What Makes a Good Order of Examples in In-Context Learning

Qi Guo[♠] , **Leiyu Wang**[♦] , **Yidong Wang**[♠] , **Wei Ye**[♠†], **Shikun Zhang**[♠†]
[♠]National Engineering Research Center for Software Engineering, Peking University
[♦]National Key Laboratory for Novel Software Technology, Nanjing University
{qguo, leiyuwang}@smail.nju.edu.cn yidongwang37@gmail.com
{wye, zhangsk}@pku.edu.cn

Abstract

Although large language models (LLMs) have demonstrated impressive few-shot learning capabilities via in-context learning (ICL), ICL performance is known to be highly sensitive to the order of examples provided. To identify appropriate orders, recent studies propose heuristic methods to evaluate order performance using a set of unlabeled data. However, the requirement of in-domain data limits their utility in real-world scenarios where additional annotated data is challenging to acquire. Additionally, these dataset-based approaches are prone to being sub-optimal for a lack of consideration for individual differences. To address the problems, we first analyze the properties of performant example orders at both corpus level and instance level. Based on the analysis we propose **DEmO** to adaptively identify performant example order for each instance without extra data. DEmO works by filtering out a subset of orders featuring label fairness, then selecting the most influential order for each test instance. The employment of a content-free metric makes DEmO independent of in-domain data. Extensive experiments indicate the superiority of DEmO over a wide range of strong baselines. Further analysis validates the generalizability across various settings. Our code is available on https://github.com/GuoQi2000/DSICL.

1 Introduction

Large language models (LLMs) demonstrate impressive abilities (Dong et al., 2022; Zhao et al., 2023b; Wei et al., 2022) across a variety of natural language processing tasks through in-context learning (ICL). In ICL, provided with prompts containing several labeled examples, models are asked to make predictions on a new example (Brown et al., 2020). In contrast to traditional learning paradigms of full data fine-tuning, ICL requires no model parameter update and has emerged as a prevailing paradigm for using LLMs.

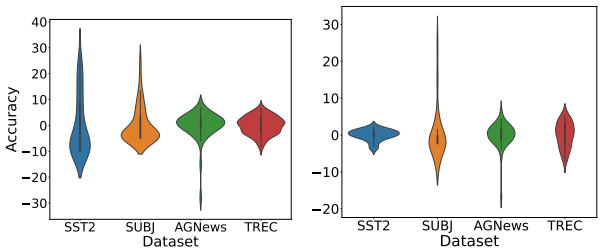

Figure 1: Zero-centered ICL accuracy of 50 random orders on Sheared LLaMA2 1.3B (left) 2023 and LLaMA2 7B (right) 2023.

However, the performance of ICL shows notable differences across various prompts (Ma et al., 2023). To enhance ICL performance, a line of studies has predominantly focused on selecting high-quality examples from a large dataset (Li et al., 2023; Yang et al., 2023; Wang et al., 2023; Zhao et al., 2023a). Nevertheless, access to a large in-domain dataset can be particularly challenging in certain fields such as medicine and finance (Alzubaidi et al., 2023). Hence, recent studies have focused on organizing examples in a reasonable order since the order of ICL can highly affect the performance (Chang and Jia, 2023). As illustrated in Figure 1, we conduct experiments on 4 classification task to show that the accuracy of different orders varies a lot across different models. One straightforward yet effective strategy is to directly evaluate the performance of different orders on a labeled validation set and select the most effective one. To reduce reliance on labeled data, several studies leverage heuristic metrics such as mutual information (Sorensen et al., 2022), global entropy and local entropy (Lu et al., 2022) to evaluate the order performance on an unlabeled dataset. Nevertheless, these methods still heavily rely on supplementary in-domain data for order assessment

[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 14892–14904

and fail in realistic few-shot scenarios, as defined by Perez et al. (2021), where only a handful of examples are available. In addition, since these methods operate at the corpus level without considering differences among instances, they tend to be sub-optimal (Wu et al., 2023).

In this paper, we aim to identify effective example orders for each test instance under realistic few-shot scenarios. Initially, we explore a fundamental question: *What constitutes a good example order for ICL?* We consider this question at both the corpus level and the instance level. We systematically evaluate the impact of corpus-level prediction distribution and instance-level influence on ICL performance. Specifically, we consider 50 random orders and evaluate them across four widely used text classification datasets to explore the correlations. Pilot experiments reveal that a reasonable example order is supposed to ideally (1) exhibit label fairness at the corpus level and (2) facilitate influential prediction at the instance level. *label fairness* means that the demonstration examples will not bias models towards certain labels. *influential prediction* means that the test instance should be dominant in the final prediction of models.

Motivated by these two insights, we introduce a novel two-stage ordering framework named **DEmO** (**D**ataset-free **E**xa**m**ple **O**rdering) that can be effortlessly implemented without extra data. To achieve label fairness while mitigating data dependency, a filtering stage is applied to pick out a candidate set of orders with balanced label distributions. In this process, a content-free input that replaces the test instance with a meaningless token is employed as a proxy for corpus-level prediction. Then to produce influential prediction at the instance level, we adaptive select candidate order that maximizes the output influence of the test instance during inference.

To verify the effectiveness, we conduct comprehensive experiments and analysis across a spectrum of established classification tasks. Experimental results reveal that our framework consistently outperforms existing dataset-based methods and exhibits generalability across various settings. Furthermore, our framework demonstrates the potential to be enhanced with access to a labeled validation set.

To summarize, our contributions include:

- To the best of our knowledge, we first explore the features of performant example orders at both corpus and instance levels. We introduce a general ordering framework to fill the gap between corpus-level and instance-level properties under realistic few-shot scenarios.
- Extensive experiments validate the superior performance of our framework over strong dataset-based baselines (e.g., 9.6% relative improvement of accuracy across nine datasets). With a validation set available, our framework further boosts the performance by 2.1%.

2 Related Works

Order sensitivity is observed universally across LLMs. With a fixed set of demonstration examples, the performance of different orders can vary from random guess to state-of-the-art (Lu et al., 2022). Recently, several studies have tried to determine performant orders. Their methods can be roughly categorized by whether an in-domain dataset is required. Typically, dataset-based approaches show superior performance to dataset-free methods by utilizing domain information but incur more computational overheads.

Dataset-free search: To counter the recency bias in prompts (i.e. LLMs tend to repeat the answer in the last example), Liu et al. (2022) organize demonstration examples by their embedding space similarity to the test instance, placing more similar examples closer to the end of prompts. Inspired by Solomonoff's general theory of inference (Solomonoff, 1964), Wu et al. (2023) employs an information-theoretic strategy to minimize the code length required to transmit task labels. However, these methods are tailored to specific demonstration selection techniques and lack general applicability.

Dataset-based search: Sorensen et al. (2022) opt for prompts with larger mutual information between the input and model output, fostering confident and diverse predictions. Lu et al. (2022) define local entropy and global entropy metrics to sequence examples, aiming for balanced predictions across pseudo samples generated by the LLM itself. In spite of the effectiveness of dataset-based methods, they exhibit notable drawbacks of high dependency on data and are limited in real-world scenarios. Compared with them, our framework shows superior performance without the need for additional data. In contrast, our framework additionally incorporates corpus-level information

to identify performant orders.

3 What Makes a Good Order for ICL

In this section, we explore the properties of performant orders at the corpus level and instance level respectively. We conduct our pilot experiments on four widely used classification datasets: SST-2 (Socher et al., 2013), Subj (Pang and Lee, 2004), TREC (Hovy et al., 2001), and AgNews (Zhang et al., 2015). Following a standard 4-shot setting, we consider 50 random permutations and evaluate them on a validation set containing 256 instances.

3.1 In-context learning Formulation

We first introduce the definition of example ordering in in-context learning. Given a set of demonstration examples $D = \{(x_i, y_i)\}_{i=1}^{n}$ and a n-element permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, a context containing all the examples is constructed as

$$C_\pi = \Omega(x_{\pi_1}, y_{\pi_1}) \oplus \ldots \oplus \Omega(x_{\pi_n}, y_{\pi_n}), \quad (1)$$

where $\oplus$ means concatenation operation and $\Omega(\cdot, \cdot)$ denotes a task-specific template that transforms a single instance (x_i, y_i) into natural language. $\Omega(\cdot, \cdot)$ includes a verbalization function $v(\cdot)$ that maps y_i into a label word (i.e. in natural language inference, the label "contradiction" can be mapped to the token “no”). During inference, the label of a test instance x_t is predicted as

$$y_{t,\pi} = \arg\max P(v(y)|C_\pi \oplus \Omega(x_t, *)). \quad (2)$$

In the following sections, $P(v(y)|C_\pi \oplus \Omega(x, *))$ is abbreviated as $p(y|x, C_\pi)$ for the sake of clarity.

3.2 Label Fairness at Corpus Level

We first explore the property at the corpus level. Considering the situation where a set of unlabeled data $D_t = \{x_{t_i}\}_{i=1}^{|D_t|}$ is available, a balanced prediction distribution is intuitively more desirable than one skewed towards specific labels. Following (Lu et al., 2022), we can directly leverage the concept of global entropy to quantitatively assess the balance in label predictions. Given a permutation π, the global entropy is defined as:

$$GLE(\pi) = -\sum_l \hat{p}_l \log \hat{p}_l, \quad (3)$$

where $\hat{p}_l = \frac{1}{|D_t|} \sum_i \mathbb{I}(y_{t_i,\pi} = l)$. The higher the entropy, the more evenly the predicted label distribution spreads across the unlabeled data, exhibiting label fairness at the corpus level.

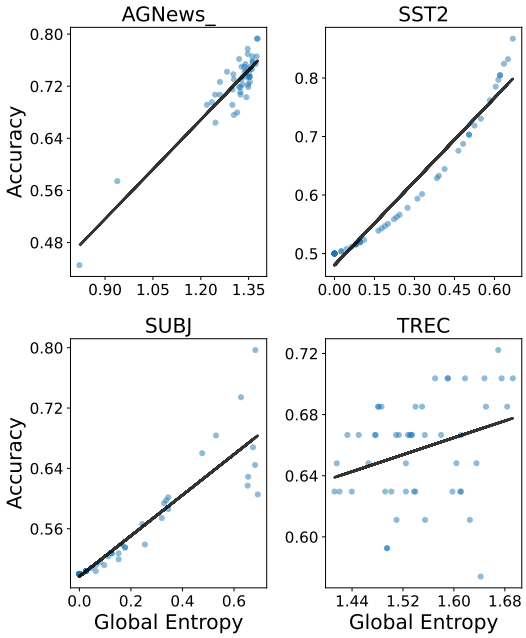

Figure 2: Each dot represents a permutation. The x-axis represents the entropy while the y-axis represents the average accuracy. Linear best fit lines are drawn to show overall trends.

We explore the relationship between global entropy and accuracy. As illustrated in Figure 2, a notable positive correlation is observed between them. Specifically, the accuracy on SST-2 tends to increase almost monotonically alongside entropy. In contrast, this positive correlation appears less pronounced on TREC. Such variation can be attributed to the imbalanced label distribution of TREC. However, considering the absence of prior knowledge of label distribution, **the assumption of label fairness at the corpus level is reasonable guidance for order search**.

3.3 Influential Prediction at Instance Level

Despite the strong positive correlation observed above, global entropy alone is insufficient to determine the optimal order. Take AGNews as an example: most orderings exhibit high global entropy, but their performances vary significantly, ranging from 68% to 80%. This variability indicates that only considering corpus-level information may lead to sub-optimal results.

Hence we delve deeper into the factors affecting the performance at the individual instance level. The concepts of influence have been previously applied in example selection scenarios (Nguyen

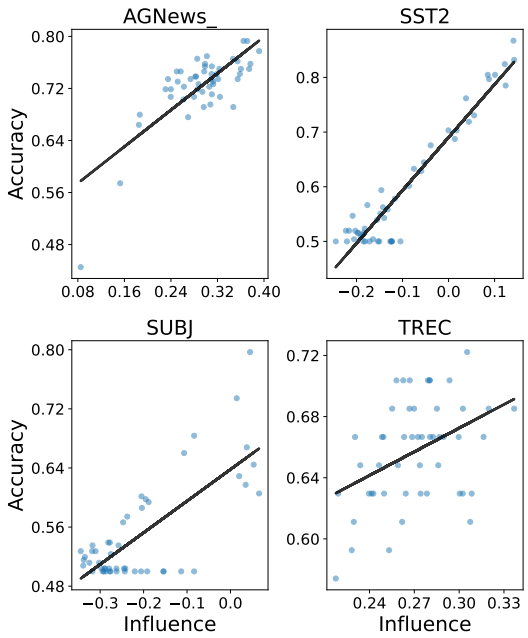

Figure 3: The relations between average influence and accuracy on the validation set.

and Wong, 2023) to measure the importance of demonstration examples. Similarly, we define an influence metric to quantify the impact of one test instance on the final prediction of LLMs and investigate its relationship with ICL performance. Mathematically, given a permutation π, the influence of test instance x_t is defined as:

$$I(x_t) = P(l|x_t, C_\pi) - E_x[P(l|x_t, C_\pi)], \quad (4)$$

where $l = \arg\max_y P(y|x_t, C_\pi)$, and $E_x[P(y = l|x_t, C_\pi)]$ is approximated by $\hat{p}_l$.

Operating under an intuitive assumption that an effective order should elevate the test instance's impact on the final prediction, we compute the average influence I across D_t for each permutation and note a consistent positive correlation. Orders demonstrating a high influence per test instance outperform those with lower average influences, **suggesting the utility of influence as an instance-level metric for identifying effective orderings.**

Rigorously, the influence metric can be defined as the probability difference between prompts containing x_t and the average of all the prompts omitting x_t. To be more detailed, the first item $P(l|x_t, C_\pi)$ can be viewed as guidance to reduce model uncertainty, and the second item $-E_x[P(l|x_t, C_\pi)]$ helps to find orders that counter the label prior in predictions.

Algorithm 1: Example Ordering

Input: examples set $D = \{(x_i, y_i)\}_{i=1}^{|D|}$; test set D_t; iteration num N; candidate set size K; template $\Omega(\cdot,\cdot)$; content-free token W
Output: Predictions for D_t.

Initialize candidate set $\Pi \leftarrow \{\}$.
for $i = 1, 2, \ldots, N$ **do**
  randomly sample a permutation π^i of D
  $E(\pi^i) = -\sum_y P(y|C_{\pi^i}) \log P(y|C_{\pi^i})$
  $\Pi \leftarrow \Pi \cup \{\pi^i\}$
end
$\Pi \leftarrow$ the top K of Π using $\{E(\pi^i)\}_{i=1}^N$
Predictions $\varepsilon \leftarrow \{\}$
for $x_t \in D_t$ **do**
  $\pi^* = \arg\max_{\pi \in \Pi} I(x_t)$
  $y_{t,\pi^*} = \arg\max_y P(y|x_t, C_{\pi^*})$
  $\varepsilon \leftarrow \varepsilon \cup \{y_{t,\pi^*}\}$
end
Return ε

4 Methodology

Based on the aforementioned insights in preliminary experiments, we propose a general two-stage ordering framework for ICL in realistic few-shot settings. We start by employing a content-free entropy metric to filter out permutations that are prone to generating imbalanced corpus-level predictions, yielding a refined set of candidate permutations. Subsequently, during the inference stage, we adaptively select the permutation with the highest influence for each test sample. Figure 4 gives an overview of our framework. The full algorithm is demonstrated in Algorithm 1.

4.1 Filtering with Content-Free Entropy

In real-world few-shot scenarios (Perez et al., 2021) where no additional data are available, it is impractical to directly apply Equation (3) to identify permutations yielding balanced predictions. To address this challenge, we turn to use a content-free output (Zhao et al., 2021) to approximate the corpus-level predictions:

$$E_x[P(y|x, C_\pi)] \approx P(v(y)|C_\pi \oplus \Omega(W, *)), \quad (5)$$

where W represents a content-free token such as '[MASK]'. Following Zhao et al. (2021), we select three tokens including '[MASK]', '[N/A]' and ''. We use the average probability of them as the approximation. By substituting the test instance with

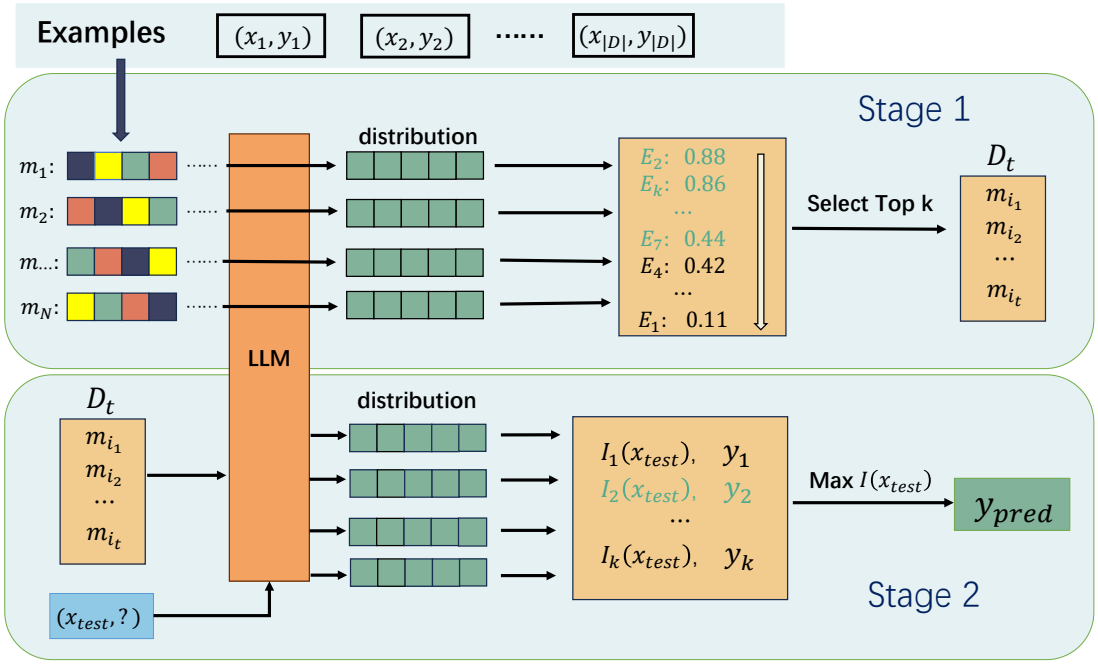

Figure 4: An overview of DEmO.

content-free tokens, the inference times required for a single permutation assessment are reduced to a constant. Let $P(y|C_\pi)$ denote the content-free output for the sake of clarity.

Similarly, we introduce a content-free entropy metric to evaluate a given permutation π:

$$E(\pi) = -\sum_{l} P(y|C_\pi) \log P(y|C_\pi). \quad (6)$$

This metric is advantageous as it is not only independent of additional data but also markedly reduces the computational cost required to assess permutations. Guided by Equation (6), we initially sample a large number of permutations and then filter out those with low content-free entropy to obtain a small candidate set Π.

4.2 Influence-Guided Assignment

After the filtering process, we adaptively select the optimal permutation from Π for each test instance. Specifically, we choose the permutation that can maximize the influence of test instance x_t:

$$\begin{aligned} \pi^* &= \arg\max_{\pi} \left(I(x_t)\right) \\ &= \arg\max_{\pi} \left(P(l|x_t, C_\pi) - P(l|C_\pi)\right). \end{aligned} \quad (7)$$

In an ideal scenario, $P(y = l|C_\pi)$ is a constant for each candidate order after the filtering stage, which is difficult to guarantee in practice. Under such circumstances, the goal is equivalent to maximizing the confidence of LLMs, connecting our method to Wu et al. (2023) that selects the orders by minimizing the description length of labels.

In the practical implementations, such a selection process inevitably introduces additional computational overhead. Hence, we set a small size of Π to make a trade-off between cost and effectiveness. The process can be accelerated via batch inference. Our subsequent analysis in Section 6.1 validates that even a small size of Π can stabilize the results.

5 Experiments

5.1 Setup

Dateset Following prior studies (Zhao et al., 2021; Wu et al., 2023; Lu et al., 2022), we carry out our experiments on nine textual classification tasks across four task scenarios, including Sentiment Classification: SST2 (Socher et al., 2013), CR (Hu and Liu, 2004) and MR (Pang and Lee, 2005); Subjectivity Classification; Subj (Pang and Lee, 2004); Natural Language Inference: SNLI (Bowman et al., 2015) and RTE (Dagan et al., 2005); Topic Classification: TREC (Hovy et al., 2001), AgNews (Zhang et al., 2015) and DBPedia (Lehmann et al., 2015).

LLMs We run our experiments on four sizes of **LLaMA2** (1.3B, 2.7B (Xia et al., 2023), 7B, and 13B (Touvron et al., 2023) parameters). Unless explicitly indicated, Sheared LLaMA2 1.3B is used for most analyses.

Baselines For a comprehensive evaluation, we compare our framework against a range of existing methods. We first consider a **Zero-shot** method without in-context examples and a **Random** method that randomly initiates permutations for each dataset. For dataset-free methods, we consider **MDL** (Wu et al., 2023) which picks 10 random permutations for each instance and selects the one with a minimum codelength to compress label information, and **Similarity** (Liu et al., 2022) that arranges the examples according to their similarity with test instance in the embedding space. For dataset-based methods, we consider Mutual Information (**MI**) (Sorensen et al., 2022), and Global Entropy **GLE** (Lu et al., 2022). We also compare with **Best-of-10**, a powerful baseline that directly selects the best permutation out of 10 randomly generated permutations based on the validation performance.

Evaluation Following previous work (Lu et al., 2022; Xu et al., 2023), we randomly sample a subset of the validation set containing 256 instances for each dataset to evaluate the accuracy. All experiments are conducted using 5 fixed sets of demonstration examples using different seeds and the mean accuracy is reported.

Implemention Details In our framework, we set the iteration num $N = 100$ and the candidate set size $K = 4$. For dataset-based methods, we randomly sample a subset containing 200 instances from the training set. We adopt standard 4-shot settings for all the datasets except 1-shot settings for DBPedia due to the limited context window size. For the **Random** baseline, we use an ensemble of 4 random permutations to make a stable estimation. The prompt templates are listed in Appendix A.

5.2 Main Result

The main experiment results are depicted in Table 1 and Figure 5. The full results are listed in Appendix B. From the experimental results, we have the following findings.

DEmO outperforms dataset-based methods on most tasks. As delineated in Table 1, our framework achieves an average enhancement of 9.6% over the random baseline, surpassing all the methods compared. Our framework is especially effective on sentiment classification tasks, where achieves 17% improvement on CR and 22% improvement on MR. It is worth mentioning that the GLE method achieves the highest accuracy on SST2, which aligns with the strong positive correlation observed in our preliminary experiments. Additionally, the Best-of-10 strategy demonstrates superior performance to other methods by a large margin on TREC, indicating the limitations of existing methods on imbalanced data.

Existing dataset-free methods fail in few-shot settings. Although the Similarity method and MDL method help to improve ICL performance in previous example selection frameworks, we observed their failure to generalize to real few-shot scenarios: The MDL method only achieves a little performance improvement, while the Similarity method even leads to a performance decline of 2% on average. One possible explanation is that example selection methods alter the distribution of training examples, limiting the generalizability of MDL and Similarity to the original distribution.

DEmO yields steady improvements across different model scales. From Figure 5a we observe a continual improvement in accuracy with the increasing scales from 1.3B to 13B. In addition, we find that as the model size gets larger, the performance gap between DEmO and other dataset-based methods is gradually narrowing. For example, on LLaMA2 13B, the GLE method achieves the best result. However, since the increasing of model scales causes higher search expenses, our framework still maintains advantages in terms of cost.

DEmO is consistently effective with an increasing number of examples. We vary the number of demonstration examples (from 2-shots to 8-shots) and compare the ICL performance on CR and SNLI. We find our framework brings consistent improvements over baselines. Noteworthy is that while the performance on the CR task demonstrates a persistent improvement, the results on the SNLI exhibit obvious variability.

Method	SST2	CR	MR	Subj	RTE	SNLI	AgNews	TREC	DBPedia	Avg.
Zero-shot	55.47	50.78	51.95	44.14	53.52	31.25	25.39	37.50	10.16	40.02
Random	69.14	69.86	65.21	50.21	54.02	37.11	70.76	63.16	75.51	61.66
Best-of-10	**85.16**	81.72	87.03	55.08	56.33	38.20	74.77	**67.97**	83.05	69.92
Similarity♠	71.33	62.27	65.47	47.73	54.53	35.39	75.08	61.72	64.38	59.77
MDL♠	67.73	72.89	63.75	47.27	56.95	36.02	74.38	64.77	82.42	62.91
MI ♦	74.45	82.03	81.48	51.80	55.08	36.48	72.42	63.28	**83.12**	66.68
GLE ♦	**85.16**	80.86	87.34	55.08	57.11	39.45	74.53	65.78	80.39	69.52
DEmO (ours)♠	82.81	**86.41**	**87.73**	**61.72**	**58.05**	**40.23**	**76.48**	64.84	82.89	**71.24**

Table 1: Results on Sheared LLaMA2 1.3B. We present the best results in **bold** and the second with underlines. ♠ and ♦ represent the dataset-free methods and dataset-based methods respectively.

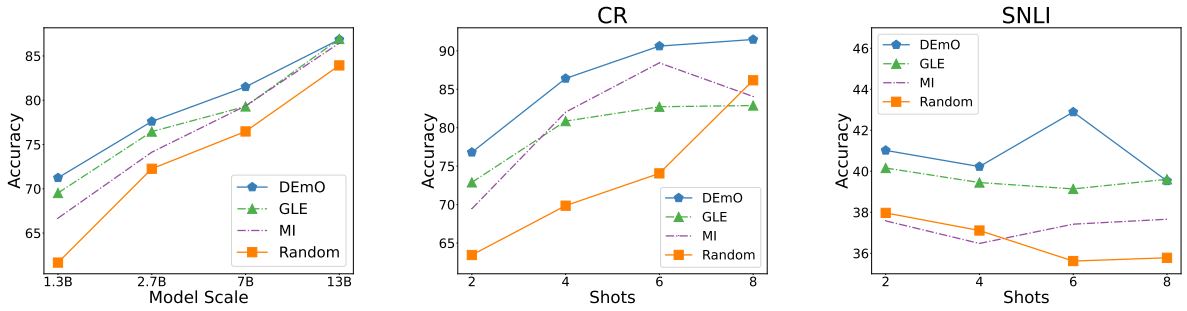

Figure 5: (a) impact of model scales across all datasets. (b,c) impact of number of examples on CR and SNLI.

We attribute the fluctuations to the inadequate reasoning capabilities of LLMs.

6 Discussions and Analysis

6.1 Generalisation Across LLMs

To further verify the generalizability our framework to LLMs beyond LLaMA2, we consider a spectrum of representative LLMs including GPT2-Large (0.8B) (Radford et al., 2019), GPT NEO (1.3B) (Black et al., 2021), BLOOM (Workshop et al., 2022) (1.7B), and OPT (1.3B) (Zhang et al., 2022). Since the context window size of GPT2 and OPT is limited to 1024 tokens, we choose 2-shot settings for AgNews and SNLI, 1-shot for DBPedia, and 4-shot for other tasks. As depicted in Figure 6, we find that **DEmO achieves substantial improvements over the random baseline, showing robustness across various LLMs.**

6.2 Generalisation to Closed-sourced LLMs

In the main experiments, we validate the effectiveness of DEmO on several open-source models. However, since DEmO requires access to the output probabilities of the model, it cannot be directly applied to powerful closed-source models (such as ChatGPT). Instead, we attempt to examine the scalability of DEmO by testing the performance of orders selected by small models on larger models. Specifically, we evaluate the orders selected by gpt2-large 0.8b and test on Chatgpt since they share the vocabulary. Table2 shows a slight increment in accuracy. We note a 7.81% increase in accuracy on the SUBJ dataset, which may be attributed to the exclusion of extremely unreasonable orders. For example, in the SUBJ dataset, subjective samples clustering at the end of the prompt may bias the prediction towards 'subjective'.

6.3 Impact of Hyperparameters

In this section, we focus on two hyperparameters in our framework: the number of iterations N and the size of the candidate set K. We conduct ablation studies to explore how these parameters influence overall performance.

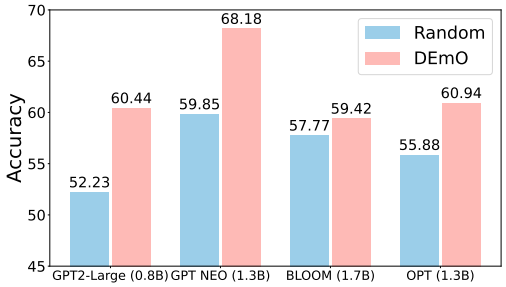

Figure 6: Average accuracy across different LLMs.

Dataset	Random	DEmO (ours)	Δ
SST2	95.31	95.31	0
CR	87.50	88.28	0.78
SNLI	71.88	71.88	0
SUBJ	68.75	76.56	7.81
AVG.	80.86	83.01	2.15

Table 2: 4-shot accuracy on four tasks where the orders are selected by gpt2-large (0.8B) and tested on Chatgpt (175B).

Increasing the iteration number helps to improve the ICL performance in the early stages. We vary the number of iterations from 10 (the same initialized permutations with dataset-based methods) to 1000. The accuracy is reported in Figure 7a. We observe a consistent enhancement in accuracy when N raises from 10 to 50 and a stable performance after approximately 100 iterations, which indicates the widespread existence of performant orders. Notably, DEmO shows a similar performance (69.61%) with GLE (69.52%) and Best-of-10 (69.92%) methods when $N = 10$. This indicates that DEmO can benefit from the increased number of searches using content-free estimation.

Increasing the candidate set size does not consistently lead to better ICL performance. As illustrated in Figure 7b, DEmO is able to achieve a stable improvement with a relatively small size of the candidate set. Because the usage of tokens increases linearly with K, increasing K will incur significant computational burdens. We set $K = 4$ to make a balance between performance and cost.

6.4 When a Validation Set is Available

Since our framework is designed for real-world few-shot settings and takes no label information into consideration, in this section we explore the potential of DEmO to be enhanced with a validation-guided search. Specifically, we additionally add a validation stage to incorporate label information into our framework. After the filtering stage, we first pick 10 permutations with the highest content-free entropy and rank them by their performance on a validation set containing 200 instances. Then we choose the best performant four permutations to construct the candidate set for the subsequent inference stage.

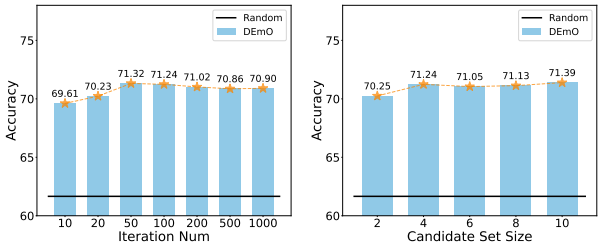

Figure 7: (a) impact of model scales across all datasets. (b) impact of number of examples on CR and SNLI.

Table 3 shows that DEmO further achieves a 2.1% improvement in accuracy, exceeding the Best-of-10 by 3.4% on Sheared LLaMA2 1.3B. These results demonstrate **the applicability of our method in full-data scenarios**.

Method	**1.3B**	**2.7B**
Random	61.66	72.26
Best-of-10	69.92	76.42
DEmO	71.24	77.61
DEmO+Validation	**73.36**	**78.61**

Table 3: Average accuracy of Sheared LLaMA2 1.3B and 2.7B in full-data scenarios.

6.5 Compared with Contextual Calibration

A common calibration technique is widely applied in few-shot settings to enhance accuracy. Zhao et al. (2021) first propose a content-free input to measure the entire test-time distribution and use it to calibrate the model outputs. However, several studies show that such a post-calibration method may fail when the prompt is of poor quality.

We compare our framework with such contextual calibration method and observe that context calibration achieves 7.6% improvement on average

Dataset	Random	CC	DEmO
SST2	69.14	**84.26**	82.81
CR	69.86	**88.69**	86.41
MR	65.21	85.49	**87.73**
SUBJ	50.21	59.45	**61.72**
SNLI	37.11	39.23	**40.23**
RTE	57.84	53.79	**58.05**
AGNews	70.76	73.09	**76.48**
TREC	63.16	60.35	**64.84**
DBPedia	75.51	78.93	**82.89**
Avg.	61.66	69.25	**71.24**

Table 4: Accuracy on Sheared LLaMA2 1.3B. **Bold** indicates the best result and underline indicates the result worse than the random baseline.

over the random baseline (Table 4). We also find that contextual calibration performs well in binary classification tasks like SST2 but produces limited improvement in multiclass tasks. On RTE and TREC, contextual calibration even harms the accuracy, showing instability. Compared with it, DEmO is consistently effective across different tasks and demonstrates a higher average accuracy.

6.6 Extension to Generation Tasks

The metrics of content-free entropy and influence are limited to tasks with discrete labels. However, for open-ended generation, the output space is the whole vocabulary. A straight way of extension to generation tasks is to directly operate the first output token probabilities on the output space rather than on specific label words. Following Zhao et al. (2021), we choose MIT Movies, a dataset of Information Extraction tasks, and adopt the same 6-shot setups. In Table 5, **we observe an improvement in Exact Match, showing the promise of operating the first output token.**

Method	Exact Match	Std.
Random	82.50	2.30
DEmO	**83.91**	1.75

Table 5: Average exact match along with the standard deviation on MIT Movies.

7 Conclulsion

In this paper, we introduce DEmO, a novel and general framework for in-context learning example ordering. DEmO identifies candidate orders with label fairness at the corpus level and adaptively selects the one enabling high influence of test instance. Different from existing methods that utilize in-domain data to find performant orders, DEmO exhibits no data dependency and can achieve impressive performance under realistic scenarios. Extensive experiments and analysis demonstrate its effectiveness and robustness across various settings.

Limitations

Despite that DEmO achieves impressive results, it still faces the following limitations: (1) Our framework is limited by the lack of label information, particularly on imbalanced datasets (2) The metrics of content-free entropy and influence are designed for classification tasks. More reasonable metrics rather than naive operation on the the first token are supposed to be explored.

References

Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, AS Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H Al-Timemy, et al. 2023. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. Journal of Big Data, 10(1):46.

Sid Black, Gao Leo, 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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In Proceedings of the 61st Annual Meeting of the

Association for Computational Linguistics (Volume 1: Long Papers), pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In Machine learning challenges workshop, pages 177–190. Springer.

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

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In Proceedings of the First International Conference on Human Language Technology Research.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2):167–195.

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. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2023. Fairness-guided few-shot prompting for large language models. arXiv preprint arXiv:2303.13217.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. arXiv preprint arXiv:2302.11042.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. Advances in neural information processing systems, 34:11054–11070.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

R.J. Solomonoff. 1964. A formal theory of inductive inference. part i. Information and Control, 7(1):1–22.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-

bog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. arXiv preprint arXiv:2301.11916, page 3.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.

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

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. arXiv preprint arXiv:2310.06694.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. 2023. k nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. arXiv preprint arXiv:2303.13824.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5443–5456, Singapore. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

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

Fei Zhao, Taotian Pang, Zhen Wu, Zheng Ma, Shujian Huang, and Xinyu Dai. 2023a. Dynamic demonstrations controller for in-context learning.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023b. A survey of large language models. arXiv preprint arXiv:2303.18223.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In International Conference on Machine Learning, pages 12697–12706. PMLR.

A Prompt Template

Table 8 shows prompt templates used for different datasets in this paper.

B Full Results on LLaMA2

The full results under 4-shot settings are listed in Table 9.

C Complexity Analysis

Time complexities of different baselines are shown in Table 6. In addition, we further quantify the cost by comparing the token usage of different methods on SST2 (the same setup in main experiments) in table 7. As seen, our method is superior or on par to previous methods.

Method	Complexity	Remark
ICL	$O(N)$	N: number of test instances
Similarity	$O(KN)$	K: number of demonstration examples
MDL	$O(LN)$	L: number of random orders
GLE	$O(LM+N)$	M: size of validation set
MI	$O(LM+N)$	M: size of validation set
DEmO	$O(L+KN)$	L: number of random orders K: size of candidate set

Table 6: Time complexities analysis of different methods.

	Dataset-based methods (GLE, MI)	MDL	DEmO(ours)
Before Inference	$1,029,740$	0	153,712
During Inference	$278,312$	$1,391,560$	556,624
Total	$1,308,052$	$1,391,560$	710,336

Table 7: Token usage of different methods on SST2.

Task	**Prompt**	**Label Names**
SST2	Review:[x]\nSentiment:[y]	negative, positive
CR	Review:[x]\nSentiment:[y]	negative, positive
MR	Review:[x]\nSentiment:[y]	negative, positive
Subj	Input:[x]\nLabel:[y]	objective, subjective
RTE	Premise: [pre]\n Hypothesis: [hyp]\nAnswer:[y]	yes, no
SNLI	Premise: [pre]\n Hypothesis: [hyp]\nAnswer:[y]	yes, maybe, no
AgNews	Article:[x]\nAnswer:[y]	World, Sports, Business, Technology
TREC	Question:[x]\nAnswer:[y]	Number, Location, Person, Description, Entity, Abbreviation
DBPedia	Article:[x]\nAnswer:[y]	Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book

Table 8: The template adopted for text classification. The right column shows the label names. We adopt the greedy-decoding method to check the probability for these tokens

Model	Method	SST2	CR	MR	Subj	RTE	SNLI	AgNews	TREC	DBPedia	Avg.
	Random	93.11	89.26	87.99	53.26	65.74	52.29	78.54	60.37	73.95	72.72
	Similarity	94.14	90.55	88.05	54.61	68.91	55.16	79.22	60.94	77.50	74.34
	MDL	94.14	90.55	88.05	54.61	68.91	55.16	79.22	60.94	77.50	74.34
2.7B	MI	93.59	91.95	90.78	53.44	66.02	53.36	80.70	58.67	78.67	74.13
	GLE	94.30	91.41	91.41	64.84	69.45	54.77	80.70	59.38	84.06	76.70
	Best-of-10	94.30	91.02	91.64	64.16	68.75	54.84	80.81	61.09	81.19	76.42
	DEmO	94.69	91.8	92.27	66.88	68.36	52.42	81.48	63.67	86.80	77.61
	Random	94.98	92.46	93.24	59.73	70.66	62.81	83.32	76.21	75.20	78.61
	Similarity	95.39	90.78	94.69	61.17	71.09	61.64	84.14	77.03	73.53	78.96
	MDL	95.47	92.58	95.00	59.14	72.43	65.86	83.11	76.95	77.28	79.76
7B	MI	96.09	92.27	95.47	62.66	73.59	65.16	83.91	77.03	72.50	79.85
	GLE	94.92	92.50	95.47	63.42	72.03	65.19	83.91	76.56	75.73	79.96
	Best-of-10	95.23	92.27	95.39	69.14	74.77	66.88	84.53	78.67	87.28	82.68
	DEmO	92.81	92.89	95.55	67.50	73.91	69.30	83.67	77.42	90.55	81.51
	Random	95.06	90.76	95.41	74.12	80.9	77.73	83.22	81.37	88.69	85.25
	Similarity	95.00	90.55	95.08	78.98	80.23	75.55	83.12	79.45	86.88	84.98
	MDL	96.09	91.17	95.70	76.17	80.73	79.38	83.77	81.33	90.62	86.10
13B	MI	95.55	91.95	95.16	81.80	80.16	78.05	83.91	80.00	91.56	86.46
	GLE	95.23	91.02	94.61	81.95	83.44	78.75	84.92	80.86	91.48	86.91
	Best-of-10	95.08	91.41	96.09	84.30	82.27	80.23	84.77	82.97	92.27	87.71
	DEmO	94.69	91.56	94.53	81.66	81.72	78.12	83.12	81.64	91.64	86.52

Table 9: 4-shot full results.