Data Whisperer: Efficient Data Selection for Task-Specific LLM Fine-Tuning via Few-Shot In-Context Learning

Anonymous ACL submission

Abstract

Fine-tuning large language models (LLMs) on task-specific data is essential for their effective deployment. As dataset sizes grow, efficiently selecting optimal subsets for training becomes crucial to balancing performance and computational costs. Traditional data selection methods often require fine-tuning a scoring model on the target dataset, which is time-consuming and resource-intensive, or rely on heuristics that fail to fully leverage the model's predictive capabilities. To address these challenges, we propose Data Whisperer, an efficient, training-free, attention-based method that leverages few-shot in-context learning with the model to be fine-tuned. Comprehensive evaluations were conducted on both raw and synthetic datasets across diverse tasks and models. Notably, Data Whisperer achieves superior performance compared to the full GSM8K dataset on the Llama-3-8B-Instruct model, using just 10% of the data, and outperforms existing methods with a 3.1-point improvement and a 7.4× speedup.

1 Introduction

Fine-tuning on task-specific data has become a standard approach for adapting large language models (LLMs) to specialized tasks (Ouyang et al., 2022). For instance, *continued pre-training* (Gururangan et al., 2020; Chang et al., 2024) involves extending the original pre-training phase on datasets closely aligned with the target domain; *instruction tuning* (Zhang et al., 2023; Üstün et al., 2024) focuses on leveraging instruction-response pairs to improve the model's ability to follow instructions; and *task-specific fine-tuning* enhances model performance on a particular task by using domain-specific data (Yang et al., 2024b; Zhang et al., 2024). As datasets continue to expand in size, however, a critical challenge arises: *how to efficiently select the optimal training examples—a process known as data selection—*to strike a balance between computational cost and model per-
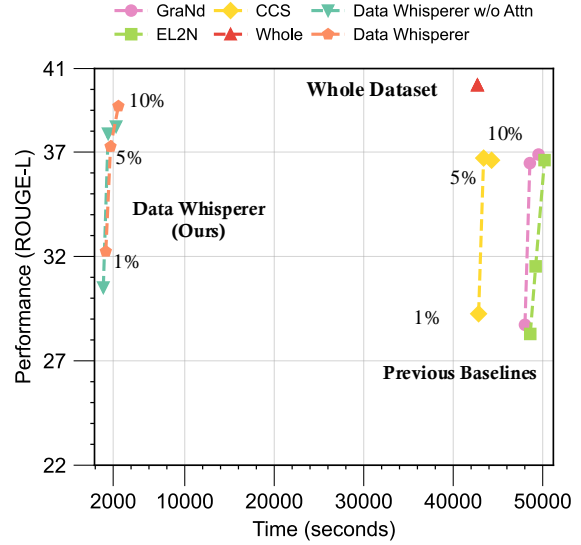

Figure 1: Comparison of the total time and performance across various data selection methods. For each method, we assessed Llama-3-8B-Instruct's performance and time cost when utilizing 1%, 5%, and 10% of the BioInstruct dataset selected by that method.

formance (Zheng et al., 2023; Paul et al., 2021; Li et al., 2023b; Zhang et al., 2024; Li et al., 2023a).

In this paper, we focus on *data selection for task-specific datasets*, aiming to extract the most informative subset from the original data to achieve nearly lossless performance compared to using the entire dataset for fine-tuning[1]. Although current data selection methods have demonstrated remarkable performance, they often suffer from significant inefficiencies. For instance, as shown in Figure 1, **previous state-of-the-art (SOTA) methods can take more time than directly using the entire BioInstruct dataset to fine-tune a Llama-3-8B-Instruct model, even when working with just 1% of the original dataset.** This inefficiency arises because these approaches depend on a *fine-tuned* model on the target dataset for scoring (Li et al.,

---

[1]We address scenarios where no more advanced LLMs are available beyond the model that is to be fine-tuned, *e.g.*, GPT-4 (OpenAI et al., 2023).

Table 1: Comparison of *Selection-to-Tuning Ratio* (STR) and performance across different selection methods on various datasets using 1%, 5%, and 10% of total data points. Lower STR values indicate higher efficiency. Evaluations were performed using the Llama-3-8B-Instruct model on eight NVIDIA A100 GPUs. ↑/↓ indicates improvement/degradation compared to random selection. "Speedup" represents the acceleration of Data Whisperer over the Nuggets (Li et al., 2023b) method. Results for Qwen and Mistral models are provided in Table 6.

| Method | GSM8K | | | | DialogSum | | | | BioInstruct | | | |
| | STR | | | Performance | STR | | | Performance | STR | | | Performance |
	1%	5%	10%	Avg. Score	1%	5%	10%	Avg. Score	1%	5%	10%	Avg. Score
Random	-	-	-	63.77	-	-	-	37.66	-	-	-	33.89
GraNd	1.04	1.05	1.08	64.98↑1.21	1.07	1.10	1.11	36.08↓1.58	1.02	1.03	1.05	34.03↑0.14
EL2N	1.05	1.07	1.12	64.44↑0.67	1.08	1.09	1.13	32.30↓5.36	1.02	1.03	1.06	32.14↓1.75
CCS	1.01	1.02	1.05	64.35↑0.58	1.01	1.03	1.05	30.65↓7.01	1.00	1.02	1.04	34.19↑0.30
Nuggets	1.22	1.23	1.26	66.07↑2.30	2.42	2.46	2.53	36.25↓1.41	0.46	0.47	0.49	32.51↓1.38
Data Whisperer	**0.13**	**0.14**	**0.17**	**68.14↑4.37**	**0.12**	**0.15**	**0.25**	**40.40↑2.74**	**0.03**	**0.04**	**0.06**	**36.46↑2.57**
Speedup	**9.38×**	**8.79×**	**7.41×**	-	**20.17×**	**16.40×**	**10.12×**	-	**15.33×**	**11.75×**	**8.17×**	-

2023a; Xia et al., 2024), typically requiring the same architecture as the model to be fine-tuned.

To this end, we first critically reevaluate the effectiveness of existing selection methods. To quantitatively and fairly assess the effectiveness of each method, we introduce the *Selection-to-Tuning Ratio* (STR), which is defined as the ratio of time spent on selection to the time required for fine-tuning the model on the entire dataset. Formally, let $t_p(\tau, \rho)$ represent the time associated with a selection method τ with a budget subset ratio ρ, and let t_{ft} denote the corresponding fine-tuning time for the entire dataset. The STR is given by:

$$\text{STR}(\tau) = \frac{t_p(\tau, \rho)}{t_{ft}}. \quad (1)$$

Intuitively, **for a data selection method τ to be considered practically efficient, its STR should ideally be less than 1**. This condition ensures that the time spent on selection aligns with the benefits obtained during fine-tuning. However, as shown in Table 1, existing methods often exhibit an STR greater than 1, which constitutes a significant bottleneck in the data selection process, thereby severely limiting the scalability of these methods for large datasets and models.

To address the inefficiencies of existing data selection methods, we introduce Data Whisperer, a training-free, attention-based, and context-aware approach. Traditional methods typically rely on an additional LLM that is fine-tuned on the target dataset to provide scoring for data selection. In contrast, our approach takes advantage of the model's inherent predictive capabilities, inspired by recent theories that equate in-context learning (ICL) with fine-tuning in LLMs (Dai et al., 2022; Mosbach et al., 2023). As shown in Figure 2, we score each training sample using few-shot ICL, where the sample itself serves as a demonstration example. The model's performance, measured by its ability to respond to associated queries based on these demonstration examples, yields a raw score for each sample. To improve efficiency, this scoring is performed with a weak-to-strong strategy, *i.e.*, using a less powerful pre-trained model from the same model family.

In typical ICL, *context-awareness* (*e.g.*, the order of examples within the context window) can influence the model's performance due to its reliance on sequence-based attention mechanisms (Guo et al., 2024; Bhope et al., 2025). To mitigate this inherent order sensitivity, we weight the demonstration scores according to their respective attention scores, as shown in Figure 2. This context-aware weighting refines the selection process by ensuring that the final score for each training sample not only reflects its raw performance but also accounts for its contextual significance within the task. The final score for each sample is computed as the average of these weighted scores, which provides a more accurate and nuanced evaluation of each data point's contribution to the model's learning process.

Our contributions can be summarized as follows:

1. We critically examine existing data selection approaches and introduce the *Selection-to-Tuning Ratio*, a novel metric that quantifies the efficiency of these methods. We observed that all prior data selection methods are more inefficient than fine-tuning the LLM with entire dataset.

2. We propose Data Whisperer, an effective, training-free and attention-based method. Unlike previous approaches, our method eliminates the need to fine-tune a separate scoring model on the target dataset, ensuring greater efficiency.

3. Data Whisperer integrates seamlessly with *weak-to-strong* few-shot ICL schemes, enabling effective performance even when a weaker model within the same model family is employed for ICL. This enhances both the scalability and effi-
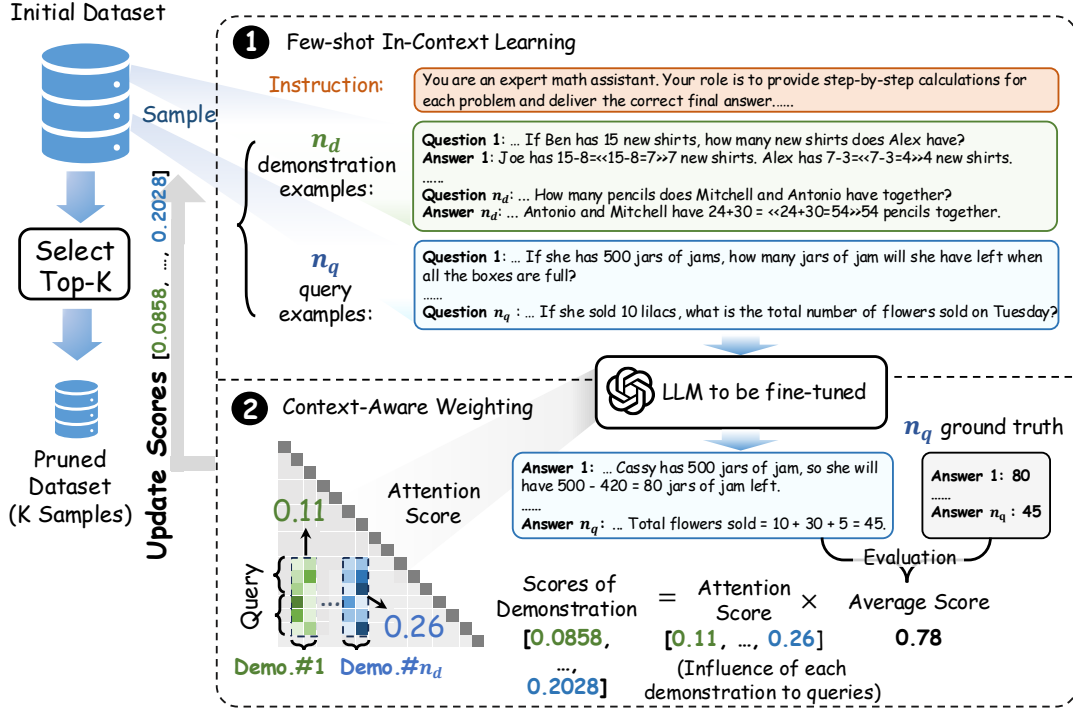
2

Figure 2: Illustration of the proposed Data Whisperer. n_d and n_q denote the number of demonstrations and queries in ICL. The pipeline consists of two main steps: (I) **Few-shot In-Context Learning**. A set of demonstration and query examples is randomly sampled from the initial dataset, and an ICL prompt is constructed with a fixed instruction. The LLM to be fine-tuned generates answers for all query examples, and the average evaluation score is computed using the ground truth answers. (II) **Context-Aware Weighting**. During each iteration of few-shot ICL, we weight the scores of the demonstration examples based on their attention scores, which quantify their influence on the queries. The scores are updated until all samples are scored. The final score for each sample is the average score across its appearances in multiple demonstration sets. Finally, we select the top-k samples from the dataset.

ciency of our method.

4. Comprehensive experiments are conducted on both real and synthetic datasets across various selection budget ratios, including BioInstruct (Tran et al., 2024), DialogSum (Chen et al., 2021), and GSM8K (Cobbe et al., 2021). We observe that Data Whisperer consistently outperforms previous SOTA methods, particularly in smaller data scenarios, while achieving faster selection times.

2 Related Work

2.1 In-Context Learning

In-Context Learning (ICL) is a powerful task adaptation technique that does not require modifying the weights of a pre-trained model (Brown et al., 2020; Olsson et al., 2022; Laskin et al., 2022). Instead, ICL adapts the model to new tasks by conditioning it on a sequence of demonstration pairs, where each demonstration consists of input-output pairs formatted according to a predefined template. This process guides the model in understanding the task. During inference, ICL involves presenting the model with a series of demonstrations followed by a query set, with the model expected to predict

the corresponding labels for the query data points based on the context.

Recent studies have explored the theoretical links between ICL and gradient descent, suggesting an implicit relationship between ICL and fine-tuning (Dai et al., 2022; Mosbach et al., 2023; Deutch et al., 2023; Zhou et al., 2024; Chen et al., 2024). One approach, Nuggets (Li et al., 2023b), uses one-shot learning for selection in *instruction-tuning* tasks with log probability scores, but is limited by the computational inefficiency of one-shot learning. In contrast, our work employs attention-based few-shot ICL on the pre-trained model, before any fine-tuning, to directly measure performance scores for *task-specific fine-tuning*. This approach leverages the model's inherent knowledge for data selection, offering a more efficient and scalable solution by linking attention-aware ICL predictions to the fine-tuning process.

2.2 Data Selection

Data selection is critical in deep learning, aiming to identify high-quality data that enhances training efficiency while maintaining comparable (or nearly

lossless) performance. Traditional methods rely on heuristic metrics, with two primary strategies: (I) *importance*-based selection, which targets challenging or essential samples (Raffel et al., 2020; Paul et al., 2021; Mirzasoleiman et al., 2020; Killamsetty et al., 2021; Tan et al., 2024), and (II) *similarity*- or *diversity*-based selection, which selects samples based on their similarity to others or their representativeness of the feature space (Bukharin and Zhao, 2023; Xia et al., 2024; Zheng et al., 2023; Wei et al., 2021; Yu et al., 2024).

For LLMs, these methods face two challenges: (I) the unreliability of heuristic metrics in high-dimensional spaces, and (II) the computational cost of fine-tuning additional scoring models. In contrast, our approach directly integrates data selection with the pre-trained model's predictions using ICL, eliminating the need for fine-tuning or heuristics.

3 Method: Data Whisperer

3.1 Preliminary: Data Selection in LLM Task-Specific Fine-Tuning

Given a large and diverse task dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, the objective of data selection is to identify a representative subset $\mathcal{D}' \subseteq \mathcal{D}$, with a budget ratio $\rho = |\mathcal{D}'|/|\mathcal{D}| < 1$, for fine-tuning a pre-trained language model $\mathcal{M}_p$. The fine-tuned model, $\mathcal{M}_f$, is then evaluated on the test set $\mathcal{D}_{\text{test}}$. The selected subset $\mathcal{D}'$ should ensure that the performance of the fine-tuned model $\mathcal{M}_f(\mathcal{D}')$ remains comparable to that of a model fine-tuned on the entire dataset $\mathcal{M}_f(\mathcal{D})$ when evaluated on $\mathcal{D}_{\text{test}}$. The data selection objective is formulated as:

$$\min_{\mathcal{D}' \subseteq \mathcal{D}, |\mathcal{D}'| = \rho \cdot |\mathcal{D}|} \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{test}}} \ell(x, y; \mathcal{M}_f(\mathcal{D}')), \quad (2)$$

where $\ell(\cdot)$ is the task-specific loss function (*e.g.*, cross-entropy). The key challenge in effective coreset selection is to efficiently identify the coreset $\mathcal{D}'$ for different selection ratios ρ.

3.2 ICL as a Data Selector for LLM Fine-Tuning

In contrast to conventional methods that rely on handcrafted heuristics of a fine-tuned model $\mathcal{M}_f$, our approach, Data Whisperer, utilizes the intrinsic ICL capabilities of the pre-trained model $\mathcal{M}_p$ for data selection. At each iteration, we randomly sample n_d demonstration examples $\mathcal{D}_d = \{(x_d^{(1)}, y_d^{(1)}), \ldots, (x_d^{(n_d)}, y_d^{(n_d)})\}$, and n_q query examples $\mathcal{D}_q = \{x_q^{(1)}, \ldots, x_q^{(n_q)}\}$, where $\mathcal{D}_d \cap \mathcal{D}_q =$

$\emptyset$. Combined with a fixed instruction I, the total context C is obtained, which is defined as:

$$C = \{I, (x_d^{(1)}, y_d^{(1)}), \ldots, (x_d^{(n_d)}, y_d^{(n_d)})\}, \quad (3)$$

The whole ICL process is formalized as:

$$\hat{y}_q^{(1)}, \ldots, \hat{y}_q^{(n_q)} = \mathcal{M}_p(C, x_q^{(1)}, \ldots, x_q^{(n_q)}). \quad (4)$$

Using performance metrics f (*e.g.*, average accuracy, or ROUGE-L (Lin, 2004), etc.), we compare the predicted outputs $\hat{y}_q^{(1)}, \ldots, \hat{y}_q^{(n_q)}$ with the ground truth labels $y_q^{(1)}, \ldots, y_q^{(n_q)}$ to compute the average performance score s:

$$s = \frac{1}{n_q} \sum_{j=1}^{n_q} f(\hat{y}_q^{(j)}, y_q^{(j)}). \quad (5)$$

This score is assigned to each sample in the demonstration set, and the process is repeated across multiple iterations. The final score for each sample is the average of its scores from all repetitions.

3.3 Context-Aware Weighting

To mitigate potential order sensitivity in few-shot ICL, we introduce a *context-aware* weighting mechanism for scoring the demonstration set, as shown in Figure 2. This mechanism leverages the self-attention scores from a specific layer l of the pre-trained model $\mathcal{M}_p$, across all attention heads h.

We focus on the attention scores corresponding to the first prediction token of a fixed layer l. Let $A^{(h)}$ denote the self-attention matrix of layer l. For each demonstration example $(x_d^{(i)}, y_d^{(i)})$, we extract the submatrix $A^{(h)}_{(x_d^{(i)}, y_d^{(i)})}$ from $A^{(h)}$, which contains the attention scores between this demonstration and all query examples $x_q^{(1)}, \ldots, x_q^{(n_q)}$. The sum of these attention scores quantifies the interaction between the demonstration and the query examples. The weight for each demonstration example, across all attention heads, is computed as:

$$w_{(x_d^{(i)}, y_d^{(i)})} = \sum_h \mathbf{1}^\top A^{(h)}_{(x_d^{(i)}, y_d^{(i)})} \mathbf{1}, \quad (6)$$

where $A^{(h)}_{(x_d^{(i)}, y_d^{(i)})}$ is the attention submatrix for head h, and the summation is performed across all heads. To account for variations in the length of demonstration examples, we normalize the summed attention scores by the length of each demonstration.

This process is repeated for each demonstration in every iteration until all samples are scored, yielding the scoring set $\mathcal{S}$. If any demonstration example

is scored more than once, the scores are averaged. Finally, we select the Top-k samples as follows:

$$\mathcal{D}' = \text{Top-k}(\mathcal{D}; \mathcal{S}), \qquad (7)$$

where $k = \lfloor \rho \cdot |\mathcal{D}| \rfloor$ denotes the coreset size. The detailed algorithm of Data Whisperer is demonstrated in Algorithm 1.

3.4 Theoretical Analysis

We now provide a theoretical analysis of Data Whisperer and demonstrate that our approach effectively performs implicit data selection for task-specific fine-tuning.

Analysis of ICL's Query Prediction. In ICL, the model adjusts its predictions based on the attention assigned to a set of demonstration examples. Let $x \in \mathbb{R}^{d_{\text{in}}}$ represent the input of a query token t, and $q = W_Q x$ denote the attention query vector. The attention result for a specific head is formulated as:

$$\mathcal{M}_p(q) = \text{Attn}(V, K, q)$$
$$= W_V[X_d; X_{q,<t}]\sigma\left(\frac{(W_K[X_d; X_{q,<t}]^\top q)}{\sqrt{d}}\right), \qquad (8)$$

where σ denotes the softmax operator, and $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ are the projection matrices for the attention queries, keys, and values, and $\sqrt{d}$ is the scaling factor. Here, $X_{q,<t}$ represents the input representations of the query tokens before token t, X_d represents the input representations of the demonstration tokens, and $[X_d; X_{q,<t}]$ represents the concatenation of these matrices. For qualitative analysis, we approximate the standard attention mechanism by relaxing it to linear attention, removing both the softmax operation and the scaling factor:

$$\mathcal{M}_p(q) \approx W_V[X_d; X_{q,<t}](W_K[X_d; X_{q,<t}])^\top q$$
$$= \underbrace{W_V X_{q,<t}(W_K X_{q,<t})^\top q}_{\text{zero shot input}} + \underbrace{W_V X_d(W_K X_d)^\top q}_{\text{ICL examples input}}$$
$$= W_{\text{zsl}}q + \sum_i (W_V x_d^{(i)})\left((W_K x_d^{(i)})^\top q\right)$$
$$= W_{\text{zsl}}q + \sum_i \left((W_V x_d^{(i)}) \otimes (W_K x_d^{(i)})\right) q$$
$$= (W_{\text{zsl}} + \Delta W_{\text{icl}})q \qquad (9)$$

In the above derivation, we observe that the attention to the demonstration tokens, X_d, can be viewed as an implicit parameter update ΔW_{icl}, which modifies the zero-shot weights W_{zsl} to adapt

to the task at hand. This update is essentially driven by the in-context learning examples and their relevance to the query tokens.

Analysis of Fine-Tuning for Query Prediction. In fine-tuning, model explicitly updates its key and value projections, W_K and W_V, through backpropagation to improve task performance. After fine-tuning, the model's attention can be expressed as:

$$\mathcal{M}_f(q) = (W_V + \Delta W_V)X_{q,<t}$$
$$\cdot X_{q,<t}^\top(W_K + \Delta W_K)^\top q$$
$$= \underbrace{W_V X_{q,<t}(W_K X_{q,<t})^\top q}_{\text{zero shot input}}$$
$$+ \Delta W_V X_{q,<t}(W_K X_{q,<t})^\top q$$
$$+ \Delta W_V X_{q,<t}(\Delta W_K X_{q,<t})^\top q$$
$$+ \Delta W_V X_{q,<t}(\Delta W_K X_{q,<t})^\top q$$
$$= (W_{\text{zsl}} + \Delta W_{\text{ft}})q. \qquad (10)$$

Here, ΔW_K and ΔW_V represent the parameter updates to the key and value projections W_K and W_V, respectively, which are learned via backpropagation from task-specific training objectives. The update ΔW_{ft} corresponds to the changes in W_{zsl} introduced by the fine-tuning process.

Connecting ICL Data Selection and Fine-Tuning. Based on the previous analysis, both ICL and FT follow a similar structure for task adaptation. Both methods modify the attention parameters W_K and W_V, but the difference lies in how the model updates them. In ICL, as shown in Eq. (9), the demonstration examples $x_d^{(i)}$ impact the query prediction by adjusting the attention weights. These weights, $(W_V x_d^{(i)}) \otimes (W_K x_d^{(i)})$, determine relevance of each example. In contrast, as shown in Eq. (10), FT explicitly refines W_K and W_V through gradients from the task-specific loss function.

Given the structural similarity between ICL and FT, we infer that using ICL for data selection is valid for FT. By selecting and weighting the most relevant examples in ICL, we achieve the same performance gains as fine-tuning, but without the need for explicit updates. Thus, Data Whisperer efficiently performs data selection through ICL, identifying the most relevant data points and adapting the model with minimal computational overhead.

4 Experiments

4.1 Experimental Setup

Datasets. We conducted experiments across four datasets, including three real-world and one syn-

Table 2: Evaluation results of different data selection methods on the GSM8k, DialogSum, and BioInstruct datasets. The model was fine-tuned on a coreset comprising 1%, 5%, and 10% of the original dataset size. ↑ indicates an improvement over random selection, while ↓ indicates a degradation compared to random selection.

Method	GSM8k			DialogSum			BioInstruct		
	1%	5%	10%	1%	5%	10%	1%	5%	10%
Llama-3-8B-Instruct (*Zero-shot*)		57.49			18.57			13.80	
+ Random	56.95	64.71	69.66	31.66	39.88	41.45	28.38	34.61	38.70
+ GraNd (Paul et al., 2021)	57.23↑0.28	67.65↑2.94	70.05↑0.39	26.02↓5.64	40.32↑0.44	41.90↑0.45	28.73↑0.35	36.47↑1.86	36.88↓1.82
+ EL2N (Paul et al., 2021)	60.16↑3.21	61.50↓3.21	71.66↑2.00	23.05↓8.61	35.14↓4.74	38.71↓2.74	28.29↓0.09	31.52↓3.09	36.62↓2.08
+ CCS (Zheng et al., 2023)	60.70↑3.75	63.64↓0.07	68.72↓0.94	23.48↓8.18	27.34↓12.54	41.12↓0.33	29.25↑0.87	36.71↑2.10	36.61↓2.09
+ Nuggets (Li et al., 2023b)	60.17↑3.22	68.65↑3.94	69.39↓0.27	31.77↑0.11	36.85↓3.03	40.13↓1.32	22.95↓5.43	36.41↑1.80	38.18↓0.52
+ Data Whisperer (ours)	**62.57**↑5.62	**69.65**↑4.94	**72.46**↑2.80	**38.05**↑6.39	**40.96**↑1.08	**42.18**↑0.73	**32.29**↑3.91	**37.27**↑2.66	**39.20**↑0.50
Whole Dataset		71.39			43.33			40.21	
Qwen-2.5-7B-Instruct (*Zero-shot*)		21.23			22.10			17.30	
+ Random	21.51	79.28	80.08	27.31	36.68	39.99	30.65	36.62	38.30
+ GraNd (Paul et al., 2021)	21.98↑0.47	82.09↑2.81	83.76↑3.68	30.61↑3.30	37.77↑1.09	40.50↑0.51	23.70↓6.95	35.89↓0.73	38.27↓0.03
+ EL2N (Paul et al., 2021)	22.13↑0.62	77.41↓1.87	82.62↑2.54	32.57↑5.26	38.90↑2.22	39.45↓0.54	30.36↓0.29	36.30↓0.32	37.26↓1.04
+ CCS (Zheng et al., 2023)	21.33↓0.18	80.75↑1.47	81.95↑1.87	31.69↑4.38	36.15↓0.53	38.05↓1.94	23.73↓6.92	34.53↓2.09	37.34↓0.96
+ Nuggets (Li et al., 2023b)	22.70↑1.19	82.09↑2.81	83.42↑3.34	32.11↑4.80	37.54↑0.86	40.23↑0.24	30.25↓0.40	36.33↓0.29	38.28↓0.02
+ STAFF (Zhang et al., 2024)	22.52↑1.01	82.22↑2.94	83.49↑3.41	30.16↑2.85	40.84↑4.16	40.54↑0.55	23.64↓7.01	37.08↑0.46	38.20↓0.10
+ Data Whisperer (ours)	**24.45**↑2.94	**83.16**↑3.88	**85.03**↑4.95	**32.95**↑5.64	**40.95**↑4.27	**43.00**↑3.01	**34.93**↑4.28	**37.57**↑0.95	**38.85**↑0.55
Whole Dataset		85.43			43.79			40.71	
Mistral-Nemo-Instruct (*Zero-shot*)		29.41			19.39			13.12	
+ Random	32.35	54.28	64.44	19.48	37.37	40.98	13.25	24.98	36.76
+ GraNd (Paul et al., 2021)	31.55↓0.80	56.02↑1.74	67.38↑2.94	19.86↑0.38	36.52↓0.85	41.79↑0.81	17.24↑3.99	25.76↑0.78	30.05↓6.71
+ EL2N (Paul et al., 2021)	30.43↓1.92	57.65↑3.37	67.14↑2.70	19.97↑0.49	35.88↓1.49	36.40↓4.58	14.93↑1.68	26.20↑1.22	30.15↓6.61
+ CCS (Zheng et al., 2023)	32.35↓0.00	53.61↓0.67	66.18↑1.74	18.55↓0.93	34.57↓2.80	42.14↑1.16	14.42↑1.17	28.12↑3.14	36.55↓0.21
+ Nuggets (Li et al., 2023b)	32.09↓0.26	58.29↑4.01	68.79↑4.35	19.52↑0.04	37.54↑0.17	40.52↓0.46	20.40↑7.15	27.40↑2.42	36.56↓0.20
+ STAFF (Zhang et al., 2024)	31.08↓1.27	63.18↑8.90	67.91↑3.47	19.49↑0.01	36.55↓0.82	42.08↑1.10	13.26↑0.01	25.40↑0.42	36.44↓0.32
+ Data Whisperer (ours)	**32.63**↑0.28	**65.91**↑11.63	**74.32**↑9.88	**21.52**↑2.04	**41.48**↑4.11	**43.36**↑2.38	**21.54**↑8.29	**29.98**↑5.00	**38.37**↑1.61
Whole Dataset		75.00			43.69			40.24	

Table 3: Evaluation results of different data selection methods on the synthetic dataset, generated from DialogSum dataset. The model was fine-tuned on a coreset comprising 5%, 10%, and 25% of the synthetic dataset size.

Method	Llama-3-8B-Instruct				Qwen-2.5-7B-Instruct				Mistral-Nemo-Instruct-2407			
	5%	10%	25%	100%	5%	10%	25%	100%	5%	10%	25%	100%
Random	30.21	31.26	33.05		26.15	27.46	31.31		19.24	32.44	33.09	
GraNd	27.04↓3.17	31.38↑0.12	33.34↑0.29		28.00↑1.85	29.79↑2.33	30.59↓0.72		20.03↑0.79	32.28↓0.16	33.05↓0.04	
EL2N	28.67↓1.54	31.26↓0.00	32.78↓0.27	35.31	27.60↑1.45	27.33↓0.13	34.91↑3.60	34.55	20.96↑1.72	30.92↓1.52	32.42↓0.67	34.36
CCS	30.19↓0.02	32.75↑1.49	33.77↑0.72		26.41↑0.26	33.59↑6.13	33.83↑2.52		19.77↑0.53	32.85↑0.41	34.07↑0.98	
Nuggets	30.39↑0.18	30.83↓0.43	33.84↑0.79		28.40↑2.25	28.02↑0.56	32.05↑0.74		19.98↑0.74	31.98↓0.46	32.46↓0.63	
STAFF	-	-	-		27.33↑1.18	29.64↑2.18	31.79↑0.48		20.88↑1.64	31.11↓1.33	31.33↓1.76	
Data Whisperer	**32.15**↑1.94	**32.81**↑1.55	**34.07**↑1.02		**31.27**↑5.12	**34.04**↑6.58	**35.20**↑3.89		**22.35**↑3.11	**35.08**↑2.64	**35.49**↑2.40	

thetic dataset, each corresponding to a distinct downstream task. Specifically, we used the following datasets: (i) the BioInstruct dataset (Tran et al., 2024) for biomedical question answering, (ii) the DialogSum dataset (Chen et al., 2021) for dialogue summarization, (iii) GSM8K (Cobbe et al., 2021) for mathematical reasoning, and (iv) a synthetic variant of DialogSum. Please see Appendix A.2 for detailed prompt designs.

Models. For each task, we evaluated the performance of three widely-used large language models (LLMs): Llama-3-8B-Instruct (Dubey et al., 2024), Qwen-2.5-7B-Instruct (Yang et al., 2024a; Team, 2024), and Mistral-Nemo-Instruct-2407 (Jiang et al., 2023). To explore the potential of using a weaker model for ICL, Data Whisperer also incorporated two smaller LLMs: Qwen-2.5-3B-Instruct (Yang et al., 2024a; Team, 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). No-

tably, since Llama-3-8B-Instruct does not have a smaller variant within its model family, it was excluded from the weak-to-strong analysis.

Baselines. We compared Data Whisperer with several state-of-the-art data selection techniques: (i) Random Selection, which randomly samples subsets from the dataset, (ii) GraNd (Paul et al., 2021), which selects samples based on large gradient norms, (iii) EL2N (Paul et al., 2021), which prioritizes samples with large discrepancies between model predictions and ground truth, (iv) CCS (Zheng et al., 2023), which balances data coverage and importance, (v) Nuggets (Li et al., 2023b), which employs one-shot learning of a fine-tuned model to identify high-quality examples, and (vi) STAFF (Zhang et al., 2024), which estimates gradient effort scores with a smaller model and validates them on the target LLM to allocate budget across different model regions.

Table 4: Ablation study on the number of demonstrations n_d and queries n_q in Data Whisperer. We observed that selecting a moderate number of n_q and n_d often yields a better performance for Data Whisperer.

Dataset	Ratio	Llama-3-8B-Instruct $n_d = 5$ / $n_q = 3$	10 / 5	15 / 10	Qwen-2.5-7B-Instruct 5 / 3	10 / 5	15 / 10
GSM8K	0%		57.49			21.23	
	1%	**62.57**	60.96	61.23	22.43	**24.45**	23.75
	5%	68.18	**69.65**	66.56	81.95	**83.01**	80.95
	10%	70.95	**72.46**	70.72	83.75	84.89	**85.03**
DialogSum	0%		18.57			22.10	
	1%	36.11	**36.47**	36.45	**32.49**	32.20	32.14
	5%	38.15	**40.96**	38.43	38.02	**40.95**	38.46
	10%	41.27	**42.18**	41.42	37.88	**43.00**	40.31
BioInstruct	0%		13.80			17.30	
	1%	**32.29**	32.24	30.45	32.34	**34.93**	32.07
	5%	32.96	**37.27**	34.01	36.74	**37.14**	36.64
	10%	38.98	**39.20**	37.67	38.20	**38.85**	38.16

Table 5: Results of different attention layers for context-aware weighting, and the impact of context-aware weighting (w/ vs. w/o) on Llama-3-8B-Instruct.

Dataset	Ratio	Random	Data Whisperer w/o Attn	Shallow	Intermediate	Deep
GSM8K	0%			57.49		
	1%	56.95	60.56	60.73	60.96	**62.30**
	5%	64.71	68.45	68.98	**69.65**	68.71
	10%	69.66	71.72	71.25	**72.46**	71.79
DialogSum	0%			18.57		
	1%	31.66	34.35	37.97	36.47	**38.05**
	5%	39.88	40.29	39.96	**40.96**	40.08
	10%	41.45	41.91	41.94	**42.18**	41.80
BioInstruct	0%			13.80		
	1%	28.38	30.51	29.27	**32.24**	29.44
	5%	34.61	36.38	36.36	**37.27**	36.32
	10%	38.70	38.83	38.86	**39.20**	38.79

Evaluation. All experiments were performed on 8 NVIDIA A100 GPUs. For model fine-tuning, we utilized LoRA (Hu et al., 2021) and performed 3 times. Performance on the BioInstruct and DialogSum datasets was assessed using the ROUGE-L metric (Lin, 2004) during both the data selection and fine-tuning stages. For the GSM8K dataset, we computed performance based on the Exact Match (EM) metric by comparing the model's generated answers against the ground truth.

Synthetic Data Generation. We generated a synthetic variant of the DialogSum dataset using the Llama-3-8B-Instruct model. For every five demonstration samples from the original dataset, we prompted the model to generate one synthetic sample, ensuring alignment with the original dialogue structure. The generated samples were manually reviewed and filtered to remove incoherent, factually incorrect, or misformatted data. See Appendix A.3 for details on synthetic data generation.

4.2 Main Results and Key Observations

• **Smaller Dataset, Comparable Performance with the Whole Dataset.** For real datasets, as shown in Table 2, results demonstrate that Data Whisperer performs remarkably well on smaller datasets, achieving comparable performance to full dataset across various selection ratios. For instance, on the GSM8K dataset, fine-tuning with 10% of data yields even better performance to the model fine-tuned on the entire dataset. For synthetic dataset in Table 3, two out of three models outperform the models fine-tuned on the entire dataset.

• **Same Size, Better Performance with SOTA Baseline Methods.** Compared to SOTA methods, Data Whisperer demonstrates consistent superior-

ity across varying dataset sizes. On real datasets, as illustrated in Table 2, Data Whisperer achieves higher accuracy. For instance, on 10% data of DialogSum with Qwen-2.5-7B-Instruct, Data Whisperer attains an accuracy of 43.00, surpassing the previous SOTA method, STAFF, by a significant margin of 2.46. Similarly, on synthetic datasets, as shown in Table 3, Data Whisperer consistently delivers the best performance across all evaluated models and data proportions, underscoring its robust generalization capabilities. Notably, with the Qwen-2.5-7B-Instruct model on 5% of the data, Data Whisperer achieves an accuracy of 31.27, outperforming the prior SOTA method, Nuggets, by a remarkable 2.87 points.

4.3 Ablation Study

• **Sensitivity of demonstration and query numbers n_d and n_q.** An interesting question is how the number of demonstration examples (n_d) and query examples (n_q) affects the performance of Data Whisperer. To explore this, we varied the values of n_d and n_q and observed their impact across all three datasets. As shown in Table 4, results indicate that while increasing n_d or n_q improves the model's performance at first, the effect tends to plateau after a certain threshold. This suggests that there exists an optimal balance between the number of demonstration and query examples, beyond which the improvements are marginal, demonstrating the robustness of Data Whisperer. We used $n_d = 10$ and $n_q = 5$ by default.

• **Ablation of Attention Layers.** The effectiveness of the context-aware weighting mechanism depends on the selected attention layer for scoring. We conducted an ablation study on the Llama-3-8B-Instruct model, evaluating shallow (Layer 5), intermediate (Layer 13), and deep (Layer 30) layers.
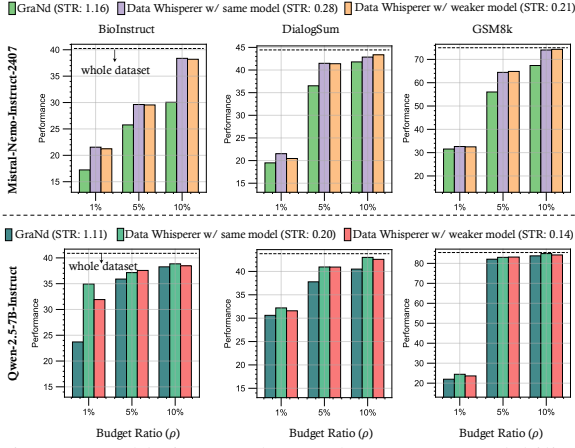
Figure 3: Ablation on the weak-to-strong scalability. For Qwen models, we used Qwen-2.5-3B-Instruct to perform ICL selection for Qwen-2.5-7B-Instruct. For Mistral models, we used Mistral-7B-Instruct-v0.2 to perform ICL selection for Mistral-Nemo-Instruct-2407. Weaker models were not fine-tuned on the task dataset. We found that using a weaker model does not significantly affect performance and provides a more efficient solution (measured by STR). **Best viewed in color.**

As shown in Table 5, the intermediate layer (*e.g.*, Layer 13) generally yielded more stable results, though all layers outperformed random selection. This suggests that Data Whisperer benefits from multiple attention layers, with intermediate layers providing more relevant contextual information for task-specific fine-tuning. These findings align with previous research showing that **intermediate layers are crucial for semantic interpretation, processing moderately complex concepts, and providing contextual information for tasks requiring understanding** (Wendler et al., 2024; Jin et al., 2025; Li et al., 2024).

• **Effect of weak-to-strong scoring**. We also investigated the impact of weak-to-strong scoring, where a weaker model is used to select data for fine-tuning a stronger model. As shown in Figure 3, results indicate that **using a weaker model does not significantly impact the overall performance of ICL selction, while providing a more efficient solution with a lower STR**. It demonstrates that Data Whisperer is scalable across different model sizes and highlights its potential for efficient fine-tuning, even with limited computational resources.

5 Discussion: What Samples Do Task-Specific Fine-Tuning Prefer?

To study what kinds of samples task-specific fine-tuning favors, we analyzed the perplexity of both the full dataset and the selected 1% subset across three different task datasets, including BioInstruct,
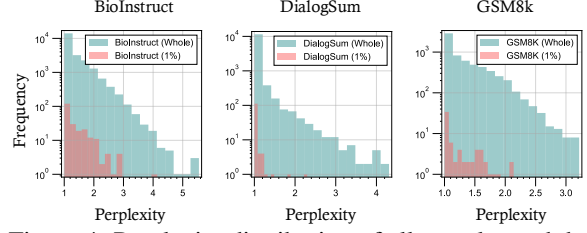
Figure 4: Perplexity distribution of all samples and the selected 1% samples by Data Whisperer. Perplexity scores are calculated using the GPT-4o-mini model API. The results show that Data Whisperer prefers selecting easier samples, which aligns with the theory in (Sorscher et al., 2022) that suggests selecting the easiest samples is optimal in small data scenarios (*e.g.*, 1% data).

GSM8K, and DialogSum. Specifically, we used GPT-4o mini (OpenAI et al., 2023) to score the perplexity of both the full data and the 1% selected subset. As shown in Figure 4, our findings show that the selected 1% of data consistently exhibits lower perplexity (*i.e.*, easier data) across all three datasets. This aligns with previous findings in (Sorscher et al., 2022), which suggest that, **in small data scenarios, the model tends to prefer easier data to boost performance**. Lower perplexity data tends to be more predictable and less ambiguous, allowing the model to learn task-specific patterns more efficiently when fine-tuned on such samples. This observation further supports the notion that task-specific fine-tuning benefits from focusing on simpler, high-confidence examples, especially in data-limited settings.

6 Conclusion

In this paper, we reevaluate existing data selection methods through the lens of the Selection-to-Tuning Ratio, revealing that many traditional approaches fall short in practical scenarios. To address this gap, Data Whisperer introduces an efficient, training-free, attention-based approach to data selection. By leveraging the theoretical connection between ICL and fine-tuning, Data Whisperer eliminates the need for fine-tuning an additional LLM on the target dataset for scoring. Specifically, we use few-shot ICL for data selection, with randomly sampled demonstration and query examples drawn from the initial dataset. Moreover, to mitigate the inherent order sensitivity in ICL, our method incorporates a context-aware weighting strategy based on attention scores. In conclusion, Data Whisperer provides a scalable, efficient solution for data selection in task-specific LLM fine-tuning, offering significant potential for enhancing both efficiency and performance.

7 Limitation

While Data Whisperer demonstrates strong performance across a range of tasks and datasets, there are opportunities for further exploration. The attention-based scoring mechanism relies on access to the internal attention layers of the LLM, which may limit its direct applicability in certain settings where such access is unavailable, such as in some commercial API-driven models (*e.g.*, GPT-4 (OpenAI et al., 2023) and DeepSeek-R1 (Guo et al., 2025)). Additionally, while our evaluations provide a solid foundation, further research is needed to explore the performance of Data Whisperer with larger LLMs and datasets. Expanding experiments to include larger and more complex models would offer valuable insights into the scalability of the method and its potential in extreme-scale scenarios. Addressing these areas in future work will help maximize the versatility and impact of Data Whisperer across a broader range of applications.

References

Rahul Atul Bhope, Praveen Venkateswaran, KR Jayaram, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. 2025. Optiseq: Optimizing example ordering for in-context learning. *arXiv preprint arXiv:2501.15030*.

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.

Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? *arXiv preprint arXiv:2406.11813*.

Bo Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. 2024. Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent. *arXiv preprint arXiv:2410.11268*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2110.14168*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*.

Gilad Deutch, Nadav Magar, Tomer Bar Natan, and Guy Dar. 2023. In-context learning and gradient descent revisited. *arXiv preprint arXiv:2311.07772*.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904, Bangkok, Thailand. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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. *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*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer.

2021. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. 2022. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2024. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, et al. 2023b. One shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. 2024. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hieu Tran, Zhichao Yang, Zonghai Yao, and Hong Yu. 2024. Bioinstruct: instruction tuning of large language models for biomedical natural language processing. *Journal of the American Medical Informatics Association*, page ocae122.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

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

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang

Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yu Yang, Siddhartha Mishra, Jeffrey N Chiang, and Baharan Mirzasoleiman. 2024b. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *arXiv preprint arXiv:2403.07384*.

Simon Yu, Liangyu Chen, Sara Ahmadian, and Marzieh Fadaee. 2024. Diversify and conquer: Diversity-centric data selection with iterative refinement. *arXiv preprint arXiv:2409.11378*.

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

Xiaoyu Zhang, Juan Zhai, Shiqing Ma, Chao Shen, Tianlin Li, Weipeng Jiang, and Yang Liu. 2024. Speculative coreset selection for task-specific fine-tuning. *arXiv preprint arXiv:2410.01296*.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. 2023. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *Preprint*, arXiv:2403.13372.

Zhenghao Zhou, Robert Frank, and R Thomas McCoy. 2024. Is in-context learning a type of gradient-based learning? evidence from the inverse frequency effect in structural priming. *arXiv preprint arXiv:2406.18501*.

A More Experimental Details

A.1 Datasets

We use three real-world datasets for corresponding downstream tasks, namely, the BioInstruct dataset (Tran et al., 2024) for biomedical question answering, the DialogSum dataset (Chen et al., 2021) for dialogue summarization, and GSM8K (Cobbe et al., 2021) for mathematical reasoning. The BioInstruct dataset contains 25,005 instruction-input-output triplets spanning diverse biomedical scenarios, including clinical decision-making, biomedical question answering, and diagnostic interpretation. We split the dataset into training and testing sets with a 9:1 ratio. The DialogSum dataset is a comprehensive collection of 13,460 dialogues curated from open dialogue repositories, addressing a variety of everyday scenarios. Each dialogue is paired with a carefully constructed reference summary, enabling robust evaluation of summarization models. The GSM8K dataset offers 8.5K linguistically diverse, high-quality grade school math word problems requiring multi-step arithmetic reasoning. This dataset is designed to facilitate research on multi-step reasoning in question answering tasks for fundamental mathematical problem solving.

A.2 In-Context Learning Prompt Template

We formalize the design of In-Context Learning templates for three datasets: BioInstruct, DialogSum, and GSM8K. Each template is structured to provide explicit guidance through task-specific instructions, demonstrations, and output constraints. **Template Design.** Our framework employs a unified template structure comprising three core components: (i) **Task Instruction.** A declarative statement that explicitly defines the task objective (*e.g.,* "Generate a concise summary of the dialogue") alongside specific constraints (*e.g.,* brevity, precision). (ii) **In-Context Demonstrations.** A set of input-output pairs that exemplify valid task execution, ensuring alignment with the task's requirements and constraints. (iii) **Output Constraints.** Rigorous formatting rules that enforce compliance with task-specific syntax, such as step-by-step derivations for GSM8K or speaker-aware summarization for DialogSum. This modular architecture ensures consistency and adaptability across diverse tasks while maintaining strict adherence to domain-specific guidelines.
Task-Specific Prompt Design. We present a systematic approach to designing In-Context Learning prompts tailored to diverse datasets. For BioInstruct, we prioritize biomedical accuracy by implementing hierarchical guidelines that emphasize logical ordering and scope limitation, while ensuring responses are strictly evidence-based and devoid of speculative content. In DialogSum, we enforce structured dialogue processing through explicit speaker tagging (*e.g.,* #Person1#), discourse compression techniques, and the generation of neutral third-person summaries. For GSM8K, we mandate a rigorous format that includes intermediate arithmetic steps and requires final answers to be boxed (*e.g.,* #### <number>), explicitly prohibiting textual explanations to maintain precision and clarity. This tailored approach ensures that each dataset's unique characteristics are effectively leveraged to optimize In-Context Learning performance. Structural schematics and formatting rules are visualized in Figure 5, 6 and 7.

You are an expert math assistant. Your role is to provide step-by-step calculations for each problem and deliver the correct final answer. Each solution should be logically structured, with no extra commentary or deviation from the required steps. Your responses must be concise, accurate, and in the exact format specified below. Your sole focus should be on solving the problem as efficiently as possible. Do not include any extraneous information.
Guidelines for your response:
1. Your response must contain only step-by-step calculations and the final answer.
2. The final output **must** be formatted as: #### <number>.
Replace `<number>` with the correct final result (either an integer or a floating-point number). No deviations or alternative formats are allowed.
3. Do not add any commentary, questions, greetings, or extra remarks.
4. Ensure your calculations are clear, concise, and correct, but only include the steps required to arrive at the final answer.
Please answer each question step by step and provide the final answer following the instructions below.

Strictly use the format specified below:
Question 1 Answer: <your step-by-step solution>
<final answer>
Question 2 Answer: <your step-by-step solution>
<final answer>
(and so on...).
Now, based on the provided questions, respond to the following mathematical problems:
Question 1: "Grace is looking to plant some lettuce in her raised bed garden. Her raised bed is comprised of 2 large beds on top with 2 medium beds on the bottom. The top bed can hold 4 rows of lettuce with 25 seeds being sown per row. The medium bed can house 3 rows with 20 seeds being sown per row. How many seeds can Grace plant in all four beds of her raised bed garden?"
Question 2: "Roberta takes $158 with her on a shopping trip to the mall. She spends $45 on new shoes, $17 less on a new bag, and a quarter of the price of the bag for lunch. How much money does she have left after these purchases?"
Question 3: "Alex was having a cookout Friday night and planned to serve burgers to his guests. He planned to cook 3 burgers for each guest and had invited 10 friends over. 1 of his friends didn't eat meat and said they would bring their own food. Another one of his friends didn't eat bread and would not need the buns. The burger buns came 8 to a pack. How many packs of buns did Alex need to buy?"
Question 4: "Jill sells girl scout cookies. This year, she wants to sell at least 150 cookie boxes. Her first customer buys 5 boxes, Her second one buys 4 times more than her first customer. Her third customer buys half as much as her second. The fourth customer buys 3 times as much as her third. Lastly, her final customer buys 10. How many boxes so Jill have left to sell to hit her sales goal?
Question 5: <Question 5>

Question 1 Answer:
The top bed can hold 4 rows of lettuce with 25 seeds being sown per row, so that's 4*25 = <<4*25=100>>100 seeds.
The medium bed can house 3 rows with 20 seeds being sown per row, so that's 3*20 = <<3*20=60>>60 seeds.
The total number of seeds that can be planted is 100 + 60 = <<100+60=160>>160 seeds.
160

Question 2 Answer:
Roberta starts with $158. She spends $45 on shoes, leaving her with $158 - $45 = <<158-45=113>>113.
She spends $17 less on a bag, so that's $45 - $17 = <<45-17=28>>28. She has $113 - $28 = <<113-28=85>>85 left.
She spends a quarter of the price of the bag for lunch, which is $28 * 0.25 = <<28*.25=7>>7. She has $85 - $7 = <<85-7=78>>78 left.
78

Question 3 Answer:
Alex invited 10 friends, but 1 didn't eat meat and 1 didn't eat bread, so he only needs to cook for 10 - 2 = <<10-2=8>>8 friends.
He planned to cook 3 burgers for each guest, so that's 8 * 3 = <<8*3=24>>24 burgers.
The buns come 8 to a pack, so he needs to buy 24 / 8 = <<24/8=3>>3 packs of buns.
3

Question 4 Answer:
Jill sells 5 boxes to her first customer, then 5 * 4 = <<5*4=20>>20 boxes to her second customer, then 20 / 2 = <<20/2=10>>10 boxes to her third customer, then 10 * 3 = <<10*3=30>>30 boxes to her fourth customer, and finally 10 boxes to her final customer.
In total, she sells 5 + 20 + 10 + 30 + 10 = <<5+20+10+30+10=75>>75 boxes.
She needs to sell at least 150 boxes, so she has 150 - 75 = <<150-75=75>>75 boxes left to sell.
75

Question 5 Answer: ...

Figure 5: ICL prompt of GSM8k dataset.

You are a medical expert. Given an input and an instruction, your objective is to respond with the correct and concise answer based on the provided context. Ensure clarity and accuracy in your answers.
Guidelines for your response:
1. **Ensure your responses are concise, clear, and focused on the provided instruction.** Provide clear, correct answers, without unnecessary elaboration.
2. **Follow the logical order of questions.** Do not skip or merge responses.
3. **Stay within the scope of the question.** Avoid additional commentary or unrelated details.
Please respond to the questions based on the given instructions and demonstrations below.

Input:
Below are some demonstrations of how to format your answers:
Instruction: <Demonstration 1 Instruction>
Input: <Demonstration 1 Input>
Answer: <Demonstration 1 Answer>
...

Strictly use the format specified below:
Question 1 Answer: <your answer to Question 1.>
Question 2 Answer: <your answer to Question 2.>
(and so on...).
Now, based on the biomedical demonstrations provided, respond to the following biomedical questions:
Question 1: Instruction: "Given a specific healthcare topic or issue, explain potential ethical considerations and implications." Input: "What are potential ethical considerations related to the use of genetic testing in healthcare, and what are the implications of these considerations?"
Question 2: Instruction: "Identify the primary role of a specified healthcare team member." Input: "Registered dietitian"
Question 3: Instruction: "Given a set of genetic test results, interpret the results and explain their implications for a patient's health and family members." Input: "A patient tests positive for a BRCA1 gene mutation."
Question 4: Instruction: "Describe the recommended lifestyle changes for a patient with high blood pressure." Input: "High blood pressure"
Question 5: Instruction: "Identify the key findings or conclusions stated in the clinical research article and summarize them in a few sentences." Input: "Summarize the key findings of the randomized controlled trial investigating the efficacy and safety of a new antihypertensive drug."

Output:
Here are the answers to the biomedical questions:

Question 1 Answer: Potential ethical considerations related to the use of genetic testing in healthcare include the potential for genetic discrimination, the risk of false positive or false negative results, and the potential for unintended consequences for individuals and families. The implications of these considerations include the need for careful counseling and informed consent, the importance of ensuring that genetic testing is used in a responsible and ethical manner, and the need for ongoing research to improve the accuracy and utility of genetic testing.

Question 2 Answer: The primary role of a registered dietitian is to provide medical nutrition therapy to patients, which involves assessing a patient's nutritional needs, developing personalized nutrition plans, and educating patients on how to implement these plans to achieve optimal health outcomes.

Question 3 Answer: A patient who tests positive for a BRCA1 gene mutation is at increased risk for developing breast and ovarian cancer. The implications of this result are significant, as it may require the patient to undergo increased surveillance, including regular mammograms and pelvic exams, and may also impact their reproductive choices. The patient may also need to consider genetic testing for their family members, as the mutation can be inherited. The patient should be referred to a genetic counselor to discuss these implications and develop a personalized plan for managing their risk.

Question 4 Answer: Lifestyle changes recommended for a patient with high blood pressure include increasing physical activity, reducing sodium intake, losing weight if overweight or obese, and limiting alcohol consumption. Additionally, patients with high blood pressure should aim to reduce their stress levels, get adequate sleep, and avoid tobacco use. A healthy diet that is low in saturated and trans fats, added sugars, and refined carbohydrates can also help to manage blood pressure.

Question 5 Answer: The key findings of the randomized controlled trial investigating the efficacy and safety of a new antihypertensive drug were that the drug was effective in reducing systolic blood pressure by an average of 10 mmHg compared to placebo, and was well-tolerated with few adverse effects. The study also found that the drug was more effective in reducing blood pressure in patients with more severe hypertension, and that it did not increase the risk of cardiovascular events or mortality.

Figure 6: ICL prompt of DialogSum dataset.

Instruction:
You are an expert assistant. Your task is to provide clear, concise, and complete summaries for the given dialogues. Your summaries should accurately capture the main points of each dialogue. Avoid unnecessary details and ensure clarity.
Guidelines for your response:
1. **Summarize the dialogue concisely and fully**, ensuring all main points are captured.
2. **Avoid adding extra commentary or irrelevant details** that are not part of the dialogue content.
3. If a dialogue is unclear, incomplete, or lacks meaningful content, respond with "No valid content to summarize."
4. **Ensure every summary field is filled out.** Leaving any field blank is not allowed.
Please summarize dialogues based on the given instructions and demonstrations below.

Input:
Below are some demonstrations of how to format your answers:
Dialogue: <Dialogue 1>
Summary: <Summary 1>
...

Strictly use the format specified below:
Summary 1: <Your summary to Dialogue 1.>
Summary 2: <Your summary to Dialogue 2.>
(and so on...).
Now, based on the provided dialogues, provide concise and complete summaries for the following dialogues:
Dialogue 1: "#Person1#: We've been cramming for tomorrow's history exam since early this morning. What do you say we take a break and listen to some music, okay?
#Person2#: Now that you mention it, I'm getting a little bumed-out from studying nonstop, too. Listening to some music for a while would suit me just fine.
#Person1#: While you're picking out a record to play, I'll grab a couple of beers out of the refrigerator.
#Person2#: You sure have a lot of discs here.
#Person1#: Yeah, I've got everything from rock n'roll to the latest new - wave stuff.
#Person2#: To tell you the truth, I'm strictly into classical music. You don't happen to have any Bach or Mozart, do you?
#Person1#: Sorry, my taste in music doesn't go back any further than the 1960's. Music written before then is just history to me.
#Person2#: Well, speaking of history, let's get back to the books. We'Ve got an exam tomorrow, remember?
"
Dialogue 2: "#Person1#: Excuse me. Have you got any books on genetic engineering? I've looked for them everywhere, but I can't find any.
#Person2#: I'm not sure if we've got any now. Books of that kind are on this shelf. Well, I'm afraid we've sold out.
#Person1#: What a pity! I should have come earlier.
#Person2#: Come some time next week if you like. I think we'll have got some by then.
#Person1#: Ok, I will.
"
Dialogue 3: "#Person1#: Excuse me. Does this bus go to the National Library of China?
#Person2#: Yes, there are three stops left.
#Person1#: How long is the ride?
#Person2#: Well, you will get there in about a quarter of an hour if the traffic isn't too heavy.
#Person1#: Thank you. Would you please let me know when we get to the stop? I'm a stranger here.
#Person2#: Sure, madam. I'll remind you.
Dialogue 4: <Dialogue 4>
Dialogue 5: <Dialogue 5>

Output:
Here are the summaries for the given dialogues:

Summary 1: #Person1# and #Person2# take a break from studying for a history exam to listen to music. #Person2# prefers classical music, but #Person1# only has music from the 1960s and later. They eventually decide to get back to studying.

Summary 2: #Person1# is looking for books on genetic engineering at a bookstore, but they are sold out. #Person2# suggests checking back next week when new stock may be available.

Summary 3: #Person1# asks #Person2# if the bus goes to the National Library of China and how long the ride will take. #Person2# confirms that the bus goes to the library and estimates the ride will take about 15 minutes.

Summary 4: ...
Summary 5: ...

Figure 7: ICL prompt of BioInstruct dataset.

A.3 Synthetic Data Generation

To evaluate the robustness of Data Whisperer on synthetic data, we constructed a synthetic variant of the DialogSum dataset using the Llama-3-8B-Instruct model. The generation pipeline comprised three key stages: *prompt design*, *generation process*, and *quality control*.

Prompt Design. For every 5 demonstration samples from the original DialogSum dataset, we crafted structured prompts to generate 1 synthetic dialogue-summary pair. These prompts enforced strict formatting rules, including speaker tagging and topic alignment, alongside explicit constraints to ensure the synthetic data closely mirrored the style and structure of the original dataset.

Generation Process. The model was instructed to produce dialogues with multiple turns, ensuring natural conversational flow and coherent narratives. Each output adhered to the JSON schema, with topics covering diverse everyday scenarios such as travel, shopping, and education.

Quality Control. A two-step filtering process was employed to ensure data quality. First, automated checks removed duplicates and syntax-invalid entries using regex-based validation. Second, three annotators independently reviewed the samples, flagging those with coherence issues, factual inconsistencies, or formatting violations. Disagreements were resolved through majority voting, and only samples free from hallucination or structural deviations were retained.

The final synthetic dataset maintained the original DialogSum distribution in dialogue length and topic diversity. Full prompt template is illustrated in Figure 8.

A.4 Hyper-Parameters of Different Methods

Data Selection. In our experiments, Data Whisperer was configured with task-specific batch sizes: 15 for the BioInstruct dataset, 5 for the DialogSum dataset, and 15 for the GSM8K dataset. By default, n_d (number of demonstrations) was set to 10, and n_q (number of queries) was set to 5. The generation temperature was fixed at 0 during the In-Context Learning process to ensure deterministic outputs. To ensure fairness, the batch size for all baseline methods was uniformly set to 16 across all tasks.

Model Fine-tuning. All models, whether trained on pruned or full datasets, shared identical hyper-parameters. We employed parameter-efficient fine-tuning via LoRA (Hu et al., 2021) for all tasks.

The fine-tuning framework, based on (Zheng et al., 2024)[2], utilized a learning rate scheduler with linear warm-up and cosine decay. A consistent batch size of 8 was applied across all experiments. For Llama-3-8B-Instruct, the learning rate was fixed at 1×10^{-4} for all datasets. For Qwen-2.5-7B-Instruct and Mistral-Nemo-Instruct-2407, the learning rate was uniformly set to 1×10^{-5}. All models were fine-tuned for 5 epochs, with each experiment repeated three times to ensure robustness and statistical reliability.

A.5 Pseudo Code of Data Whisperer

The detailed pseudo code of Data Whisperer is demonstrated in Algorithm 1.

B Extensive Evaluation Results of STR

As defined in Eq. (1), the *Selection-to-Tuning Ratio* (STR) quantifies the efficiency of data selection methods by comparing the computational cost of selection to the time saved during fine-tuning. A method τ is efficient if STR < 1, indicating that the time invested in selection is offset by faster fine-tuning. An STR below 1 ensures that the selection process positively contributes to overall pipeline efficiency, making it a practical and scalable choice for optimizing model training workflows. More evaluation results of STR are demonstrated in Table 6.

You are an expert assistant tasked with generating new dialogue-summary pairs for the DialogSum dataset. The dialogue should be natural and involve two or more participants, with a clear and meaningful conversation. Your summary should accurately capture the main points of the dialogue, ensuring clarity and conciseness.
Guidelines for generating data:
1. **Generate a natural, coherent dialogue** involving two or more participants. Ensure the dialogue is meaningful and engaging.
2. **Create a concise, accurate summary** that captures the main points of the dialogue without introducing unnecessary details or commentary.
3. If the dialogue lacks clarity, completeness, or meaningful content, respond with "No valid content to summarize." in the summary.
4. **Ensure the generated dialogue and summary pair are returned in the correct JSON format.** Make sure that both the dialogue and summary are accurate and coherent.
5. **Generate new and unique data**. Each dialogue-summary pair should be fresh and differ from previously provided content, maintaining relevance to the topic.
The generated data should be returned in a JSON object format.

Input:
Below are the original demonstrations of how to format your samples:
Dialogue: <Dialogue 1>
Summary: <Summary 1>
Dialogue: <Dialogue 2>
Summary: <Summary 2>
...

Strictly use the format specified below to generate new samples in JSON format:
{"dialogue": "New dialogue", "summary": "Summary of the dialogue", "topic": "Topic of the dialogue"}
Now, based on the provided demonstrations, generate a new dialogue and the corresponding summary in JSON format:

Output:
Here is a new dialogue and summary in JSON format:
{
"dialogue": "#Person1#: I'm so excited to be here at the art museum. I've never seen so many masterpieces in one place.
#Person2#: Yes, it's amazing. I've been here many times, but I still find something new to appreciate each time.
#Person1#: What's your favorite piece here?
#Person2#: Ah, that's a tough one. I think it's the Monet water lilies. The way he captures the light is just breathtaking.
#Person1#: I agree. I've always loved his work. Do you think we could get a closer look at it?
#Person2#: Of course. Let me just check if it's allowed. (checks the sign) Yes, we can get up close.
#Person1#: Wow, it's even more beautiful up close. I can see all the tiny brushstrokes.
#Person2#: Exactly. That's what I love about Impressionism. The attention to detail is incredible.
#Person1#: I never knew that. I'll have to look into it more.
#Person2#: Definitely. It's a fascinating style.
#Person1#: I'm so glad we came here today. I feel like I've learned something new.
#Person2#: Me too. And it's always great to share that with someone who appreciates it as much as I do.",
"summary": "Two art enthusiasts visit the museum and discuss their favorite piece, Monet's water lilies. They appreciate the attention to detail in Impressionist art and enjoy getting a closer look at the painting.",
"topic": "Art and Appreciation"
}

Figure 8: Data synthesis prompt of DialogSum dataset.

Algorithm 1 Data Whisperer for Data Selection

Input:
$\mathcal{M}_p$ – Pre-trained language model
$\mathcal{D}$ – Full task dataset
n_d – Number of samples for demonstration at a time
n_q – Number of samples for query at a time
I – Instruction for ICL generation
f – Performance metric
ρ – Budget ratio
Output: $\mathcal{D}'$ – Selected coreset ($\mathcal{D}' \subseteq \mathcal{D}, |\mathcal{D}'| < |\mathcal{D}|$)

1: **procedure** DATAWHISPERER($\mathcal{M}_p, \mathcal{D}, n_d, n_q, I, f, \rho$)
2:     $\mathcal{S} \leftarrow \mathbf{0}$ ▷ Initialize the score set
3:     **while** Not all samples in $\mathcal{D}$ have been processed **do**
4:         $\mathcal{D}_d \leftarrow$ Randomly select n_d samples from $\mathcal{D}$
5:         $\mathcal{D}_q \leftarrow$ Randomly select n_q samples from $\mathcal{D}$
6:         $C \leftarrow \{I, \mathcal{D}_d\}$ ▷ Form input context
7:         $\hat{y}_q^{(1)}, \ldots, \hat{y}_q^{(n_q)} \leftarrow \mathcal{M}_p(C, \mathcal{D}_q)$ ▷ Generate query predictions via ICL
8:         $s \leftarrow \frac{1}{n_q} \sum_{j=1}^{n_q} f(\hat{y}_q^{(j)}, y_q^{(j)})$
9:         **for** $i = 1 \rightarrow n_d$ **do**
10:             Let h denote each head in the attention mechanism
11:             $w_i \leftarrow \sum_h \mathbf{1}^\top A^{(h)}_{(x_d^{(i)}, y_d^{(i)})} \mathbf{1}$ ▷ Compute weights
12:         **end for**
13:         $w^* \leftarrow \text{Normalize}(w)$
14:         $s^* \leftarrow s \odot w^*$ ▷ Element-wise multiplication with attention weights
15:         $\mathcal{S}[\mathcal{D}_d] \leftarrow \mathcal{S}[\mathcal{D}_d] + s^*$ ▷ Accumulate scores for corresponding indices in $\mathcal{D}_d$
16:     **end while**
17:     $k \leftarrow \lfloor \rho \cdot |\mathcal{D}| \rfloor$
18:     $\mathcal{D}' \leftarrow \text{Top-k}(\mathcal{D}; \mathcal{S})$ ▷ Select top m samples based on scores
19:     **return** $\mathcal{D}'$
20: **end procedure**

Table 6: Additional results on the Selection-to-Tuning ratio and performance scores for the Qwen and Mistral models are presented. The symbols ↑ and ↓ indicate improvements and degradations compared to random selection, respectively. "Speedup" refers to the acceleration achieved by Data Whisperer (w2s) over Nuggets. "w2s" denotes using the weaker model within the same family for data selection across each LLM.

Model	Method	GSM8K STR 1%	GSM8K STR 5%	GSM8K STR 10%	GSM8K Performance Avg. Score	DialogSum STR 1%	DialogSum STR 5%	DialogSum STR 10%	DialogSum Performance Avg. Score	BioInstruct STR 1%	BioInstruct STR 5%	BioInstruct STR 10%	BioInstruct Performance Avg. Score
Qwen-2.5-7B-Instruct	Random	-	-	-	60.29	-	-	-	34.66	-	-	-	35.19
	GraNd	1.06	1.06	1.07	62.61↑2.32	1.09	1.11	1.13	36.29↑1.63	1.07	1.08	1.34	32.62↓2.57
	EL2N	1.06	1.06	1.07	60.72↑0.43	1.10	1.13	1.15	36.97↑2.31	1.08	1.08	1.32	34.64↓0.55
	CCS	1.04	1.04	1.05	61.34↑1.05	1.01	1.04	1.06	35.30↑0.64	1.02	1.03	1.25	31.87↓3.32
	Nuggets	1.10	1.11	1.11	62.74↑2.45	3.66	3.69	3.71	36.63↑1.97	2.19	2.20	2.41	34.95↓0.24
	STAFF	1.05	1.05	1.06	62.74↑2.45	1.06	1.08	1.10	37.18↑2.52	1.06	1.06	1.33	32.97↓2.22
	Data Whisperer	**0.08**	**0.08**	**0.08**	**64.21↑3.92**	**0.16**	**0.19**	**0.29**	**38.97↑4.31**	**0.15**	**0.27**	**0.46**	**37.12↑1.93**
	Data Whisperer (w2s)	**0.05**	**0.05**	**0.06**	**63.67↑3.38**	**0.10**	**0.13**	**0.23**	**38.37↑3.71**	**3.71**	**0.20**	**0.37**	**35.98↑0.79**
	Speedup	21.52×	21.12×	19.29×	-	35.67×	27.48×	16.12×	-	25.12×	10.94×	6.48×	-
Mistral-Nemo-2407-12B	Random	-	-	-	50.36	-	-	-	32.61	-	-	-	25.00
	GraNd	1.08	1.09	1.24	51.65↑1.29	1.05	1.07	1.09	32.72↑0.11	1.13	1.23	1.40	24.35↓0.65
	EL2N	1.11	1.13	1.36	51.74↑1.38	1.06	1.09	1.11	30.75↓1.86	1.14	1.23	1.39	23.76↓1.24
	CCS	1.04	1.01	1.22	50.71↑0.35	1.01	1.04	1.05	31.75↓0.86	1.02	1.11	1.25	26.37↑1.37
	Nuggets	2.34	2.35	2.50	53.06↑2.70	2.22	2.24	2.26	32.53↓0.08	5.15	5.23	5.36	28.12↑3.12
	STAFF	1.06	1.08	1.22	54.06↑3.70	1.04	1.06	1.07	32.71↑0.10	1.10	1.20	1.39	25.03↑0.03
	Data Whisperer	**0.24**	**0.26**	**0.30**	**57.62↑7.26**	**0.08**	**0.12**	**0.25**	**35.45↑2.84**	**0.24**	**0.39**	**0.65**	**29.96↑4.96**
	Data Whisperer (w2s)	**0.11**	**0.12**	**0.12**	**57.21↑6.85**	**0.07**	**0.11**	**0.24**	**35.07↑2.46**	**0.18**	**0.33**	**0.60**	**29.66↑4.66**
	Speedup	20.52×	20.45×	20.89×	-	31.75×	20.69×	9.53×	-	29.18×	15.61×	8.95×	-