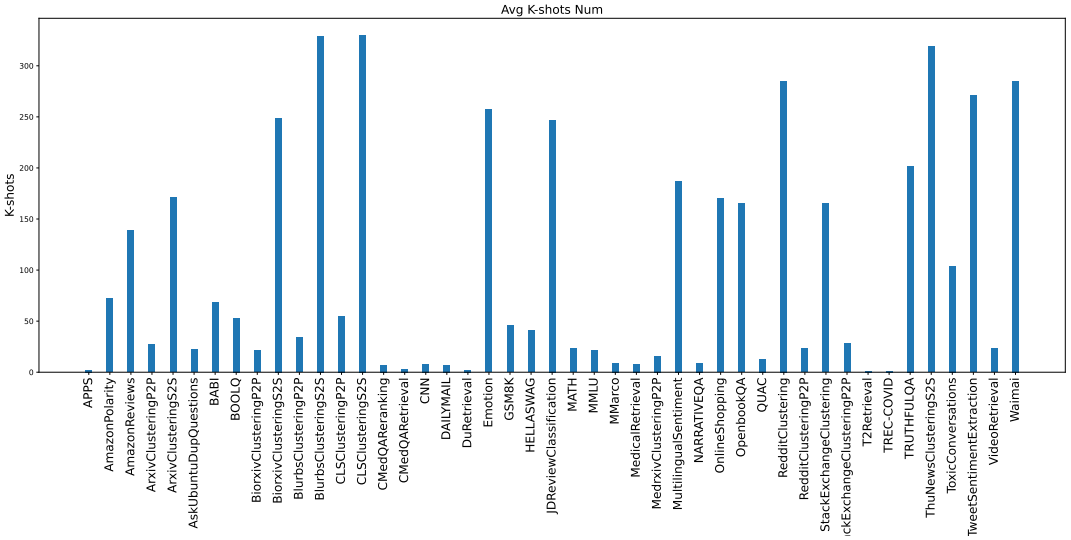

Figure 5: The k-shots distributions of each task dataset.

icity detection, while test domains extend to dining and common sense. For Reasoning, math-related domains are included in both meta-train and test sets. When assembling the training set, we ensure an equitable distribution of data among tasks, keeping the difference in data volume between any two tasks to no more than ten times, thereby enhancing model performance. We generate demonstrations with varying k-shots using the training set. For each task, we infer 100 randomly selected test set entries per k-shot, assessing the model's performance with different k-shot ranges from 0 to 350.

B Experiment Details

B.1 Evaluation

We evaluated our method on 12 datasets, each with 1,600 samples, totaling 19,200 test samples, and sampling rates ranging from 2% to 100%. For each dataset, we collected 16 types of demonstrations with various k-shot values, including 0, 1, 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, and 300. For reference, Table 10 presents the results from Table 3 for the total test set. The findings show that performance trends remain consistent across different sampling instances from each dataset.

B.2 Hyperparameters Study

α plays a crucial role in determining the model's performance, as illustrated in Figure 7, the variance

Dataset	Models	k=0	k=1	k=3	k=5	k=10	k=20	k=30	k=40	k=50	k=60	k=70	AVG	MAX
OpenbookQA^{id}	NFT	0.29	0.25	0.24	0.23	0.27	0.26	0.19	0.16	0.2	0.18	0.18	0.22	0.29
	IT	0.71	0.51	0.58	0.6	0.6	0.64	0.62	0.58	0.58	0.53	0.52	0.59	0.71
	MetaICL	0.63	0.65	0.68	0.69	0.73	0.75	0.76	0.76	0.76	0.76	0.74	0.72	0.76
	DrICL	**0.74**	**0.74**	**0.77**	**0.76**	**0.77**	**0.79**	**0.78**	**0.79**	**0.79**	**0.78**	**0.77**	**0.77**	**0.79**
CLSClusteringS2S^{id}	NFT	0.17	0.03	0.01	0	0	0	0.02	0.03	0.01	0.02	0.12	0.04	0.17
	IT	0.86	0.81	0.74	0.71	0.66	0.53	0.52	0.49	0.53	0.52	0.51	0.63	0.86
	MetaICL	0.82	0.85	0.84	0.86	0.85	0.85	0.83	0.85	0.85	0.84	0.83	0.84	0.86
	DrICL	0.85	**0.86**	**0.86**	0.84	**0.87**	**0.86**	**0.86**	**0.87**	**0.85**	**0.86**	**0.88**	**0.86**	**0.88**
ArxivClusteringS2S^{ood}	NFT	0.02	0.03	0.06	0.06	0.05	0.05	0.03	0.03	0.04	0.06	0.02	0.04	0.06
	IT	0.36	0.32	0.25	0.29	0.25	0.22	0.22	0.23	0.21	0.19	0.16	0.25	0.36
	MetaICL	0.35	0.39	0.35	0.35	0.34	0.37	0.37	0.39	0.38	0.36	0.33	0.36	0.39
	DrICL	0.33	0.34	**0.37**	**0.36**	**0.36**	**0.39**	**0.4**	0.38	0.36	**0.42**	**0.34**	**0.37**	**0.42**

Table 10: The evaluation on the whole test set of OpenbookQA, CLSClusteringS2S, and ArxivClusteringS2S. **Bold** indicates that our model performs the best.

in performance between zero-shot and many-shot scenarios is model-dependent.

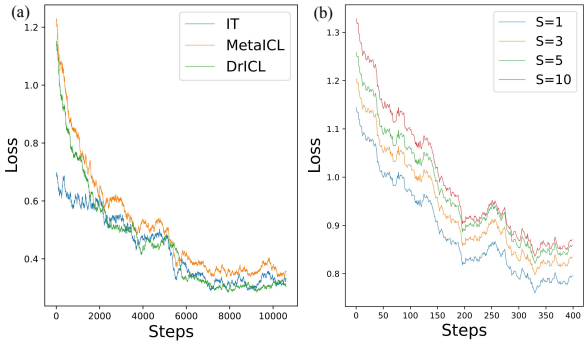

Figure 6: (a) The many-shot training loss of DrICL converges to a lower level compared to IT and MetaICL. (b) The optimal performance is when the parameter S is set to 1 on Llama-2-7b-chat-hf.

γ adjusts the sensitivity of the rewards and makes the training process stable. Figure 8 illustrates that the best setting of γ is 11.

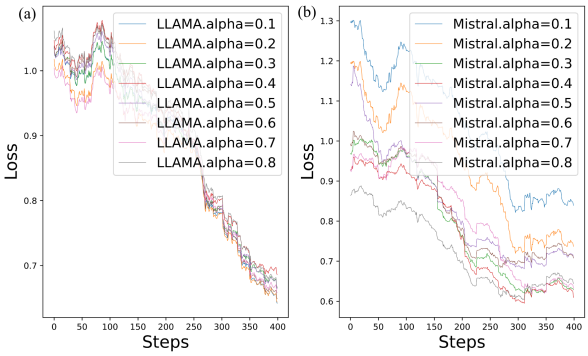

Figure 7: (a) The optimal performance is when the parameter α is set to 0.2 on Llama-2-7b-chat-hf. (b) The optimal performance on Mistral-7B-Instruct-v0.2 is achieved with α set to 0.4.

S represents the loss computed from three randomly sampled positions within the sampling window to calculate the reward. A high value of S leads to non-representative sampling. From our experiments in Figure 6(b), we found that the best training results were achieved with $S = 1$.

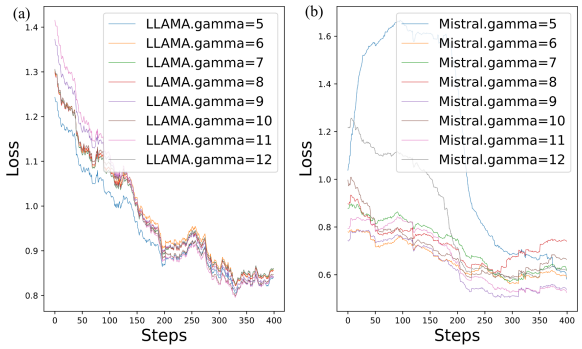

Figure 8: (a) and (b) show the optimal γ settings for Llama-2-7b-chat-hf and Mistral-7B-Instruct-v0.2, with both models achieving the best performance at $\gamma = 11$.

B.3 Data Noise Sensitivity

Table 11 illustrates the loss variance at different training stages with and without local reweighting.

Methods	Variance Across Training Progress				
	20%	40%	60%	80%	100%
DrICL w/o Local	4.90	1.40	0.93	0.59	1.80
DrICL	6.60	1.85	0.55	0.35	0.32

Table 11: The loss variance during the whole training process.