MINT 🍃: Multi-Constrained Coreset Selection for Efficient Instruction Tuning in LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Instruction tuning plays a crucial role in training large language models (LLMs). By enhancing the model's ability to follow instructions, it makes the model better understand and respond to user prompts. Nowadays, many high-quality instruction tuning datasets have been constructed, but few studies have explored how to efficiently utilize these high-quality datasets during supervised fine-tuning (SFT). This work aims to select a subset of instruction examples that achieve similar model performance on all downstream tasks as using the full instruction dataset. Therefore, it significantly improves training efficiency. Specifically, we propose a coreset-based approach that takes into consideration the difference of the instruction examples in improving the model's instruction-following capability. The key idea is inspired by our theoretical finding that in instruction tuning, the training loss can be decomposed into two components that effectively quantify the contribution of an instruction to the two fundamental capabilities of LLMs, namely *knowledge-related* capability and *instruction following* capability. We then revisit the objective of the classical coreset approaches to balance the two capabilities when selecting instruction examples. Leveraging the submodular property of this optimization problem, we design an efficient algorithm to achieve a bounded approximation error. Experiments on WizardLM AND aLPACA-gpt4 across 10 downstream tasks demonstrate that `MINT` reduces computational costs by $3\times$ on LLaMA-3.1-8B and Mistral-7B. Code and data is available at `https://anonymous.4open.science/r/MINT-2545`.

1 Introduction

Recently, Large Language Models (LLMs) have significantly advanced the field of artificial intelligence [57, 17, 30]. The instruction tuning process, also known as Supervised Fine-tuning (SFT), has notably improved the ability of these models to follow human instructions [36] and efficiently tailor LLMs to particular domains [25, 48].

Recent research [48, 23, 28] has shown that when fine-tuning the model for a particular domain, it becomes essential to carefully select instruction examples, as many existing instruction-tuning datasets include a large portion of examples that are irrelevant or even harmful to the target domain [48, 43, 21]. On the other hand, this *data selection* problem is largely *overlooked* when the goal of instruction tuning is to improve the capacity of LLMs to follow instructions *in general* [9, 16, 52]. However, fine-tuning LLMs on large training corpora is computationally expensive and time-consuming [55, 19, 35]. Many small organizations cannot afford it. This underscores the necessity of data selection in this scenario. That is, if we were able to *select a high-quality, representative subset of data* (a.k.a, the *coreset* [10, 34]), on which fine-tuning an LLM would produce a model with its performance competitive to a model fine-tuned on the whole training set, it would significantly improve training efficiency and reduce cost.

To fill this gap, we propose `MINT`, a novel coreset selection framework. The key idea is inspired by our observation that different instruction examples impact model performance differently. Generally

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

speaking, LLMs exhibit two fundamental capabilities [55, 36, 25]: knowledge-related capability (i.e., the generation content contains correct knowledge of the real world) and instruction following capability (i.e., guiding the models to follow diverse task instructions and producing the corresponding desired outputs). In the pre-training stage, LLMs have already well captured the real-world knowledge. Therefore, in the SFT stage, training should focus more on the instruction following capability than on learning new knowledge. For instance, certain instruction-response examples (e.g., Q: "Provide the orbital period of the Moon around the Earth." A: "Approximately 27.3 days.") emphasize factual knowledge already acquired during pre-training, while others (e.g., Q: "Prepare a report following these specified steps." A: "[Detailed step-by-step report.]") specifically enhance the model's capability to understand and accurately follow diverse instructions.

Therefore, when selecting instruction examples, an ideal strategy should take into account the difference between the examples. However, existing coreset selection methods [31, 32, 39, 54] treat all examples equally. Directly applying these methods to select instruction examples tends to yield suboptimal performance. Designing such an ideal strategy is challenging, as it cannot simply overlook knowledge-related capability. Instead, it should *judiciously select a coreset* that balances the two capabilities, while preserving the merit of classical coreset selection methods, i.e., the selected coreset should closely approximate the full train set with a theoretical guarantee with respect to the performance of the trained model.

The primary principle of coreset selection is to select a weighted subset to approximate the gradient of data instances in the full training set. An approximation bound is achieved by limiting the gradient approximation error (GA error), which measures the difference between the gradient of the full dataset and the weighted sum of the gradients computed from the coreset. The theoretical foundation of MINT is that the overall training loss of SFT can be decomposed into two components that respectively quantify each data instance's contribution to (1) knowledge-related capability and (2) instruction following capability. Leveraging this theory, MINT selects a coreset that naturally trades off between the two capabilities by computing the respective gradients of these two parts and aggregating these two pieces to approximate the full gradient.

To be specific, we first define an optimal coreset selection problem with a dual-constraint where each constraint limits the GA error with respect to either knowledge-related capability or instruction following capability. This effectively retains the instruction following capability, while at the same time mitigating the degradation of the base model's knowledge-related capability. Then, we prove that this problem can be reduced to a single-constraint problem with a submodular property, allowing us to design an efficient coreset selection strategy with a bounded approximation error.

Contributions. We summarize our main contributions as follows:

1. We theoretically quantify the impact of each instruction–response corpus on the model's knowledge-related capability and instruction following capability by decomposing the loss function.

2. By formalizing and solving a dual-constraint minimization problem for the gradient approximation error, we convert a large instruction tuning dataset to a smaller subset, which preserves its instruction following capability without sacrificing the overall model performance. This effectively reduces the complexity of training.

3. Experiments on several advanced LLMs and real world datasets show that MINT selects a well-performing coreset, achieving both data-effective and data-efficient results.

2 Preliminary

Supervised Finetuning (SFT). Suppose that θ denotes the model parameter and D denotes the full training dataset. During the SFT stage, the full training dataset D with N data instances can be represented as an instruction–response corpus $D = \{(x_i, y_i)\}_{i=1}^N$. For each data instance, the instruction sequence is x_i and the response sequence is y_i, where $|y_i|$ denotes the length of the response sequence y_i (*i.e.*, how many tokens it contains), y_i^t denotes the t-th token in y_i and $y_i^{<t} = (y_i^1, \ldots, y_i^{t-1})$ denotes the prefix sequence before the t-th token in sequence y_i. Thus, the SFT loss can be written as:

$$\mathcal{L}_{\text{SFT}}\big((x_i, y_i); \theta\big) \;=\; -\frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \log p_\theta\big(y_i^t \,\big|\, (x_i, y_i^{<t})\big). \tag{1}$$

Coreset. The state-of-the-art coreset selection framework [10, 34] selects the smallest number of data instances $C \subseteq D$ (with non–negative weights $\{\omega_j\}_{j=1}^{|C|}$). Formally, the objective is:

$$\min_{C\subseteq D,\omega_j\geq 0}|C| \quad s.t. \quad \max_{\theta\in\Theta} \underbrace{\| \underbrace{\sum_{i=1}^{|D|}\nabla\mathcal{L}_i(\theta)}_{\text{full gradient}} - \underbrace{\sum_{j=1}^{|C|}\omega_j\nabla\mathcal{L}_{\gamma(j)}(\theta)\|}_{\text{coreset gradient}}}_{\text{gradient approximation error}} \leq \varepsilon \tag{2}$$

Here, the index mapping $\gamma(j) = i$ ($j \in [1, K], i \in [1, N]$) denotes that the j-th data instance (x_j, y_j) in C is the i-th item in D. In practice, directly specifying a fixed error bound ε as in the optimization objective (2) is challenging. Instead, it is often more practical to specify the coreset size $|C|$ to K and minimize the gradient approximation error. This leads to a dual formulation of the original problem, where the objective becomes:

$$\min_{C\subseteq D,\omega_j\geq 0}\max_{\theta\in\Theta} \|\sum_{i=1}^{|D|}\nabla\mathcal{L}_i(\theta) - \sum_{j=1}^{|C|}\omega_j\nabla\mathcal{L}_{\gamma(j)}(\theta)\| \quad s.t. \quad |C| \leq K \tag{3}$$

Next, we use an example to better illustrate the main idea of coreset.

`EXAMPLE`: Let's consider a toy dataset containing 8 data instances with gradients $\{\nabla\mathcal{L}_i(\theta)\}_{i=1}^{8}$. Suppose that for any θ, $\nabla\mathcal{L}_1(\theta) \approx \nabla\mathcal{L}_2(\theta) \approx \nabla\mathcal{L}_3(\theta)$, $\nabla\mathcal{L}_4(\theta) \approx \nabla\mathcal{L}_5(\theta) \approx \nabla\mathcal{L}_6(\theta)$, $\nabla\mathcal{L}_7(\theta) \approx \nabla\mathcal{L}_8(\theta)$. In this case, based on Equation 3, if one wants to find an optimal coreset with a size of 3, *i.e.*, K = 3, the optimal solution could be $C^* = \{(x_2, y_2), (x_5, y_5), (x_8, y_8)\}$, where $\gamma(1) = 2, \gamma(2) = 5, \gamma(3) = 8$ and $\omega_1 = 3, \omega_2 = 3, \omega_3 = 2$. That is, C^* is the optimal coreset that well approximates the full gradient because $\left\|\sum_{i=1}^{8}\nabla\mathcal{L}_i(\theta) - \sum_{j=1}^{3}\omega_j\nabla\mathcal{L}_{\gamma(j)}(\theta)\right\|$ is minimized, which is close to 0.

3 The `MINT` Approach

3.1 Theoretical Foundation: SFT Loss Decomposition

By adding $\log p_\theta(y^t \mid y^{<t})$, the Equation 1 can be written as:

$$\mathcal{L}_{\text{SFT}}\big((x, y); \theta\big) = -\frac{1}{|y|}\sum_{t=1}^{|y|}\Big[\log p_\theta\big(y^t \mid (x, y^{<t})\big) + \log p_\theta\big(y^t \mid y^{<t}\big) - \log p_\theta\big(y^t \mid y^{<t}\big)\Big] \tag{4}$$

$$= -\frac{1}{|y|}\sum_{t=1}^{|y|}\Big[\log p_\theta\big(y^t \mid y^{<t}\big) + \log\frac{p_\theta\big(y^t \mid (x, y^{<t})\big)}{p_\theta\big(y^t \mid y^{<t}\big)}\Big] \tag{5}$$

$$= \underbrace{-\frac{1}{|y|}\sum_{t=1}^{|y|}\log p_\theta\big(y^t \mid y^{<t}\big)}_{\mathcal{L}_{\text{PT}}(y;\theta)} + \underbrace{-\frac{1}{|y|}\sum_{t=1}^{|y|}\log\frac{p_\theta\big(y^t \mid (x, y^{<t})\big)}{p_\theta\big(y^t \mid y^{<t}\big)}}_{\mathcal{L}_{\text{IFL}}(y \mid x;\theta)}. \tag{6}$$

where the first component $\mathcal{L}_{\text{PT}}(y; \omega)$ denotes **pre-training loss**, because it is in the same format of the pretraining loss of LLMs [35]. It represents the negative log-likelihood of predicting the next token y^t given only the previous tokens $y^{<t}$. This component measures how well the model predicts the response tokens without using the instruction x. Therefore, it mainly reflects knowledge inherently stored in the answers.

Next, we show that the second component represents the **instruction following loss**. It measures the log-probability ratio of generating y with instruction x versus without it. This captures the additional information provided by the instruction x that helps produce the correct response y, effectively quantifying to what extent the instruction improves the model's ability to generate the desired output.

To prove this, we first introduce a metric, called *Instruction-Following Difficulty (IFD)* [23, 24, 22], which, given an instruction (x, y), identifies discrepancies between the expected responses of a model and its generation capability.

$$\text{IFD}(y \mid x;\theta) = \frac{\text{PPL}(y \mid x;\theta)}{\text{PPL}(y;\theta)} \tag{7}$$

where $\text{PPL}(y \mid x;\theta) = \exp\left(-\frac{1}{|y|}\sum_{t=1}^{|y|}\log p_\theta(y^t \mid (x, y^{<t}))\right) = \exp\left(\mathcal{L}_{\text{SFT}}((x,y);\theta)\right)$, and $\text{PPL}(y;\theta) = \exp\left(-\frac{1}{|y|}\sum_{t=1}^{|y|}\log p_\theta(y^t \mid y^{<t})\right) = \exp\left(\mathcal{L}_{\text{PT}}(y;\theta)\right)$.

Intuitively, IFD measures the potential of this instruction to improve the instruction following capability of a model. The key observation here is that taking a log on $IFD(y \mid x;\theta)$ will derive the exact form of $\mathcal{L}_{\text{IFL}}(y \mid x;\theta)$, i.e., the second component in Equation 4.

$$\mathcal{L}_{\text{IFL}}(y \mid x;\theta) = \log \text{IFD}(y \mid x;\theta). \tag{8}$$

This shows that $\mathcal{L}_{\text{IFL}}$ primarily focuses on fully exploring the training examples to improve the instruction following capacity of a model, thus representing the *instruction following loss*.

Notably, $\mathcal{L}_{\text{IFL}}$ is derived directly from the SFT loss, namely the actual objective used during training. This thus *for the first time* theoretically justifies the effectiveness of *instruction-following difficulty (IFD)* and, in turn, proves the critical role of training examples in improving the instruction following capability of a model.

In summary, this decomposition highlights that the SFT loss reflects two core model capabilities: knowledge-related capability, driven by $\mathcal{L}_{\text{PT}}$, and instruction following capability, driven by $\mathcal{L}_{\text{IFL}}$.

Problem Definition. Similar to traditional coreset methods, our goal is to find the smallest coreset C that represents the full dataset D, such that the gradient approximation error between C and D remains within a user-specified budget ε. Specifically, for the SFT stage of large models, the constraint on the right-hand side of Eq. 2 becomes:

$$\max_{\theta\in\Theta}\left\|\sum_{i=1}^{|D|}\nabla\mathcal{L}_{\text{SFT}}((x_i,y_i);\theta) - \sum_{j=1}^{|C|}\omega_j\nabla\mathcal{L}_{\text{SFT}}((x_{\gamma(j)},y_{\gamma(j)});\theta)\right\| \le \varepsilon. \tag{9}$$

Note that by Sec. 3.1 the SFT loss decomposes into PT and IFL components. To achieve finer control over the gradient approximation errors within the budget ε, and to balance the contributions of PT and IFL components in subset selection, we introduce a variable α to allocate this budget. This leads to a dual-constraint optimization objective:

$$\min_{C\subseteq D,\ \omega_j\ge 0\ \forall j} |C| \quad \text{s.t.} \begin{cases} \max_{\theta\in\Theta}\left\|\sum_{i=1}^{|D|}\nabla\mathcal{L}_{\text{PT}}(y_i;\theta) - \sum_{j=1}^{|C|}\omega_j\nabla\mathcal{L}_{\text{PT}}(y_{\gamma(j)};\theta)\right\| \le \alpha\varepsilon, \\ \max_{\theta\in\Theta}\left\|\sum_{i=1}^{|D|}\nabla\mathcal{L}_{\text{IFL}}(y_i|x_i;\theta) - \sum_{j=1}^{|C|}\omega_j\nabla\mathcal{L}_{\text{IFL}}(y_{\gamma(j)}|x_{\gamma(j)};\theta)\right\| \le (1-\alpha)\varepsilon. \end{cases} \tag{10}$$

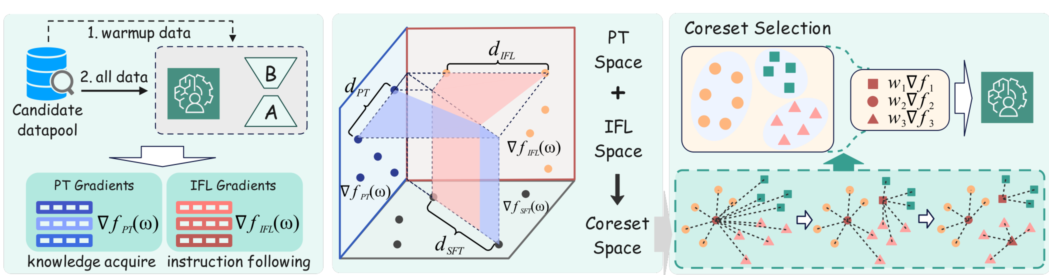


Figure 1: The Overall Framework of `MINT`.

3.2 `MINT` Overview

As shown in Figure 1, `MINT` is an efficient coreset selection framework customized to the SFT stage of LLMs. Specifically, we first randomly sample data instances from the candidate data pool D to warm up the base model. This enables the model to learn the general data distribution while simultaneously obtaining the initial Low-Rank Adaption (LoRA) parameters and the optimizer state. We then leverage this learned model to infer data instances and compute the loss and gradients. In particular, we decompose the overall training loss $\mathcal{L}_{\text{SFT}}$ into two components $\mathcal{L}_{\text{PT}}$ and $\mathcal{L}_{\text{IFL}}$ (Equation 4), and compute their gradients $\nabla\mathcal{L}_{\text{PT}}$ and $\nabla\mathcal{L}_{\text{IFL}}$ separately during backpropagation. Subsequently, typical coreset frameworks select the coreset C based on the gradient distance, which denotes the normed difference between gradient vectors computed from different data instances. The goal is to ensure that the selected instances in the coreset C can effectively represent the entire dataset D. In our case, since we have two types of gradients, we can derive two corresponding types of gradient distance, i.e., d^{PT} and d^{IFL}. Afterwards, in Section 3.3, we theoretically prove that we can compute a weighted distance d^{SFT} based on d^{PT}, d^{IFL} and α, which can serve as the gradient distance for solving Equation 10.

Then, we formulate the coreset selection process as a submodular optimization problem, which thus can be solved by a greedy algorithm with an (1-$\frac{1}{e}$) approximation error. Specifically, in each iteration of the algorithm, the data instance that has the highest utility is greedily added to the coreset. The utility measures how much the gradient approximation error would be reduced, if the instance was added into the coreset. Once an instance is added, we update the mapping γ of instances in D based on the d^{SFT} distances between instances of D and C, and update the weights in the coreset. Iteratively, we select K instances as the final coreset.

3.3 Dual–Constraint Coreset Selection

The optimization problem in Eq. (10) is NP-hard, as it requires evaluating all $2^{|D|}$ subsets $C \subseteq D$. However, we show that the aforementioned dual-constraint optimization problem can be reformulated as a submodular set cover problem, which admits efficient approximation algorithms.

Formulation as a Submodular Set Cover Problem. To precisely formulate our coreset selection task as a submodular set cover problem, we define deterministic bounds $\mathcal{B}PT(C)$ and $\mathcal{B}IFL(C)$ for any subset $C \subseteq D$ and parameter $\omega \in \mathcal{W}$.

$$\mathcal{B}_{\text{PT}}(C) \stackrel{\text{def}}{=} \sum_{i=1}^{|D|} \min_{c_j \in C} d_{ij}^{\text{PT}}, \quad \mathcal{B}_{\text{IFL}}(C) \stackrel{\text{def}}{=} \sum_{i=1}^{|D|} \min_{c_j \in C} d_{ij}^{\text{IFL}}. \tag{11}$$

Here, the pairwise distances are used to measure the normed difference between the gradient of data instance $c_i \in D$ and data instance $c_j \in C$. The pairwise distances d_{ij}^{PT} and d_{ij}^{IFL} are seperately defined as:

$$d_{ij}^{\text{PT}} \stackrel{\text{def}}{=} \max_{\theta \in \Theta} \|\nabla\mathcal{L}_{\text{PT}}(y_i;\theta) - \nabla\mathcal{L}_{\text{PT}}(y_j;\theta)\| \tag{12}$$

$$d_{ij}^{\text{IFL}} \stackrel{\text{def}}{=} \max_{\theta \in \Theta} \|\nabla\mathcal{L}_{\text{IFL}}(y_i|x_i;\theta) - \nabla\mathcal{L}_{\text{IFL}}(y_j|x_j;\theta)\| \tag{13}$$

Substituting $\mathcal{B}_{PT}(C)$ and $\mathcal{B}_{IFL}(C)$ into Eq. (10), we obtain a simplified optimization problem with scalar constraints:

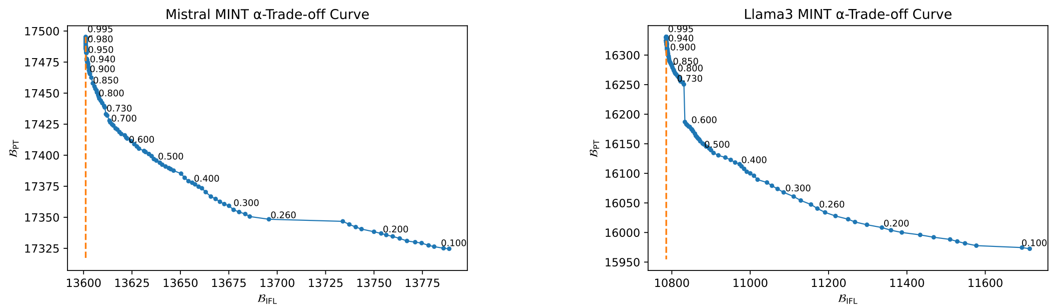
$$\min_{C \subseteq D} |C| \quad \text{s.t.} \quad \mathcal{B}_{\text{PT}}(C) \leq \alpha\varepsilon, \quad \mathcal{B}_{\text{IFL}}(C) \leq (1-\alpha)\varepsilon. \tag{14}$$

These constraints allow us to separately quantify how closely the coreset approximates both knowledge-related capability (PT) and instruction following capability (IFL). To unify these two constraints into a single condition, we introduce a composite distance metric that combines d_{ij}^{PT} and d_{ij}^{IFL}:

$$d_{ij}^{\text{SFT}} \stackrel{\text{def}}{=} \frac{1}{\alpha} d_{ij}^{\text{PT}} + \frac{1}{1-\alpha} d_{ij}^{\text{IFL}} \tag{15}$$

Intuitively, this composite metric simultaneously captures the deviations of the gradients w.r.t. both PT and IFL, where α controls their relative importance. Based on this definition, we demonstrate that satisfying a unified constraint on the composite metric naturally ensures compliance with the individual constraints in Eq. (14). Specifically, for each data, let j_i^* denote the nearest representative in C according to the composite distance d_{ij}^{SFT}. Since both d_{ij}^{PT} and d_{ij}^{IFL} are non-negative real numbers, we analyze each term individually. Clearly, by the definition of the composite metric in Eq. (15), we have:

$$\frac{1}{\alpha} d_{ij_i^*}^{\text{PT}} \leq d_{ij_i^*}^{\text{SFT}}, \quad \frac{1}{1-\alpha} d_{ij_i^*}^{\text{IFL}} \leq d_{ij_i^*}^{\text{SFT}} \tag{16}$$



(a) Mistral

(b) Llama-3

Figure 2: Trade-off curves of Mistral and Llama-3 produced by the `MINT` method.

Multiplying the inequalities above by α and $(1-\alpha)$, respectively, we directly obtain:

$$d^{\mathrm{PT}}_{ij_i^*} \le \alpha\, d^{\mathrm{SFT}}_{ij_i^*}, \quad d^{\mathrm{IFL}}_{ij_i^*} \le (1-\alpha)\, d^{\mathrm{SFT}}_{ij_i^*}. \tag{17}$$

Therefore, summing these inequalities over all points directly obtains:

$$\mathcal{B}_{\mathrm{PT}}(C) = \sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{PT}}_{ij} \le \alpha \sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{SFT}}_{ij}, \mathcal{B}_{\mathrm{IFL}}(C) = \sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{IFL}}_{ij} \le (1-\alpha) \sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{SFT}}_{ij} \tag{18}$$

Consequently, if a coreset $C \subseteq D$ satisfies the unified distance constraint $\sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{SFT}}_{ij} \le \varepsilon$, it automatically meets the two separate budgets outlined in Eq. (14), ensuring both $\mathcal{B}_{\mathrm{PT}}(C) \le \alpha\varepsilon$ and $\mathcal{B}_{\mathrm{IFL}}(C) \le (1-\alpha)\varepsilon$. This allows us to simplify the original dual constraint optimization problem (Eq. (14)) into a single scalar constraint:

$$\min_{C \subseteq D} |C| \quad \text{s.t.} \quad \sum_{i=1}^{|D|} \min_{c_j \in C} d^{\mathrm{SFT}}_{ij} \le \varepsilon, \tag{19}$$

Since d^{SFT}_{ij} is a non-negative linear combination of the distances d^{PT}_{ij} and d^{IFL}_{ij}, it naturally inherits their metric properties. Prior work [31, 32, 39] has established that optimization problems involving such metrics can be equivalently formulated as a submodular set cover problem.

Solving the Unified Constraint via a Greedy Framework. Solving the optimization problem described by Eq. (19) remains NP-hard due to its combinatorial nature. Typical approaches [31, 32, 39] leverage greedy approximation frameworks, widely adopted in existing literature due to their theoretical guarantees arising from submodularity. The solution iteratively builds the coreset by progressively adding items based on their marginal benefit. At each iteration, the algorithm evaluates candidate points not yet included in the coreset, quantifying their potential to minimize the composite distance metric d^{SFT}_{ij}. The selected candidate is the one yielding the greatest improvement, namely the maximum reduction in total gradient approximation error when added to the current coreset.

Evaluating each candidate involves reassessing the impact of adding the candidate point on the approximation error of all other points. This benefit-driven evaluation ensures the gradual optimization of the coreset, producing a theoretically justified approximation solution.

Automated Determination of α. As discussed above, for any arbitrary α, our strategy can select an optimal coreset under this α. Thus, the next significant module is how to choose a good α to balance the two capabilities. To this end, we conduct experiments using one dataset with two different models as illustrative examples. Additional experiments with more datasets are provided in the Appendix. Specifically, for each experimental setting, we first uniformly sample a subset $D_{sub} \subseteq D$ from the original dataset D, and then, for each candidate α, we apply the `MINT` algorithm to obtain the coreset C. Subsequently, we calculate the upper bounds $\mathcal{B}_{\mathrm{IFL}}(C)$ and $\mathcal{B}_{\mathrm{PT}}(C)$, which serve as estimates of the gradient approximation errors $\varepsilon_{\mathrm{ifl}}$ and $\varepsilon_{\mathrm{pt}}$, respectively. Plotting these errors yields an intuitive visualization of how varying α impacts the trade-off.

Table 1: Accuracy on General Task and Domain Task. The **bold** and underlined values represent the first and second best performance, respectively. CSQA, WG, SiQA denotes CommonSenseQA, WinoGrande, SocialiQA respectively. We run each experiment for three times and report the average.

Method	**GPU** $\cdot Hour$	**General Tasks**								**Domain Tasks**		**Overall**
		MMLU	ARC-C	CSQA	WG	LogiQA	PiQA	SiQA	BoolQ	MATH	MBPP	
LLaMA3-8B												
`Random`	0.5	63.6	57.4	73.1	76.1	30.0	81.9	50.4	83.6	40.7	50.4	60.72
`BM25`	0.5	64.1	57.9	73.6	76.7	31.3	82.0	51.3	83.0	39.9	51.6	61.14
`IFD`	1.33	64.7	57.4	72.1	76.1	30.9	81.4	49.8	82.2	39.6	50.6	60.48
`LESS`	20.8	66.0	58.2	73.8	76.7	32.1	81.2	50.8	83.6	41.1	50.4	61.39
`DSIR`	0.5	64.6	57.1	74.1	76.4	30.4	81.9	50.8	82.3	39.9	50.6	60.81
`TAGCOS`	3.15	64.3	58.2	74.3	76.3	30.7	81.9	51.1	82.1	40.5	49.2	60.86
`CRAIG`	3.15	65.1	58.3	74.5	77.3	30.6	81.8	51.0	83.5	41.2	50.8	61.41
`Total`	15	**66.9**	**59.7**	**76.8**	**78.6**	**33.7**	**83.3**	**52.9**	**85.6**	**42.7**	**53.0**	**63.32**
`MINT(Ours)`	5.8	65.9	59.0	76.1	78.3	32.1	82.3	51.9	85.1	42.1	52.3	62.51
Mistral-7B												
`Random`	0.5	56.6	50.4	61.6	69.0	33.0	80.1	47.0	83.8	33.3	41.8	55.66
`BM25`	0.5	58.3	52.5	63.4	73.3	33.8	80.8	47.2	82.9	34.8	41.4	56.84
`IFD`	1.33	58.6	56.4	57.5	72.1	35.6	82.8	47.9	84.1	34.9	41.7	57.16
`LESS`	20.8	57.7	54.5	60.0	73.3	36.4	81.7	47.5	83.8	35.6	41.0	57.15
`DSIR`	0.5	56.6	51.6	55.2	71.6	35.0	81.6	47.8	84.5	33.4	41.6	55.89
`TAGCOS`	3.15	58.4	55.3	63.3	73.2	35.3	83.1	47.4	84.8	36.4	42.8	58.00
`CRAIG`	3.15	**59.3**	55.3	61.1	73.9	35.0	83.2	48.3	84.9	35.8	43.4	58.02
`Total`	15	59.1	55.0	64.6	75.1	37.7	**83.8**	**50.1**	85.3	**36.7**	44.9	**59.23**
`MINT(Ours)`	5.8	58.6	**56.8**	**65.2**	**73.9**	**37.3**	83.2	49.5	**85.6**	36.6	**45.6**	**59.23**

Figure 2 reports the empirical trade-off curves obtained by running the above search procedure on Mistral-7B(Fig. 2a) and Llama-3-8B(Fig. 2b). The vertical dashed line in each subplot marks the point where $\mathcal{B}_{\mathrm{IFL}}(C)$ attains its minimum along the curve. From the experiments, we can observe that despite model-specific differences in scale, both curves exhibit the same qualitative behaviour: as α increases, the instruction following error $\mathcal{B}_{\mathrm{IFL}}(C)$ decreases while the pre-training error $\mathcal{B}_{\mathrm{PT}}(C)$ increases. Moreover, the rate of decrease of $\mathcal{B}_{\mathrm{IFL}}(C)$ quickly slows down, while the increase of $\mathcal{B}_{\mathrm{PT}}(C)$ accelerates significantly. Consequently, beyond a certain value of α, the curve flattens into a plateau. Further increasing α yields almost no additional reduction in $\mathcal{B}_{\mathrm{IFL}}(C)$ but continues to enlarge $\mathcal{B}_{\mathrm{PT}}(C)$.

Therefore, recalling our motivation, we emphasize that during the SFT stage, the instruction following capability is crucial. Hence, when selecting the subset, we prioritize maximizing instruction following capability without significantly sacrificing overall knowledge-related capability. The observed plateau represents an optimal region, as it signifies a state where additional increases in α yield negligible improvement in instruction following yet continue to deteriorate pre-training performance. Consequently, we select the minimal α within this plateau to optimally balance these competing objectives. Formally, let $\mathcal{B}^*_{\mathrm{PT}}(\alpha)$ and $\mathcal{B}^*_{\mathrm{IFL}}(\alpha)$ denote the minimal errors achieved by solving the submodular maximization problem for a given α using the unified distance metric in Eq. (15). We seek an α^* such that:

$$\alpha^* = \arg\max_{\alpha \in (0,1)} \left\{ \alpha \;\middle|\; \left| \frac{d\mathcal{B}^*_{\mathrm{IFL}}(\alpha)}{d\alpha} \right| \leq \delta(\alpha), \frac{d\mathcal{B}^*_{\mathrm{PT}}(\alpha)}{d\alpha} \geq 0 \right\}, \tag{20}$$

where $\delta(\alpha)$ represents the inherent noise in the gradient approximation errors, which can be automatically estimated using statistical approaches such as Gaussian modeling, confidence intervals, or uncertainty quantification methods. Condition $\left| \frac{d\mathcal{B}^*_{\mathrm{IFL}}(\alpha)}{d\alpha} \right| \leq \delta(\alpha)$ identifies the plateau where $\mathcal{B}^*_{\mathrm{IFL}}(\alpha)$ stabilizes, indicating diminishing returns for instruction following. $\frac{d\mathcal{B}^*_{\mathrm{PT}}(\alpha)}{d\alpha} \geq 0$ ensures that the knowledge-related error does not decrease.

4 Experiment

In this section, we use different datasets to fine-tune the base models and conduct sufficient ablation studies to demonstrate the efficiency and effectiveness of `MINT`.

4.1 Experiment Setup

Training Settings. We fine-tune two foundational models, LLAMA-3-8B [13] and Mistral-7B [20], on a single A800 GPU. For all experiments, we train for 4 epochs with a maximum learning rate of 2e-5, using the AdamW optimizer and a linear learning rate scheduler with a 0.03 warmup ratio. The maximum token length during fine-tuning is set to 1024. Following prior works [48, 54], we warm up a model using LoRA [19] on a random 5% subset of the data for fair comparison. Subsequent subset selection is based on this warmed-up model, while training is performed on the base model. To leverage gradients accurately within computational constraints, we adopt the LESS setup, incorporating Adam optimizer gradients in gradient computation and projecting the final gradients to 8192 dimensions.

Training datasets. To evaluate our proposed method, we utilize two publicly available datasets designed for instruction-tuning large language models: `WizardLM` [51] and `Alpaca-GPT4` [38]. The `WizardLM` dataset comprises 70,000 instruction-following examples. It employs an evolutionary approach to instruction generation, ensuring variety and quality in tasks that span simple queries, intricate reasoning, and creative outputs. Similarly, the `Alpaca-GPT4` dataset, provided by vicgalle, contains 52,002 instruction-response pairs generated using GPT-4. Building on the original Alpaca dataset, it emphasizes diverse, high-quality instructions covering coding, reasoning, and general knowledge queries, making it well-suited for enhancing model performance on instruction-driven tasks. To ensure gradients fully reflect the training data's information within computational constraints, we only retain training data with a length of less than 1024 tokens. Due to the space limitation, we put the results of `WizardLM` in the Appendix.

Baselines. We compare `MINT` with several baselines. (1) `Random`. We randomly sample QA pairs from D, which are then used for fine-tuning. (2) `Total`. We fine-tune our model using all the QA pairs in candidate data pool D. (3) `DSIR` [50]. leverages n-gram features to assign weights to candidate training data D, from which C is constructed by sampling according to these estimated weights. (4) `BM25` [40]. Training instances are ranked based on TF-IDF features, measuring their similarity to the validation set. The top-K most relevant examples are then selected from the candidate pool D to construct C. (5) `LESS` [49] selects SFT data instances with the highest influence scores. (6) `IFD` [23] selects QA pairs with the highest IFD scores. (7) `CRAIG` [31] selects the coreset without distinguishing the PT and IFL loss. (8) `TAGCOS` [54] selects data instances by clustering gradients to identify the coreset. (9) `MINT` is our solution.

Evaluation Datasets. To comprehensively evaluate the capabilities of finetuned models, we conduct experiments on various downstream tasks covering the following significant categories. (1) $General\ Tasks$: MMLU [18], ARC-C [8], CommonSenseQA [45], WinoGrande [41], LogiQA [27], PiQA [4], SocialiQA [42] and BoolQ [7]. (2) $Domain\ Tasks$: MATH [2], MBPP [3]. Evaluations are conducted using the lm-evaluation-harness [12] framework and the average accuracy ($i.e.$, Overall Score) is reported for comparison.

4.2 Result

Overall Performance. In Table 1, we selected 5% data instances for each baseline. We can observe that `MINT` surpasses all the baseline methods on accuracy across all models and downstream tasks. Specifically, when implementing fine-tuning on Llama-3.1-8B, `MINT` achieves an accuracy improvement of 1% in the average score compared with `LESS`, while saving approximately $2\times$ GPU computation, $w.r.t.$ the computational cost. `MINT` surpasses `LESS` due to the fact that the data instances selected by `LESS` are quite similar to the ones in reference dataset $\mathcal{D}_r$, resulting in limited data diversity. `MINT` outperforms `IFD` because the IFD score is solely based on the inherent instruction following properties of individual data instances, without taking into account their distribution within the overall dataset. Also, `MINT` outperforms `CRAIG` and `TAGCOS` because they select coreset data solely based on the overall gradient and pays limited attention to how individual training examples influence the instruction following ability of LLMs. In terms of the computational cost, we can observe that the FLOPs consumed by the `Total` method is notably high. This is due to their necessity of training with all data in D, which incurs prohibitively expensive cost.

Table 1 also presents a comparison of all the baselines across different data selection ratios. Interestingly, we find that the selection of just 5% of data instances for most tasks produces superior results compared to the use of complete D. This demonstrates the effectiveness of `MINT`. Even for the difficult task MMLU, selecting only 15% of the data instances in D can achieve comparable performance to `Total` across all the training settings.

4.3 Ablation Study

In this section, we demonstrate the impact of hyperparameters on experimental results, specifically focusing on the subset size and the capacity balance parameter α.

[*Subset Ratio.*] We perform an ablation study on the subset ratio to evaluate its impact on the performance of our method. Specifically, we experiment with subset ratios of 5%, 10%, and 15%. Additionally, we report results for a random sampling baseline and training with the entire dataset under the same subset size for comparison. The performance comparisons are illustrated in Table 2. Overall, we observe that the evaluation scores consistently improve as the subset ratio increases, both for the random baseline and our proposed method. Notably, training on the entire dataset achieves the best performance. However, our method consistently outperforms random sampling under the same subset ratios. Furthermore, when the subset ratio reaches 15%, the performance of our method closely approaches that of training on the full dataset, demonstrating its effectiveness in selecting highly representative data.

[*Balancing parameter* α] We conduct an extensive analysis on the balancing parameter α used within our method to validate the effectiveness of our proposed Automated Determination of α approach. Specifically, experiments were performed on the Llama3-8B model, varying α to observe its impact on model performance. Utilizing our noise-based plateau detection strategy, we identify an optimal value of $\alpha = 0.95$. The corresponding experimental results are illustrated in Table 3. Overall, we observe that the selected hyperparameter ($\alpha = 0.95$) demonstrates superior average performance. Scores decline on either side of $\alpha = 0.95$, reinforcing the general effectiveness of our method. More specifically, each evaluation dataset exhibited at least one fluctuation in performance scores as α varied, confirming the effectiveness and sensitivity of our balancing parameter.

5 Related Work

Nowadays, research on data selection in the instruction tuning stage typically focuses on filtering low-quality data and selecting examples that benefit target domains.

Filtering Low Quality Data. Initially, researchers often designed hand-crafted heuristics [44, 37], to filter low-quality data. Deduplication is another typical technique to select pretraining data, such as [37] and SemDedup [1] which use keyword-based and semantic deduplication, respectively. Although these methods effectively filter out noise and redundant data from noisy data sources such as the web, they rely on simple heuristics and cannot be well generalized. In addition, researchers also leverage high-performance models ($e.g.$, GPT-4) to select high-quality data. Although large models can effectively assess data quality due to their semantic comprehension capacity, the metrics utilized to rate data ($e.g.$, writing style, educational value etc.) heavily rely on human intuition [47, 56, 15]. Moreover, perplexity also serves as a metric for selecting high-probability data in a language model. In [5, 29, 33, 46], perplexity (PPL) is utilized to filter data. However, as also noted in Qurating [47], we observe that this method often incorporates a significant amount of simple and redundant data, because they are easy for the model to predict.

Selecting Domain-related Data. To meet users' specific needs or domain requirements, many methods select data with distributions similar to the downstream application for instruction tuning. For instance, certain approaches [11, 50] employ n-gram similarity to assist in choosing corpora that is semantically aligned with the validation set. [14, 6] demonstrate that influence function can reveal the impact of training data on large model performance for specific tasks. Consequently, LESS [49] and MATES [53] utilize influence function to select data during the SFT and pretraining phases. To improve the generalization of data selection method, many researchers train a surrogate model to measure the relevance of each data point to the downstream application. DeepSeekMath [43] proposes an active learning strategy to train a web data classifier. Similarly, in MATES [53], a surrogate model was developed to estimate the influence scores of the data instances. RHO-1 [26] used a surrogate model trained with high-quality data to perform token-level data filtering. However, these techniques usually require significant GPU resources for training surrogate model, and classifiers tend to be domain-specific, limiting their adaptability across various domains.

References

[1] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. arXiv preprint arXiv:2303.09540, 2023.

[2] A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019.

[3] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, et al. Program synthesis with large language models. arXiv preprint arXiv:2108.07732, 2021.

[4] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical commonsense in natural language. In Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.

[5] J. Chen, Z. Chen, J. Wang, K. Zhou, Y. Zhu, J. Jiang, Y. Min, W. X. Zhao, Z. Dou, J. Mao, et al. Towards effective and efficient continual pre-training of large language models. arXiv preprint arXiv:2407.18743, 2024.

[6] S. K. Choe, H. Ahn, J. Bae, K. Zhao, M. Kang, Y. Chung, A. Pratapa, W. Neiswanger, E. Strubell, T. Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. arXiv preprint arXiv:2405.13954, 2024.

[7] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In NAACL, 2019.

[8] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. ArXiv, abs/1803.05457, 2018.

[9] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

[10] D. Feldman. Core-sets: Updated survey. Sampling techniques for supervised or unsupervised tasks, pages 23–44, 2020.

[11] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027, 2020.

[12] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

[13] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

[14] R. Grosse, J. Bae, C. Anil, N. Elhage, A. Tamkin, A. Tajdini, B. Steiner, D. Li, E. Durmus, E. Perez, et al. Studying large language model generalization with influence functions. arXiv preprint arXiv:2308.03296, 2023.

[15] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, et al. Textbooks are all you need. arXiv preprint arXiv:2306.11644, 2023.

[16] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[17] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. Authorea Preprints, 2023.

[18] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. Proceedings of the International Conference on Learning Representations (ICLR), 2021.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. ICLR, 1(2):3, 2022.

[20] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[21] C. Lee, J. Han, S. Ye, S. J. Choi, H. Lee, and K. Bae. Instruction matters: A simple yet effective task selection for optimized instruction tuning of specific tasks. arXiv preprint arXiv:2404.16418, 2024.

[22] M. Li, L. Chen, J. Chen, S. He, J. Gu, and T. Zhou. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In L.-W. Ku, A. Martins, and V. Srikumar, editors, Findings of the Association for Computational Linguistics ACL 2024, pages 16189–16211, Bangkok, Thailand and virtual meeting, Aug. 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-acl.958`.

[23] M. Li, Y. Zhang, S. He, Z. Li, H. Zhao, J. Wang, N. Cheng, and T. Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. arXiv preprint arXiv:2402.00530, 2024.

[24] M. Li, Y. Zhang, Z. Li, J. Chen, L. Chen, N. Cheng, J. Wang, T. Zhou, and J. Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In K. Duh, H. Gomez, and S. Bethard, editors, Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7595–7628, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.naacl-long.421`.

[25] Y. Li, B. Hui, X. Xia, J. Yang, M. Yang, L. Zhang, S. Si, L.-H. Chen, J. Liu, T. Liu, et al. One-shot learning as instruction data prospector for large language models. arXiv preprint arXiv:2312.10302, 2023.

[26] Z. Lin, Z. Gou, Y. Gong, X. Liu, Y. Shen, R. Xu, C. Lin, Y. Yang, J. Jiao, N. Duan, et al. Rho-1: Not all tokens are what you need. arXiv preprint arXiv:2404.07965, 2024.

[27] J. Liu, L. Cui, H. Liu, D. Huang, Y. Wang, and Y. Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning, 2020.

[28] W. Liu, W. Zeng, K. He, Y. Jiang, and J. He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. arXiv preprint arXiv:2312.15685, 2023.

[29] M. Marion, A. Üstün, L. Pozzobon, A. Wang, M. Fadaee, and S. Hooker. When less is more: Investigating data pruning for pretraining llms at scale. arXiv preprint arXiv:2309.04564, 2023.

[30] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey. arXiv preprint arXiv:2402.06196, 2024.

[31] B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In International Conference on Machine Learning, pages 6950–6960. PMLR, 2020.

[32] B. Mirzasoleiman, K. Cao, and J. Leskovec. Coresets for robust training of deep neural networks against noisy labels. Advances in Neural Information Processing Systems, 33:11465–11477, 2020.

[33] N. Muennighoff, A. Rush, B. Barak, T. Le Scao, N. Tazi, A. Piktus, S. Pyysalo, T. Wolf, and C. A. Raffel. Scaling data-constrained language models. Advances in Neural Information Processing Systems, 36, 2024.

[34] A. Munteanu and C. Schwiegelshohn. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. KI-Künstliche Intelligenz, 32:37–53, 2018.

[35] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian. A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435, 2023.

[36] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744, 2022.

[37] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. arXiv preprint arXiv:2306.01116, 2023.

[38] B. Peng, C. Li, P. He, M. Galley, and J. Gao. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277, 2023.

[39] O. Pooladzandi, D. Davini, and B. Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In International Conference on Machine Learning, pages 17848–17869. PMLR, 2022.

[40] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389, 2009.

[41] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. arXiv preprint arXiv:1907.10641, 2019.

[42] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi. Social iqa: Commonsense reasoning about social interactions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4463–4473, 2019.

[43] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[44] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. arXiv preprint arXiv:2402.00159, 2024.

[45] A. Talmor, J. Herzig, N. Lourie, and J. Berant. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL `https://aclanthology.org/N19-1421`.

[46] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. arXiv preprint arXiv:1911.00359, 2019.

[47] A. Wettig, A. Gupta, S. Malik, and D. Chen. Qurating: Selecting high-quality data for training language models. In Forty-first International Conference on Machine Learning, 2024. URL `https://openreview.net/forum?id=GLGYYqPwjy`.

[48] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen. Less: Selecting influential data for targeted instruction tuning. arXiv preprint arXiv:2402.04333, 2024.

[49] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen. Less: Selecting influential data for targeted instruction tuning. In Forty-first International Conference on Machine Learning, 2024.

[50] S. M. Xie, S. Santurkar, T. Ma, and P. S. Liang. Data selection for language models via importance resampling. Advances in Neural Information Processing Systems, 36:34201–34227, 2023.

[51] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, Q. Lin, and D. Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In The Twelfth International Conference on Learning Representations, 2024. URL `https://openreview.net/forum?id=CfXh93NDgH`.

[52] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[53] Z. Yu, S. Das, and C. Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models. arXiv preprint arXiv:2406.06046, 2024.

[54] J. Zhang, Y. Qin, R. Pi, W. Zhang, R. Pan, and T. Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. arXiv preprint arXiv:2407.15235, 2024.

[55] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, et al. Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792, 2023.

[56] Y. Zhang, Y. Luo, Y. Yuan, and A. C. Yao. Autonomous data selection with language models for mathematical texts. In ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models, 2024.

[57] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023.

A Ablation Study Result

Table 2: Ablation Study of α.

α	10%	25%	50%	63%	75%	90%	95%	99.9%
Accuracy	59.97	60.55	61.41	61.77	62.15	62.43	62.51	62.51

Table 3: Ablation Study of Coreset Size

Method	Subset Ratio	General Task								Domain Task		Overall
		MMLU	ARC-C	CSQA	WG	LogiQA	PiQA	SiQA	BoolQ	MATH	MBPP	
LLaMA3-8B												
`Random`	5%	63.6	57.4	73.1	76.1	30.0	81.9	50.4	83.6	40.7	50.4	60.7
	10%	63.8	57.7	73.4	76.3	30.3	82.3	51.4	83.5	40.9	51.0	61.1
	15%	64.7	58.3	74.3	76.9	31.7	82.8	52.2	83.7	41.3	51.7	61.8
	100%	66.9	59.7	76.8	78.6	33.7	83.3	52.9	85.6	42.7	53.0	63.3
`MINT(Ours)`	5%	65.9	59.0	76.1	78.3	32.1	82.3	51.9	85.1	42.1	52.3	62.5
	10%	65.9	57.2	73.8	75.5	30.4	82.5	51.2	85.0	42.3	52.0	61.6
	15%	66.7	59.8	77.6	78.8	33.5	83.3	51.9	85.3	42.3	52.6	63.2

B Algorithm pseudocode

Algorithm 1: `MINT` with Pre-computed Distances

Input: Candidate pool $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$; extraction ratio γ $(0 < \gamma \leq 1)$; pre-trained parameters ω

Output: Coreset $C \subseteq \mathcal{D}$, weights W

Function `MINT_CORESET`$(\mathcal{D}, \gamma, \alpha, d_{ij}^{\mathrm{PT}}, d_{ij}^{\mathrm{IFL}})$**:**

  $k \leftarrow \lceil \gamma\,|\mathcal{D}| \rceil$; $\quad C \leftarrow \emptyset$;

  foreach $i, j \in [1, |\mathcal{D}|]$ **do**

    $d_{ij}^{\alpha} \leftarrow \dfrac{d_{ij}^{\mathrm{PT}}}{\alpha} + \dfrac{d_{ij}^{\mathrm{IFL}}}{1-\alpha}$;

  $F(C) \leftarrow$ `FacilityLocation`(d_{ij});

  while $|C| < k$ **do**

    foreach $u \in \mathcal{D} \setminus C$ **do**

      $\Delta(u) \leftarrow F(C \cup \{u\}) - F(C)$;

    $u^{\star} \leftarrow \arg\max_u \Delta(u)$; $\quad C \leftarrow C \cup \{u^{\star}\}$;

  initialise $\lambda_j \leftarrow 0, \ \forall j \in C$;

  foreach $i \in [1, |\mathcal{D}|]$ **do**

    $j^{\star} \leftarrow \arg\min_{j \in C} d_{ij}$;

    $\lambda_{j^{\star}} \leftarrow \lambda_{j^{\star}} + 1$;

  return $C, \ W = \{\lambda_j\}_{j \in C}$;

foreach $i \in [1, |\mathcal{D}|]$ **do**

  $g_i^{\mathrm{PT}} \leftarrow \nabla_{\omega} \mathcal{L}_{\mathrm{PT}}(y_i)$; $\quad g_i^{\mathrm{IFL}} \leftarrow \nabla_{\omega} \mathcal{L}_{\mathrm{IFL}}(y_i \mid x_i)$;

foreach $i, j \in [1, |\mathcal{D}|]$ **do**

  $d_{ij}^{\mathrm{PT}} \leftarrow \|g_i^{\mathrm{PT}} - g_j^{\mathrm{PT}}\|$; $\quad d_{ij}^{\mathrm{IFL}} \leftarrow \|g_i^{\mathrm{IFL}} - g_j^{\mathrm{IFL}}\|$;

$\mathcal{D}', d_{ij}^{\mathrm{PT}'}, d_{ij}^{\mathrm{IFL}'} \leftarrow$ `SubsampleForAlphaSearch`$(\mathcal{D}, d_{ij}^{\mathrm{PT}}, d_{ij}^{\mathrm{IFL}})$;

$\alpha^{\star} \leftarrow$ `NoiseAwarePlateauDetected`$(\mathcal{D}', \gamma, \omega, d_{ij}^{\mathrm{PT}}, d_{ij}^{\mathrm{IFL}},$ `MINT_CORESET`$)$;

return `MINT_CORESET`$(\mathcal{D}, \gamma, \alpha^{\star}, d_{ij}^{\mathrm{PT}}, d_{ij}^{\mathrm{IFL}})$;

Broader Impact

By improving data efficiency during supervised fine-tuning (SFT), our proposed framework `MINT` can substantially reduce the computational resources and time required to adapt large models to general instruction-following tasks. This reduction in resource demands is particularly beneficial for smaller organizations and research groups with limited computational budgets, thereby democratizing access to advanced LLM capabilities and fostering wider participation in AI development. Furthermore, by carefully balancing the retention of pre-trained knowledge with the enhancement of instruction-following ability, our method promotes the development of more reliable and robust models.

On the societal level, increased efficiency and accessibility of LLM fine-tuning can accelerate the deployment of AI-powered tools in education, healthcare, and other domains, enabling more personalized and context-aware applications. However, as with all AI systems, there remain concerns regarding potential biases and misuse. By advocating for transparent, theoretically grounded data selection methods, we encourage responsible development practices that prioritize model integrity and fairness. Ultimately, our work contributes to making large-scale language model fine-tuning more sustainable, equitable, and aligned with practical deployment needs, advancing the broader goals of ethical and inclusive AI.

Limitations

`MINT` assumes that the pre-trained LLM has already acquired robust real-world knowledge. If the base model's knowledge is incomplete or biased, the coreset selected to focus on instruction-following might not fully compensate for these deficiencies, potentially limiting the performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We believe the claims in our abstract and introduction accurately reflect the paper's contributions, including theoretical analysis and experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We've discussed the limitations in page 10.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We've provided clear descriptions and equations in the theoretical part, especially in Sec 3 and Supplementary Material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We've provided detailed information about the experiment setting in Sec 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We've provided our complete code in anonymous link mentioned in abstract. The datasets used in our paper are all open-source datasets which are easy to downloaNIPSd.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We've provided detailed information about the experiment setting in 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We've mentioned in Table 1 that all experiments are conducted 5 times with random seed to confirm the statistical significance. The average results are reported in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We've mentioned in Sec 4.1 that all experiments are conducted on a single Nvidia A100 80GB GPU.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm our paper conform with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [Yes]

   Justification: We've discussed the broader impact in Page 10.

   Guidelines:

   - The answer NA means that there is no societal impact of the work performed.
   - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
   - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Creators or original owners of assets (e.g., code, data, models), used in the paper are all properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.