SeCoKD: Aligning Large Language Models for In-Context Learning with Fewer Shots

Anonymous ACL submission

Abstract

Previous studies have shown that demonstrations can significantly help Large Language Models (LLMs) perform better on the given tasks. However, this so-called In-Context Learning (ICL) ability is very sensitive to the presenting context, and often dozens of demonstrations are needed. In this work, we investigate if we can reduce the shot number while still maintaining a competitive performance. We present *SeCoKD*, a self-Knowledge Distillation (KD) training framework that aligns the student model with a heavily prompted variation, thereby increasing the utilization of a single demonstration. We experiment with the *SeCoKD* across three LLMs and six benchmarks focusing mainly on reasoning tasks. Results show that our method outperforms the base model and Supervised Fine-tuning (SFT), especially in zero-shot and one-shot settings by 30% and 10%, respectively. Moreover, SeCoKD brings little negative artifacts when evaluated on new tasks, which is more robust than Supervised Fine-tuning.

1 Introduction

When scaling up Large Language Model (LLM)s, the ability of ICL emerges (Brown et al., 2020; Agarwal et al., 2024; Dong et al., 2022). Models can learn from a few demonstrations and thus can be generalized to various downstream tasks without updating the parameters (Wei et al., 2023). However, the mechanism behind the few-shot learning ability remains unclear. Large language models are very sensitive to the quality of demonstrations, such as the number of demonstrations (Pan, 2023; Chen et al., 2023), the order of reasoning steps (Lu et al., 2021; Zhao et al., 2021), and the correctness of labels (Halawi et al., 2023). Moreover, the design of a demonstration also plays an important role (Zhao et al., 2021; Wang et al., 2022; Fu et al., 2022; Wei et al., 2022). As a result, it is not trivial to design a proper demonstration and the importance of prompt engineering continues to increase (Reynolds and McDonell, 2021; Dong et al., 2022; Zhou et al., 2022). Currently, it is common to use dozens of demonstrations together to overcome the possible weakness of a single prompt. However, we argue that humans often do not need more than two examples in the context of Q&A. One demonstration can serve as a guideline and show the correct format for answering the question, but more similar demonstrations are irrelevant to the correctness of the answer. In other words, humans do one-shot or zero-shot learning and they are not few-shot learners.

In this paper, we propose a simple yet effective KD method called *SeCoKD*, which stands for **Se**lf **Co**ntext **K**nowledge **D**istillation. Our method significantly reduces the number of demonstrations needed in the context by increasing the utilization of a single demonstration. The intuition is that since an LLM can answer a question correctly when triggered by a certain amount of external information (few-shot learning), we could use less information (one-shot learning) by aligning the model space and the task space through self-KD. It differs from internalizing knowledge; instead, it promotes the model to utilize existing information to activate its internal knowledge, a process previously achieved by providing a handful of examples.

First, we show that SeCoKD strongly improves the model performance on zero-shot and one-shot learning. We also consider the model trained with supervised fine-tuning as a strong baseline. In comparison, our method achieves better performance, especially when the original training set doesn't provide reasoning steps. For example, when performing one-shot ICL on the ARC-C (Clark et al., 2018a) dataset, the Mistral-7B fine-tuned with our method scores 60% accuracy, 10% higher than the initial model and 3% higher than the SFT version.

Second, we demonstrate that SeCoKD not only enhances performance on the training task but also

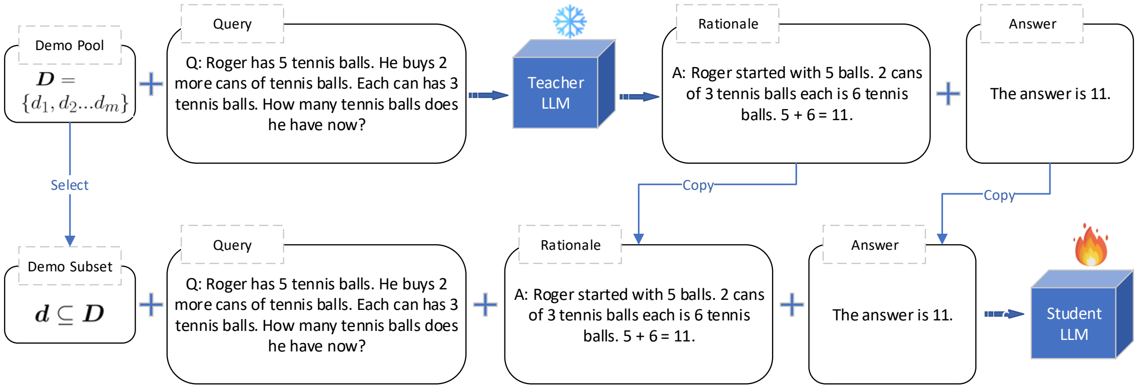
Demo Pool
$D = \{d_1, d_2 ... d_m\}$
Query
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
Teacher LLM
Rationale
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11.
Answer
The answer is 11.
Select
Copy
Copy
Demo Subset
$d \subseteq D$
Query
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
Rationale
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11.
Answer
The answer is 11.
Student LLM

Figure 1: Overview of the SeCoKD framework. The teacher model first generates high-quality rationale and answers for a query through 8-shot ICL. Then a student is trained using fewer demonstrations and the teacher's output.

maintains robustness across different tasks, unlike SFT, which can reduce accuracy on unseen tasks. This indicates that our method is more robust compared to SFT.

Third, empirical experiments suggest that SeCoKD simplifies tasks by converting difficult queries into easier ones when the same demonstration is provided. In contrast, while SFT occasionally outperforms SeCoKD in accuracy, its improvements are inconsistent: some queries that are initially easy for the base model become significantly more challenging after SFT.

In summary, the contributions of this study are as follows:

- To the best of our knowledge, this work represents the first approach deliberately designed to reduce the number of demonstrations used for ICL by enhancing the model's ability to utilize a single demonstration.
- We design a KD training pipeline called SeCoKD and conduct comprehensive empirical experiments on various reasoning tasks in the ICL setting. In total, 6 datasets and 3 different models are used in this study.
- We investigate the robustness of SeCoKD in comparison to the SFT and show that our method not only provides more consistent improvements but also generalizes well to unseen tasks.

2 Related Work

2.1 Few-Shot In-Context Learning

Recent work (Radford et al., 2019) demonstrated that large Pretrained Language Models can perform incredibly well on standard NLP tasks without being fine-tuned on task-specific datasets. Furthermore, Brown et al. (2020) suggested that the performance can be improved by feeding extra information in the input context. It is typically done by providing *demonstrations* of the same task. A *demonstration* refers to a text sequence that contains at minimum a query and its corresponding answer, concatenated by a predefined pattern. Additional information such as instructions and rationale can also be included. Although being competitive in certain tasks, ICL suffers from instability. Its performance depends heavily on the model size (Wei et al., 2023), the overall format of sequences (Min et al., 2022), number of demonstrations (Chen et al., 2023; Halawi et al., 2023), etc. As a result, there are no gold standards for designing context and the studies about ICL are mostly empirical.

On one hand, some works showed that enriching context can be beneficial. Agarwal et al. (2024) proposed many-shot learning to make full use of the allowed context length. With hundreds or thousands of demonstrations, models constantly perform better than just using a few demonstrations.

On the other hand, Chen et al. (2023) revealed that more demonstrations do not always bring benefits. Instead, ICL with one proper demonstration may perform better than few-shot learning using multiple random demonstrations. Towards more efficient ICL, existing works focus on demonstration selection (Li and Qiu, 2023; Wu et al., 2022; Li et al., 2023b) or context compression (Wingate et al., 2022; Ge et al., 2023). Zhang et al. (2022) proposed a reinforcement learning approach to select a handful of demonstrations from up to 1000 examples. Pan et al. (2024) developed a task-agnostic prompt compression technique that achieved a compression ratio of up to 5x without losing much performance. However, there is no existing approach to improve the model's internal ability to handle arbitrary demonstration, which can lead to a more fundamental solution. To fill this research void, we focus on reducing the number of demonstrations as much as possible while maintaining performance and robustness.

2.2 Distillation of Language Models

Knowledge Distillation (Hinton et al., 2015; Gou et al., 2021) is a technique in machine learning that involves transferring knowledge from a larger, more complex model (often referred to as the "teacher" model) to a smaller, more efficient model (known as the "student" model). The goal is to enable the student model to achieve performance similar to or close to that of the teacher model but with reduced computational cost and lower resource requirements. Xu et al. (2024) recently summarized three main motivations for applying KD in context of LLM: *a*) trying to let the open-source models mimic and learn from the more powerful closed–source model, *b*) offering compressed and efficient models, *c*) enhancing models using self-generated data through self KD.

The last point is an emerging research topic since the recent LLMs can generate high-quality data that can be used for self-improvement. In Sun et al.'s (2024) work, the authors synthesized around 360k training samples with LLaMA-65b and later fine-tuned the same model with these data. Thanks to the self-alignment between the model and the generated data, their model surpassed many models trained with human-curated samples. Extending the idea of self-improvement, we propose to use the same model to generate a high-quality rationale for a query that serves as the most aligned supervision to train a student model.

3 SeCoKD Overview

The primary training objective of SeCoKD is to have the student model emulate the teacher model, which is activated by a handful of demonstrations. Concretely, let $\boldsymbol{D} = \{d_1, d_2, d_3...d_n\}$ denotes a set of demonstrations and $\boldsymbol{d} \subseteq \boldsymbol{D}$ denotes a subset. (x, y, θ) are the input query, true label, and model parameters, respectively. In the setting of few-shot learning we have $P_{\mathcal{M}} = (y \mid x, \boldsymbol{D}, \theta_{\mathcal{M}})$ for the model $\mathcal{M}$. After applying our training method, we showcase that the updated Model $\mathcal{M}'$ also performs well with a high $P_{\mathcal{M}'} = (y \mid x, \boldsymbol{d}, \theta_{\mathcal{M}'})$. Since we focus on a self-distillation manner and we fine-tune the model with LoRA, the expression can be rewritten as $P_{\mathcal{M}'} = (y \mid x, \boldsymbol{d}, \theta_{LoRA}, \theta_{\mathcal{M}})$. As depicted in Figure 1, the whole pipeline can be divided into two steps. First, the teacher model is prompted with a set of demonstrations and a query. Each demonstration contains a question, a rationale, and an answer. The reasons to include some reasoning steps are two-fold: *a*) It is shown that Chain-of-Thought (CoT) prompting increases the reasoning ability of LLMs and thus the performance will be better (Wei et al., 2022; Shao et al., 2023). *b*) We need more tokens generated from the teacher model as supervision of the student model. For each task, we use a carefully curated demonstration set as gold samples. We then extract the reasoning part and the answer from the teacher model's output and save them for later use.

In the second step, we randomly sample a subset of the available demonstrations, concatenate it with the same query as in the first step, and use this sequence as input for the student model. Then we apply Sequential-Level KD (Kim and Rush, 2016) to fine-tune the student model.

To explain the whole pipeline mathematically, we first obtain the teacher output as

$$\mathbf{r} = g(f_{teacher}\,(\boldsymbol{D}, x)) \qquad (1)$$

where $f(\cdot)$ is the generation function and $\boldsymbol{D}$ is the demonstration pool. We use the extraction function $g(\cdot)$ to obtain the teacher-forcing supervision for the student model. Our training objective is to find parameters θ of the student model S that maximize the sequential-level log-probability sum:

$$\begin{aligned}\mathcal{M}_{Seq}(\theta) &= \mathbb{E}_{(\boldsymbol{pre},\boldsymbol{r})\sim\mathcal{D}}\left[\log S_\theta(\hat{\boldsymbol{r}} = \boldsymbol{r}; \boldsymbol{pre})\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{L_r} \log S_\theta\left(\hat{r}_i = r_i; \boldsymbol{pre}, \boldsymbol{r}_{<i}\right)\right]\end{aligned} \qquad (2)$$

where $\boldsymbol{pre}$ denotes the student input, containing the selected demonstrations and the query. Given this objective, the corresponding loss function to be minimized can be framed as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{L_r^{(j)}} \log S_\theta\left(\hat{r}_i^{(j)} = r_i^{(j)} \mid p^{(j)}, r_{<i}^{(j)}\right) \qquad (3)$$

Demonstration Selection We develop two strategies to select demonstrations for the student model. **SeCoKD-S** randomly samples one demonstration out of the demonstration pool. This represents the extreme case where we hypothesize that one example can already provide enough guidance for the model. **SeCoKD-M**, on the other hand, samples a different number of demonstrations from 1~4 for the student model, providing a stronger initial guidance.

4 Experimental Settings

We aim to evaluate the performance of SeCoKD compared to directly supervised fine-tuning and the base model. Inspired by Wei et al. (2022), we choose 6 popular benchmarks, covering topics of arithmetic reasoning, commonsense reasoning, and symbolic reasoning. We conducted experiments with some of the most advancing LLMs. However, we could only test the models with less than 10 billion parameters due to the computation limits.

4.1 LLMs

We evaluate our method on three GPT-like autoregressive transformer language models. We use the 4-bit quantized version to save computation resources (Dettmers et al., 2022). The **Llama 2-7B** (Touvron et al., 2023) is one of the most popular open-source LLMs. **Llama 3-8B** (AI@Meta, 2024) is the latest member in the Llama family and appears to be the SOTA in various benchmarks. We also use the **Mistral-7B** (Jiang et al., 2023) which leverages the sliding window attention (SWA) mechanism to handle variants sequence lengths effectively. We conducted all experiments on a single NVIDIA V100 (40G) GPU. For the training, we use the same LoRA[1] (Hu et al., 2021) configuration for all models, and the trainable parameters thus reduce to around 0.18% of the full size. All results reported are the average of three runs and training-related hyperparameters are listed in A.2.

4.2 Datasets

We evaluate all methods on 6 benchmarks listed in Table 1. For mathematical reasoning tasks, we selected three datasets. The **GSM8K** (Cobbe et al., 2021) contains 8.5K high-quality and diverse text-based grade school math problems. The **SVAMP** (Patel et al., 2021) applies different types of variations to the existing math problems and creates a more robust benchmark. In **AQUA-RAT** (Ling et al., 2017), the answers to the math problems are multiple choices. This introduces diversity into our experiments. For the commonsense reasoning tasks, we selected **ARC-C** (Clark et al., 2018b) which contains relatively difficult natural grade-school level questions, and the **CSQA** (Talmor et al., 2019) which utilized crowd-workers to create multiple-choice questions that cover a wide range of topics. We chose the **Coin-Flip** dataset introduced by Wei et al. (2022) for the symbolic reasoning task. In this task, the model is asked if a coin is still heads-up after n people flip it. For each task, we randomly sample 800 pieces of data for training and 200 for testing.

Dataset	Rationale	Multiple Choice
ARC-C	curated	✓
CSQA	✓	✓
SVAMP	same as GSM8K	
AQUA-RAT	✓	✓
GSM8K	✓	
COIN-FLIP	✓	

Table 1: We need to apply CoT prompting in our training pipeline. We reuse the existing demonstrations for GSM8K, COIN-FLIP, SVAMP and CSQA tasks from Wei et al. (2022). For ARC-C we manually curate 8 gold demonstrations. The full demonstrations are listed in A.3

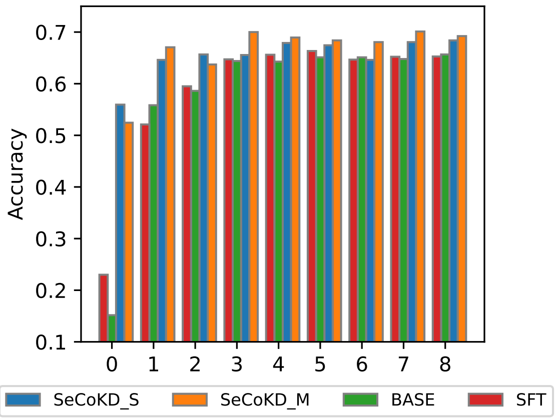

Figure 2: Comparison of 4 methods with different shot numbers. The X-axis represents the number of demonstrations. The Y-axis shows the average accuracy of all six tasks. SeCoKD significantly outperforms the other two baselines in zero-shot and one-shot scenarios.

[1] https://huggingface.co/docs/diffusers/en/training/lora

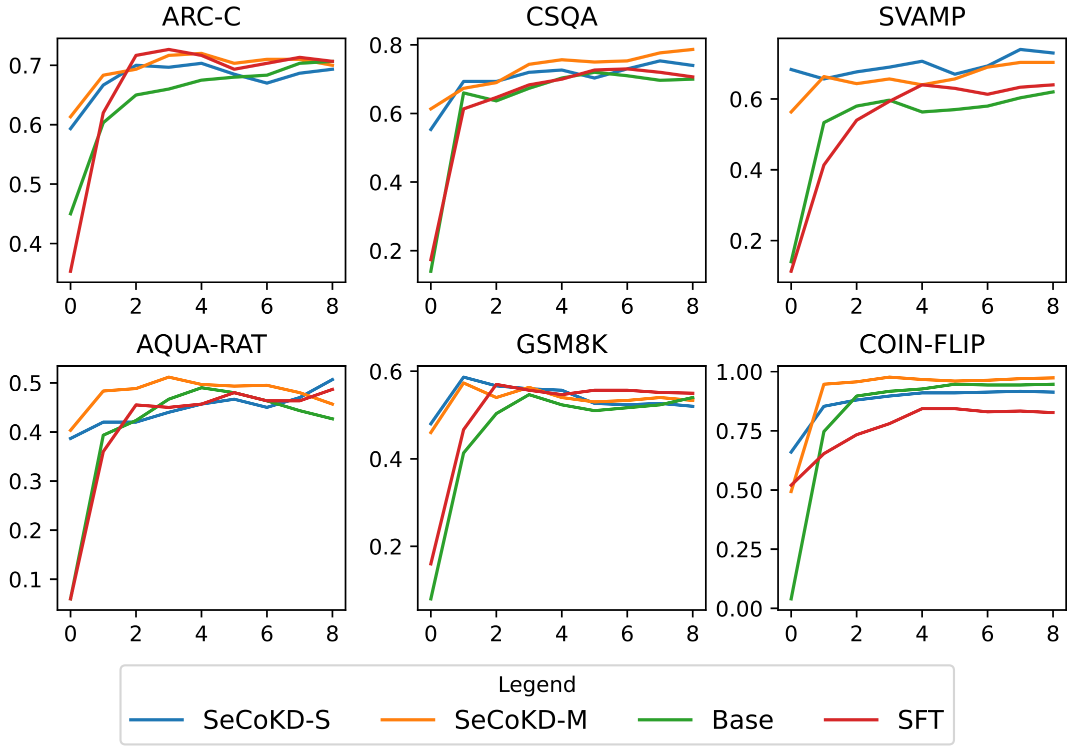

Figure 3: Few-Shot performance on each task. The X-axis represents the number of demonstrations. Our methods SeCoKD-S and SeCoKD-M perform much better in zero-shot and one-shot compared to the two baselines.

5 Results and Discussion

5.1 Results for Few-Shot ICL

Figure 2 shows an overall comparison between SeCoKD and baseline methods. The underlying model here is Llama 3-8B and more results regarding different LLM structures can be found in A.1. First, while all fine-tuned models perform better than the base model, the two variants SeCoKD-S and SeCoKD-M are better than SFT in most scenarios. We observe the largest margins in the zero-shot case, meaning that context information is successfully compressed. Second, we notice that the difference between the SeCoKD-S and SeCoKD-M is quite small. This means that in the distillation process, the model does not need a strong initial context to align with the guidance from the teacher. In the following experiments, we only use the SeCoKD-S to reduce computational resources. Last, starting from Four-Shot, adding more demonstrations seems to have no more positive impact on the performance of all methods. This observation is consistent with the study from Min et al. (2021). In their work, ICL brings only marginal improvements also after around 4 demonstrations. This indicates that there is a performance upper bond for the model that can be lifted by training, not by ICL.

In Figure 3 we look separately at performance comparison on each dataset. We could see that in all tasks, the base model struggles in the zero-shot case, delivering the poorest performance. However, when providing more demonstrations, the performance is significantly increased up to an upper bound. After that, more demonstrations seem to have limited help, sometimes even degrading the performance for example for the AQUA-RAT task. The models trained with SFT also perform generally not well in the zero-shot settings except for the COIN-FLIP task. It even degrades the model's performance on ARC-C. When providing more demonstrations, SFT can offer only limited improvement. Conversely, models trained with SeCoKD exhibit significantly better zero-shot performance across all tasks. Furthermore, the one-shot accuracy with SeCoKD already achieves optimal performance, indicating that more than one demonstration is unnecessary due to the effectiveness of the KD pipeline.

Table 2 presents a comparison of one-shot accuracy on six different tasks across three models: Llama 3-8B, Llama 2-7B, and Mistral-7B.

		ARC-C	CSQA	SVAMP	AQUA-RAT	GSM8K	COIN-FLIP
Llama 3 -8B	Base	0.6	0.66	0.53	0.39	0.41	0.74
	SFT	0.62	0.61	0.41	0.36	0.46	0.65
	SeCoKD-S	0.67	**0.69**	**0.66**	0.44	**0.58**	0.85
	SeCoKD-M	**0.68**	0.67	**0.66**	**0.48**	0.57	**0.94**
Llama 2 -7B	Base	0.4	0.42	0.29	0.14	0.05	0.51
	SFT	0.34	0.41	0.22	**0.18**	0.08	0.55
	SeCoKD-S	**0.48**	0.52	0.3	0.15	**0.19**	0.62
	SeCoKD-M	0.45	**0.53**	**0.32**	0.14	0.18	**0.63**
Mistral -7B	Base	0.5	0.68	0.53	0.25	0.28	0.61
	SFT	0.58	**0.7**	0.65	**0.32**	0.44	0.59
	SeCoKD-S	0.59	0.68	0.62	0.27	**0.6**	0.74
	SeCoKD-M	**0.60**	0.69	**0.65**	0.28	0.58	**0.78**

Table 2: Comparison of one-shot accuracy on different tasks and different models. Bold values represent the best results within a model structure. We could see that in most cases SeCoKD performs the best.

The methods compared are Base, SFT, SeCoKD-S, and SeCoKD-M. SeCoKD generally performs best across different tasks and models, showing the highest accuracy in most cases. For instance, in the ARC-C task, SeCoKD-M achieves 68% accuracy with the Llama 3 model, outperforming the Base method at 60% and SFT at 62%. Similarly, in the GSM8K task, SeCoKD-M reaches 60% accuracy with the Mistral model, while Base and SFT score 28% and 44%, respectively. SeCoKD-S often closely follows SeCoKD-M or performs slightly better, such as in the CSQA task with Llama 3, where SeCoKD-S scores 69% compared to SeCoKD-M's 67%. In contrast, Base and SFT methods typically show lower performance compared to SeCoKD methods, with SFT sometimes even performing worse than the Base model, especially in the Llama 2 model, where SFT scores 41% in the CSQA task compared to the Base's 42%. Overall, SeCoKD methods, significantly improve one-shot accuracy across various tasks compared to Base and SFT methods, especially in more complex models like Llama 3 and Mistral.

5.2 Robustness of SeCoKD

We demonstrate the superiority of SeCoKD over SFT by highlighting its robustness in cross-task testing scenarios. Our approach involves tuning a model on each individual task and then evaluating it not only on the test set of the same task but also on the test sets of other tasks. The rationale behind this experiment is twofold: 1. A model that is effectively trained on a specific task should exhibit the best performance on that task compared to other model variants. 2. The training objective aims to enhance the model's ability to utilize demonstrations. Therefore, ideally, training on one task should also positively impact the model's performance on other tasks.

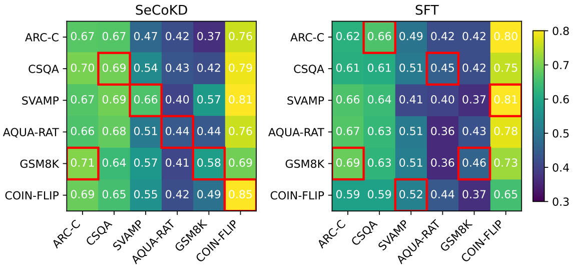

Figure 4: Cross-task tests of one-shot performance on different benchmarks. The Y-axis is the training task, and the X-axis represents the testing task. The cell value represents the absolute accuracy and we use the red boxes to highlight the best score in a column. For example, the top right cell shows the evaluation accuracy on the COIN-FLIP task when the model is trained on the ARC-C task.

Figure 4 shows the accuracy of the Llama 3 model on different tasks. When comparing within a column, it is evident that SeCoKD generally achieves the highest accuracy on the task used for training, with the exception of the commonsense reasoning task ARC-C. Here the model trained with the mathematical reasoning task GSM8K performs the best, 4% better than the model trained with ARC-C task. In this case, the model trained on the mathematical reasoning task GSM8K outperforms the one trained on ARC-C by 4%. However, training with SFT does not yield the best results

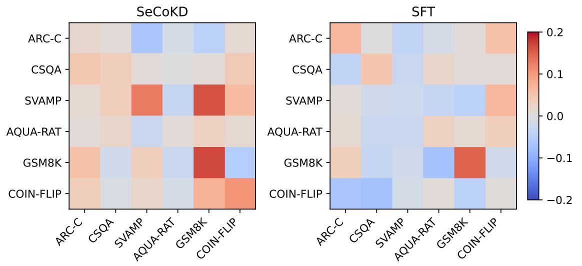

Figure 5: Cross-task evaluation of one-shot performance across different benchmarks. The Y-axis indicates the training task, while the X-axis represents the testing task. To assess the impact of the training method on model performance, we subtract the baseline accuracy from the accuracy achieved post-training. A red cell color indicates that the trained model outperforms the base model, whereas blue cells signify a decline in performance after training.

for the specific task in most cases. For instance, in the AQUA-RAT evaluation, the model trained on CSQA performs nearly 10% better than the one trained on AQUA-RAT. For the COIN-FLIP task, this performance gap can reach up to 15%.

We also utilize cross-task testing to showcase the high robustness of SeCoKD by visualizing the performance gap between post-training and baseline models in Figure 5. The color scale indicates the change in post-training compared to the baseline in terms of one-shot accuracy, with red indicating improvement and blue indicating a decline. We can see that SeCoKD has a more significant positive transfer effect, as evidenced by the broader spread of red cells across the heatmap, suggesting it generalizes better across tasks compared to SFT.

5.3 Simplifying tasks with SeCoKD

In this section, we emphasize the benefits of SeCoKD by showing that training with this method makes a task easier to solve. Inspired by Chen et al. (2023), we also provide a metric to make the measurement of easiness more tangible.

positive and negative demonstration Following the definitions in Chen's paper, a positive demonstration helps the model to answer correctly in the setting of one-shot learning. A negative demonstration, in contrast, results in a false answer.

Easy, Hard, and Hard$*$ sample For each task, there are in total eight existing or hand-crafted gold demonstrations. We conduct one-shot experiments using these demos and classify the sample into three categories based on the number of positive demonstrations n:

- easy: $n \geqslant 6$.
- hard: $6 > n > 1$.
- $hard*$: $1 \geq n$.

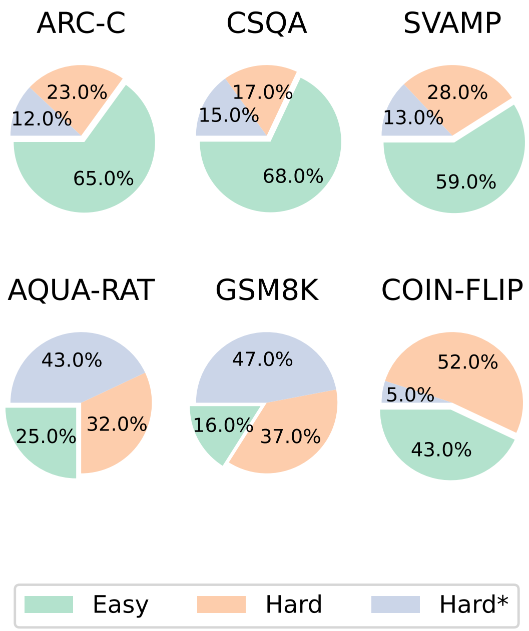

Figure 6: Queries in a dataset are categorized into three classes, representing their easiness to be solved with one-shot ICL. $Hard*$ means none of the existing demonstrations can lead to a correct answer. We can see that all datasets are very biased.

In the following experiments, we focus on the Mistral model, the conclusions drawn from the other two models are similar. Figure 6 visualizes the initial category distribution. We can see that the AQUA-RAT dataset stands out with a large portion of Hard* tasks (43%), indicating that it is predominantly challenging. Only a quarter of the dataset is categorized as Easy. GSM8K is also highly challenging with the smallest Easy category (16%) and a majority of samples falling under Hard (37%) and Hard* (47%), highlighting the dataset's complexity. As a result, we could observe very low one-shot accuracy for these two datasets in Table 2, both below 30%. The majority of the data in ARC-C, SVAMP, and CSQA is classified as Easy, suggesting that a significant portion of the samples can be easily addressed using demonstrations in one-shot learning. However, there is still a notable portion that ranges from hard to very hard, indicating a substantial amount of more challenging tasks.

	ARC-C	CSQA	SVAMP	AQUA-RAT	GSM8K	COIN-FLIP
SeCoKD	**1.22**	**0.96**	**2.18**	**1.35**	**3.56**	**4.57**
SFT	0.58	0.29	0.93	1.01	1.08	0.71

Table 3: Improvement Scores of SFT and SeCoKD. Larger is better.

Improvement Score To measure the change in data distribution with regard to the three categories, we develop a metric called *improvement score (IS)*:

$$IS = \exp\left(\frac{1}{N}\sum_{i=0}^{N}\frac{(n_i - m_i)}{D}\right) \tag{4}$$

where n and m represent the number of positive demonstrations obtained using the fine-tuned model and the base model, respectively. D is the size of the demonstration set which is 8 in our case. A higher IS value indicates that more demonstrations are considered positive for a given query, making the query an easier task. This metric is advantageous because it evaluates the transformation of individual samples into easier ones, rather than just comparing the overall data distribution. Essentially, IS measures the proportion of samples that become easier to handle, offering a more nuanced assessment of the training method.

From Table 3, it is evident that SeCoKD consistently outperforms SFT across all datasets. SeCoKD demonstrates particularly high scores in the COIN-FLIP and GSM8K datasets. While SFT occasionally achieves better accuracy scores, it often leads to a significant portion of previously easy tasks becoming more difficult, which is an undesirable outcome. For instance, as shown in Table 2, the Mistral model trained with SFT achieves an accuracy score of 0.32 on the AQUA-RAT dataset, whereas SeCoKD scores slightly lower at 0.28. However, SFT has an Improvement Score (IS) of 1.01, which is smaller than the IS achieved by the SeCoKD method. This indicates that despite the higher accuracy, SFT makes the tasks more challenging overall. SeCoKD, on the other hand, excels at preserving previously positive tasks while effectively converting difficult tasks.

6 Conclusion

We introduce SeCoKD, a Knowledge Distillation framework that enhances the In-Context Learning abilities of Large Language Models using fewer demonstrations. Our experiments show that SeCoKD significantly improves model performance, robustness, and efficiency compared to traditional methods like Supervised Fine-tuning.

SeCoKD-trained models excel with minimal demonstrations, achieving optimal accuracy with just one demonstration. They outperform base models by an average of 10% in one-shot ICL scenarios and show enhanced robustness without negative cross-task performance impacts, which is a common issue with SFT. Cross-task testing highlights SeCoKD's robustness and generalization, with models performing well not only on their training tasks but also on other tasks. This indicates effective compression and alignment of task-relevant knowledge. SeCoKD models also simplify complex tasks, demonstrating a higher capability to internalize and utilize fewer demonstrations. This benefit is quantified through metrics distinguishing positive and negative demonstrations and classifying task difficulty based on model responses. Overall, SeCoKD offers a promising solution for enhancing LLM performance in few-shot and zero-shot learning contexts, providing a more efficient and scalable approach for leveraging demonstrations in language model training.

7 Limitations

While SeCoKD shows significant promise, there are several limitations to consider. Firstly, the scope of our experiments is limited to models with fewer than 10 billion parameters due to computational constraints. This restriction may limit the generalizability of our findings to larger models, which are increasingly prevalent in current research and applications (Chung et al., 2022; Wei et al., 2023). Secondly, the benchmarks used in this study are focused primarily on reasoning tasks. While these benchmarks are diverse, extending the evaluation to include a broader range of tasks, such as language generation (Li et al., 2023a), summarization (He et al., 2023), or translation (Zhu et al., 2024), would provide a more comprehensive understanding of SeCoKD's effectiveness. Moreover, more cross-studies would help to assess the sustainability of SeCoKD's performance improvements over different types of tasks. Thirdly, in this study, we primarily focus on the self-KD settings, we save the opportunity to study distillation between different scales of models for future work. Finally, the computational overhead associated with training using SeCoKD, especially in resource-constrained environments, needs further investigation. Addressing these limitations in future research will be essential for fully realizing the potential of SeCoKD and extending its applicability to a wider range of LLMs and tasks.

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*.

AI@Meta. 2024. Llama 3 model card.

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.

Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023. How many demonstrations do you need for in-context learning? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11149–11159.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

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

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

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

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. 2023. Overthinking the truth: Understanding how language models process false demonstrations. *arXiv preprint arXiv:2307.09476*.

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19485–19494.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

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*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jia Li, Yunfei Zhao, Yongmin Li, Ge Li, and Zhi Jin. 2023a. Towards enhancing in-context learning for code generation. *arXiv preprint arXiv:2303.17780*.

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

Xiaonan Li and Xipeng Qiu. 2023. Finding supporting examples for in-context learning. *arXiv e-prints*, pages arXiv–2302.

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

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

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

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

Jane Pan. 2023. *What in-context learning "learns" in-context: Disentangling task recognition and task learning*. Ph.D. thesis, Princeton University.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *Preprint*, arXiv:2103.07191.

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.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. In *International Conference on Machine Learning*, pages 30706–30775. PMLR.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.

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

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. *arXiv preprint arXiv:2210.03162*.

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

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

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

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.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Shaolin Zhu, Menglong Cui, and Deyi Xiong. 2024. Towards robust in-context learning for machine translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16619–16629.

A Example Appendix

A.1 Further Results on Few-Shot Learning

Figure 7 and Figure 8 show the average performance comparison using Llama2 7B and Mistral 7B models. For the Llama2 model, we see a huge improvement when training with SeCoKD. However, our main conclusion stays unchanged: compared to SFT and the base model, SeCoKD has a much better performance in the zero-shot and one-shot settings.

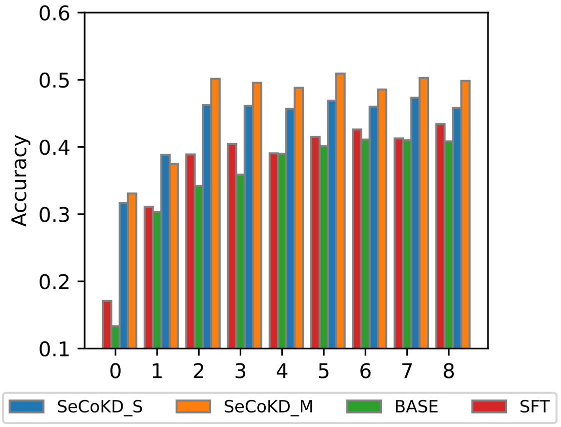

Figure 7: Comparison of 4 methods using Llama2 with different shot numbers. The X-axis represents the number of demonstrations. The Y-axis shows the average accuracy of all six tasks.

A.2 Hyperparameters

Table 4 summarizes the Lora configurations we used in our study. We used a relatively small rank (fewer trainable parameters) since we do not want to teach the model new associations beyond its knowledge. We target the 4 main linear layers of a transformer block but we did not tune this hyperparameter.

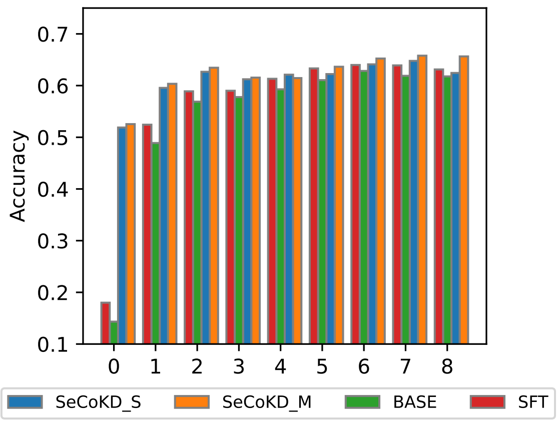

Figure 8: Comparison of 4 methods using Mistral with different shot numbers. The X-axis represents the number of demonstrations. The Y-axis shows the average accuracy of all six tasks.

For training, we used the paged_adamw_32bit optimizer as suggested in the QLoRA paper(Dettmers et al., 2023). The batch size for training and evaluation is two because of the computational limitation. The learning rate is set to 1e-4 with a warmup ratio of 0.02. The best checkpoint evaluated on the testing set is saved as the final result.

r (rank)	32
lora_alpha	64
target_modules	["q_proj", "k_proj", "out_proj","v_proj"]
lora_dropout	0.05
bias	"none"

Table 4: Lora configuration for all models.

A.3 Full Demonstrations

We reuse the existing demonstrations for GSM8K, COIN-FLIP, and CSQA tasks from Wei et al. (2022). For SVAMP, we use the same set of demonstrations as for GSM8K since they are both mathematical reasoning tasks with similar formats. For ARC-C we present the curated demonstrations in Table 5.

1. Question:George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat? (A) dry palms. (B) wet palms. (C) palms covered with oil. (D) palms covered with lotion. Answer: Dry surfaces will more likely cause more friction via rubbing than other smoother surfaces, hence dry palms will produce the most heat. The answer is: (A)

2. Question:Which factor will most likely cause a person to develop a fever? (A) a leg muscle relaxing after exercise. (B) a bacterial population in the bloodstream. (C) several viral particles on the skin. (D) carbohydrates being digested in the stomach. Answer: Option (B), bacterial population is the most likely cause for a person developing fever. The answer is: (B)

3. Question:Which change in the state of water particles causes the particles to become arranged in a fixed position? (A) boiling. (B) melting. (C) freezing. (D) evaporating. Answer: When water is freezed, the particles are arranged in a fixed position; the particles are still moving for all other options. The answer is: (C)

4. Question:When a switch is used in an electrical circuit, the switch can (A) cause the charge to build. (B) increase and decrease the voltage. (C) cause the current to change direction. (D) stop and start the flow of current. Answer: The function of a switch is to start and stop the flow of a current. The answer is: (D)

5. Question:Which of the following statements best explains why magnets usually stick to a refrigerator door? (A) The refrigerator door is smooth. (B) The refrigerator door contains iron. (C) The refrigerator door is a good conductor. (D) The refrigerator door has electric wires in it. Answer: Since iron is a ferromagnetic material that is strongly attracted to magnets The answer is: (B)

6. Question:Which of these do scientists offer as the most recent explanation as to why many plants and animals died out at the end of the Mesozoic era? (A) worldwide disease. (B) global mountain building. (C) rise of mammals that preyed upon plants and animals. (D) impact of an asteroid created dust that blocked the sunlight. Answer: The most accepted and supported explanation among scientists for the mass extinction event at the end of the Mesozoic era is (D) the impact of an asteroid that created dust blocking sunlight. This event led to drastic changes in climate and ecosystems, making it impossible for many species to survive. The answer is: (D)

7. Question:A boat is acted on by a river current flowing north and by wind blowing on its sails. The boat travels northeast. In which direction is the wind most likely applying force to the sails of the boat? (A) west. (B) east. (C) north. (D) south. Answer: The boat travels northeast, and the river current flows north. This implies that to achieve a northeast direction, the boat must receive an additional force component to the east. The answer is: (B)

8. Question:Which landform is the result of the constructive force of a glacier? (A) valleys carved by a moving glacier. (B) piles of rocks deposited by a melting glacier. (C) grooves created on a granite surface by a glacier. (D) bedrock hills roughened by the passing of a glacier. Answer: The constructive process results in the accumulation of debris and rocks, contributing to the formation of new landforms such as moraines, which are essentially piles of rocks and soil deposited by glaciers. The answer is: (B)

Table 5: Full prompts for the ARC-C dataset.