MIG: Automatic Data Selection for Instruction Tuning by Maximizing Information Gain in Semantic Space

Anonymous ACL submission

Abstract

Data quality and diversity are pivotal in constructing effective instruction-tuning datasets. With the increasing availability of open-source instruction-tuning datasets, it is advantageous to automatically select high-quality and diverse subsets from a vast amount of data. Existing methods typically prioritize instance quality and use heuristic rules to maintain diversity. However, this absence of a comprehensive view of the entire collection often leads to suboptimal results. Moreover, heuristic rules generally focus on distance or clustering within the embedding space, which fails to capture the intent of complex instructions in semantic space accurately. To bridge this gap, we propose a unified dataset information measurement method. This method models the semantic space by constructing a label graph and quantifies diversity based on the distribution of information within the graph. Based on such measurement, we further introduce an efficient sampling method that selects data samples iteratively to **M**aximize the **I**nformation **G**ain (MIG) in semantic space. Experiments on various datasets and base models demonstrate that MIG consistently outperforms state-of-the-art methods. Notably, the model fine-tuned with 5% Tulu3 data sampled by MIG achieves comparable performance to the official SFT model trained on the full dataset, with improvements of +5.73% on AlpacaEval and +6.89% on Wildbench. This finding shows the potential for unified dataset measurement in guiding instruction data selection. Code will be available.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in following human instructions in a wide range of tasks (Wang et al., 2023). Through large-scale pretraining, these models acquire general knowledge and are subsequently refined by instructing (Brown et al., 2020; Taori et al., 2023; Chiang et al., 2023; Touvron et al., 2023) to better align with diverse human intentions (Zhou et al., 2023a). Instruct-tuning leverages instruction-response pairs to supervise base models to respond to human instructions accurately and contextually appropriately. Recent research (Zhou et al., 2023a; Chen et al., 2024) underscores the importance of data quality over quantity for good instruct-tuning. In particular, LIMA (Zhou et al., 2023a) shows that a dataset of only 1000 human-curated instructions can achieve performance comparable to much larger datasets. However, initial efforts are typically based on manually curated high-quality instruction datasets, which are time-consuming and labor intensive (Chiang et al., 2023).

More recently, a line of work (Chen et al., 2024; Liu et al., 2024b) proposes methods to automatically identify an optimal subset from a large data pool by defining and selecting data points with desirable properties. These methods (Bukharin et al., 2024; Yu et al., 2024b) posit that **quality** and **diversity** are essential attributes for an effective instruct-tuning dataset. They define these attributes from various perspectives and iteratively select samples that best fulfill such defined criteria. Regarding quality, previous studies propose measures based on instruction complexity (Lu et al., 2024; Zhao et al., 2024), model perplexity and uncertainty (Li et al., 2024b), or scores assigned by powerful external models (Chen et al., 2024; Liu et al., 2024b). However, most existing methods lack measurement of diversity and only maintain it through heuristic rules, such as maximizing the coverage of label set (Lu et al., 2024), applying diversity filters to mitigate redundancy (Liu et al., 2024b), or enforcing a fixed number of samples per cluster (Ge et al., 2024; Yu et al., 2024b). Some approaches (Bukharin et al., 2024) quantify diversity using submodular functions. However, they require computationally expensive iterative pairwise embedding similarity calculations, making them

inefficient on large data pools. To solve these issues, several essential questions are raised: 1) How can we unify quality and diversity in dataset measurement? 2) How do we efficiently select samples based on such an evaluation?

To this end, we propose an information-based measurement for instruction-tuning datasets and a corresponding efficient sampling algorithm that aims to **M**aximize the **I**nformation **G**ain (MIG). The information within a dataset is distributed over a set of semantic labels, and the total information is the sum of the information associated with each label. Each data point contributes to the information of its associated labels, the contribution being proportional to its quality. To ensure a label-balanced data distribution, the information for each label is computed using a marginal diminishing function. Additionally, we account for semantic correlations between labels by propagating information along the edge of the label graph to capture the information distribution more accurately. To efficiently select a dataset that maximizes the total information, we employ a greedy strategy that iteratively selects data points that maximize the information gain according to the current state of the label graph.

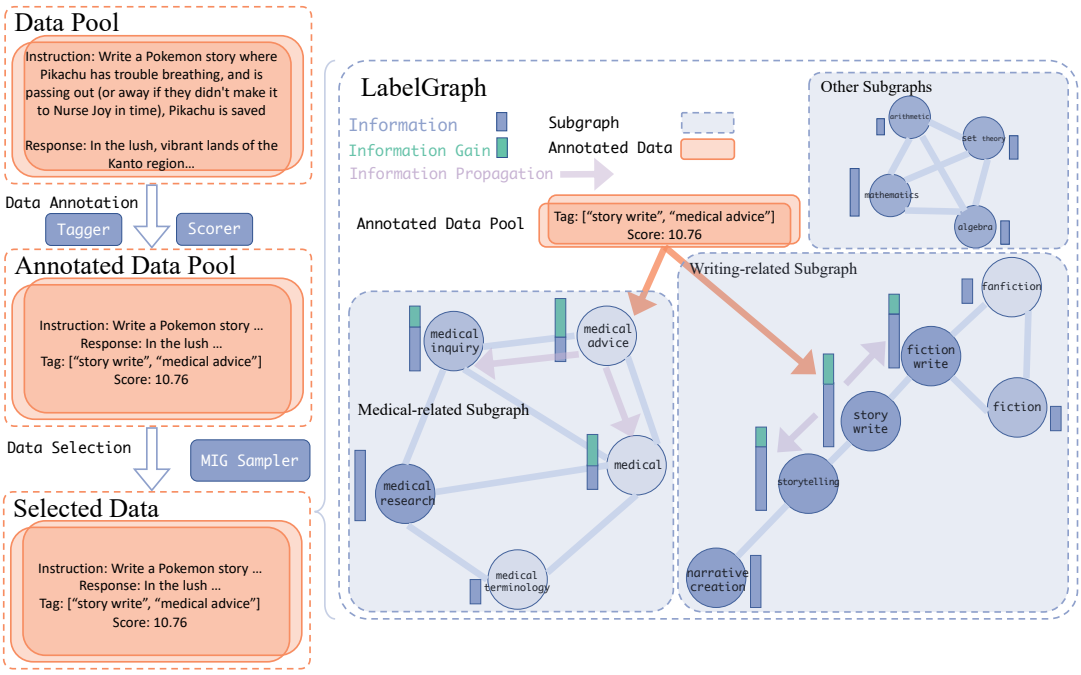
Through extensive experiments with data pools of varying sizes and LLMs of different families, namely Llama (Touvron et al., 2023) and Mistral (Jiang et al., 2023), MIG demonstrates its effectiveness in both human-preference and knowledge-based evaluations. Notably, on the Tulu3 (Lambert et al., 2024) pool, MIG achieves average improvements of **+1.49%** on six knowledge-based benchmarks (Clark et al., 2018; Suzgun et al., 2022; Hendrycks et al., 2021; Chen et al., 2021; Cobbe et al., 2021; Zhou et al., 2023b) and **+1.96%** on three human-preference benchmarks (Zheng et al., 2023; Dong et al., 2024; Lin et al., 2024) compared to previous state-of-the-art data selection methods (Liu et al., 2024b; Bukharin et al., 2024). When combining both evaluations, MIG achieves average improvements of **+2.20%** compared to the second-best method (Bukharin et al., 2024), and even outperforms the model trained on the full Tulu3 data by **+1.73%**, with a substantial boost in human-preference based evaluations (**+4.59%** on average). MIG also outperforms other methods on the Openhermes2.5 (Teknium, 2023) and X_{sota} (Lu et al., 2024; Liu et al., 2024b), further demonstrating its generalizability across different settings. Additionally, MIG shows significant efficiency, making it particularly suited for sampling on large-scale data pools.

In summary, our contributions are as follows:

- We design an instruction-tuning dataset information measurement via label graph. It comprehensively evaluates dataset quality and diversity.
- We propose an efficient and effective sampling algorithm, MIG, to select samples that maximize the information gain on the label graph iteratively.
- Extensive experiments on various data pools and models demonstrate that MIG enhances the quality and diversity of sampled data and improves model performance on comprehensive human-preference and knowledge-based benchmarks. The correlation between parameters in MIG and the attributes of sampled data is well studied.

2 Related Work

Data Selection for Instruction Tuning. Instruction-tuning data can significantly enhance base LLMs. Increasing data quality and diversity rather than quantity has been shown to more effectively induce instruction-following abilities. Consequently, data selection strategies aim to identify the most optimal data subsets. These methodologies generally fall into two categories: (1) **Quality-based approaches** prioritize high-quality data points, where high quality is defined through various perspectives, such as instruction complexity and response quality. INSTRUCT-MINING (Cao et al., 2024) identifies key natural language metrics as indicators for high-quality instruction data. Instruction-Following Difficulty (IFD) (Li et al., 2024b) highlights inconsistencies between a model's anticipated responses and its self-generated outputs. Nuggets (Li et al., 2023b) measures quality based on the disparity between one-shot and zero-shot performance. LESS (Xia et al., 2024a) uses gradient features to select samples based on their similarity to a few representative examples. SelectIT (Liu et al., 2024a) selects high-quality data based on the intrinsic uncertainty reflected by LLMs from token, sentence, and model levels. Additionally, some methods employ external LLMs to assess data quality, such as ALPAGASUS (Chen et al., 2024), which uses a well-designed prompt applied to ChatGPT to assess the quality of each data tuple. (2) **Diversity-based approaches** aim to select data subsets with board coverage of the data pool. DiverseEvol (Wu et al., 2023) maintains high diversity within selected subsets by progressively



(a) Data Selection Pipeline (b) MIG Sampler

Figure 1: Illustration of (a) Data Selection Pipeline and (b) MIG Sampler. Given the raw data pool, our pipeline first applies a tagger and scorer to annotate the data. Next, MIG constructs the label graph based on the tag set and iteratively selects the data point that maximizes the information gain, considering the current state of the label graph. The selected data are then used for supervised fine-tuning (SFT) training of LLMs.

choosing data points that are distant from existing ones in the current embedding space of the model. ZIP (Yin et al., 2024) prioritizes subsets with low compression ratios. **(3) Comprehensive approaches** aim to balance both quality and diversity. *#InsTag* (Lu et al., 2024) employs ChatGPT to generate detailed open-ended tags for instructions and prioritize complex data with more tags while maximizing topic coverage. DEITA (Liu et al., 2024b) prioritizes high-quality data points while avoiding duplicates in the embedding space. CaR (Ge et al., 2024) and kMQ (Yu et al., 2024b) clusters and samples from each cluster according to quality. However, these methods lack a comprehensive and unified measure for subsets and rely on heuristic rules to balance quality and diversity.

Submodular function for Diversity Measurement. Submodular functions are effective for modeling information in subsets. Maximizing a submodular function, such as facility location, graph cut, or log determinant, is equivalent to identifying non-redundant subsets. Leveraging this property, QDIT (Bukharin et al., 2024) measures diversity using the facility location function (Cornuéjols et al., 1983), combining it linearly with quality scores. Similarly, DPP (Wang et al., 2024) employs the Log Determinant Distance to quantify subset diversity and introduces a hyperparameter to control the relative importance of diversity and quality. Although this NP-hard problem can be approximated with a greedy algorithm (Nemhauser et al., 1978; Minoux, 2005), real-world SFT scenarios present significant challenges due to the high storage and computational costs of calculating pairwise instance distances in the embedding space.

3 Method

3.1 Preliminary

Task. Given a data pool D_P, a budget N, and an information measure $E(D)$ over any dataset D, the goal is to select a subset $D_S \subset D_P$ of size N that maximizes $E(D)$. Formally,

$$D_S = \underset{D \subset D_P, |D|=N}{\operatorname{argmax}} E(D) \quad (1)$$

Data. Each data point is formed as:

$$d_i = \{(q_i^j, r_i^j)_{j=1}^M, L_i, s_i\} \quad (2)$$

where $(q_i^j, r_i^j)_{j=1}^M$ represents M rounds of query-response pairs used for training, L_i is the set of labels (e.g., task category, knowledge domain, and

other meta information) associated with d_i, and s_i is the quality score.

3.2 Information Measurement

Label Graph. Previous studies (Lu et al., 2024; Ge et al., 2024; Yu et al., 2024b) assume that data labels (including embedding-based clusters) are independent, ignoring the semantic relationships among them. However, such label associations are crucial for label-balance sampling. Intuitively, we can model labels as nodes, their associations as edges, and the intensity of associations as edge weights, thus forming an undirected weighted graph $G_L = (L, E_L)$, where L represents the label set with a size of K and E_L represents edges. Specifically, we use the similarity in the embedding space as edge weights and remove edges whose weights are below a threshold T to ensure computational efficiency. E_L can be formed as a weighted adjacency matrix $W_L \in \mathbb{R}^{K\times K}$ with elements:

$$w_{pq} = \sigma[w(l_p, l_q) \geq T] \cdot w(l_p, l_q) \tag{3}$$

where $w(l_p, l_q)$ represents similarity between label l_p and l_q, and $\sigma(\cdot)$ returns 1 when the input is True.

Data Point Information. We first define the information contributed by a single data point and then generalize it to the entire dataset. Under the label set L, a data point d_i can be formed as a binary label vector with its associated labels L_i:

$$\mathbf{v}_i = \{v_k^i = \sigma(l_k \in L_i)\}_{k=1}^{K} \tag{4}$$

The information of d_i is distributed over L_i and is proportional to its quality score s_i. Thus, the raw information of d_i can be formed as:

$$E_i = s_i \cdot \mathbf{v}_i \tag{5}$$

Beyond directly contributing to the information of L_i, a data point also influences its neighboring labels through a propagation process along the edges of the label graph. Formally, the propagation from l_p to l_q is:

$$a_{pq} = \frac{\alpha w_{pq}}{w_p + \alpha \sum_{k,k\neq p} w_{pk}} \tag{6}$$

where w_p equals 1 and α is a hyper-parameter controlling the intensity of information propagation. Let A be the propagation matrix, then the propagated information vector of d_i is:

$$\hat{E}_i = AE_i \tag{7}$$

Dataset Information. To promote a diverse distribution of labels within the label graph, we apply a monotonically increasing yet upper-convex function ϕ to compute the label information instead of a simple summation. The diminishing marginal information gain is negatively correlated with the existing information on a label. Thus, information gains on labels with less information are prioritized. Formally, the information of the dataset is defined as:

$$E(D) = \Phi(\sum_{i\in D} AE_i) = \Phi(A\sum_{i\in D} s_i\mathbf{v}_i) \tag{8}$$

where Φ is an element-wise operation that applies ϕ on each vector element.

3.3 MIG Sampling

Directly selecting D_S from D_P is computationally infeasible as the combination $C^N_{|D_P|}$ grows quickly. Therefore, we propose a greedy strategy, iteratively selecting the data point that yields the maximum information gain:

$$d_k = \underset{d\in D_P^k}{\text{argmax}}\{E(D_S^k \cup \{d\}) - E(D_S^k)\} \tag{9}$$

where D_S^k and D_P^k denote the selected subset and remaining candidate pool at iteration k. We approximate the information gain in Eq. 9 via a gradient-based approach:

$$G_k = \frac{\partial E(D_S^k)}{\partial E} = A\Phi'(A\sum_{i\in D_S^k} E_i) \tag{10}$$

where Φ' represents the derivative of Φ. By iteratively selecting points that maximize the incremental gain:

$$d_k = \underset{d\in D_P^k}{\text{argmax}}\, G_k E_d \tag{11}$$

we efficiently sample a subset D_S that balances both label diversity and data quality. The whole sampling process is detailed in Alg. 1.

3.4 Data Selection Pipeline

As illustrated in Fig. 1, starting from the raw data pool, we first utilize a tagger and scorer to annotate the data. Then, we perform tag normalization following (Lu et al., 2024), which includes frequency filtering and semantic aggregation, resulting in the label set L. Next, we compute the similarity between labels to construct the label graph and the propagation matrix. MIG then performs iterative sampling to obtain the final subset.

Algorithm 1: MIG Sampling

Data: Initial Data Pool D_P, Label Sets L, Sample Budget N

Result: The Sampled Dataset D_S

Initialize Empty D_S;
Initialize Propagation Matrix A;
while $|D_S| < N$ **do**
  $G \leftarrow \Phi'(A \sum_{k \in D_S} E_k)$;
  $d_i \leftarrow \text{argmax}_{d_i \in D_P} GE_i$;
  $D_S \leftarrow D_S \cup \{d_i\}$;
  $D_P \leftarrow D_P \setminus \{d_i\}$;
return D_S

4 Experiments

4.1 Setups

Datasets. To investigate data selection across various scenarios and demonstrate the robustness of MIG, we use three distinct data pools:

• Tulu3 (Lambert et al., 2024): A large-scale, real-world SFT dataset presented by Ai2, containing a million-level records across a wide variety of subjects, including mathematics, programming, and user dialogues.

• Openhermes2.5 (Teknium, 2023): With over 1 million data, sourced from 16 distinct origins, including MetaMath (Yu et al., 2024a), CamelAI (Li et al., 2023a), and others.

• X_{sota}, a combined data pool following (Lu et al., 2024; Liu et al., 2024b): A combined data pool consisting primarily of high-quality conversations from datasets such as WizardLM (Alpaca), WizardLM (ShareGPT), UltraChat (Ding et al., 2023), and ShareGPT (Chiang et al., 2023), totaling 300K data points.

Benchmarks. We use both human-preference and knowledge-based benchmarks to evaluate alignment performance comprehensively.

• Human-preference Benchmarks: MT-bench (Zheng et al., 2023) and AlpacaEval (Dubois et al., 2024) and Wildbench (Lin et al., 2024), which features challenging, real-world user queries.

• Knowledge-based Benchmarks: We evaluate across six tasks. For natural language reasoning, we use ARC (Clark et al., 2018) and Big-Bench-Hard(BBH) (Suzgun et al., 2022). For world knowledge, we evaluate using MMLU (Hendrycks et al., 2021), a dataset of multiple-choice academic questions. For code generation, we utilize the HumanEval (Chen et al., 2021) benchmark, consisting of 164 coding problems, to assess LLMs' code-writing abilities. For mathematical reasoning, we use GSM8k (Cobbe et al., 2021), which includes 1319 grade school math problems. For instruction-following evaluation, we employ IFEval (Zhou et al., 2023b).

Baselines. We compare our methods against six strong data selection approaches: IFD (Li et al., 2024b), ZIP (Yin et al., 2024), *#InsTag* (Lu et al., 2024), DEITA (Liu et al., 2024b), CaR (Ge et al., 2024), and QDIT (Bukharin et al., 2024). Additionally, random selection (Xia et al., 2024b) is also considered a strong baseline, especially for comprehensive knowledge-based evaluations.

Implementation Details. For the Tulu3 data pool, we conduct a grid search to determine the data bucket size of 50K, with training for three epochs. For Openhermes2.5, we follow the settings in (Xia et al., 2024b), sampling 50K data points and training for three epochs. For X_{sota}, we sample 6K data points and train for six epochs following (Lu et al., 2024; Liu et al., 2024b).

We use the widely adopted LLaMA3.1-8B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023) as our base models and fine-tune them using the Llama-Factory framework (Zheng et al., 2024). Please refer to Appx. A.1 for detailed training and evaluation setup.

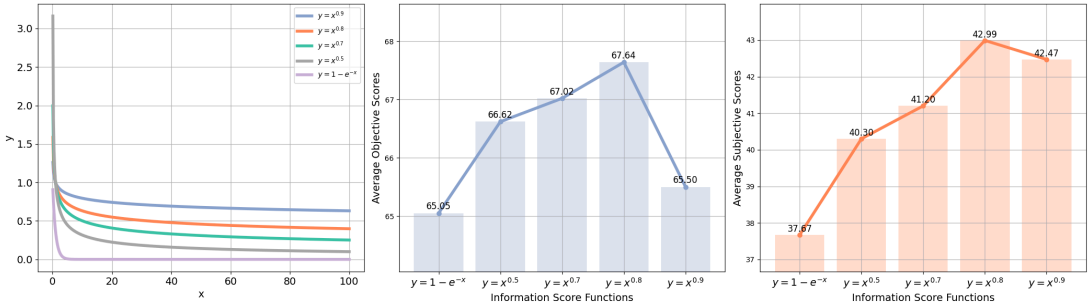
To replicate baselines on Tulu3 and Openhermes2.5, we adjust certain parameters to fit the large-scale datasets. Specifically, for IFD computation, we directly use base models following (Li et al., 2024a). For *#InsTag* and the label set in MIG, we utilize the released InsTagger. To avoid the high cost of scoring millions of samples with ChatGPT (Chen et al., 2024), we adopt DEITA scores for the quality assessment of CaR, QDIT, and MIG as described in Sec 4.3.

4.2 Main Results

Main Comparison. Table 1 presents the performance of MIG for instruction data selection compared to several baselines across various benchmarks. All data selection methods are applied to select 50K samples, as detailed in the grid search experiment in Sec 4.3. Based on Llama3.1-8B, MIG outperforms all baselines on most tasks, with average improvements of **+1.49%** and **+1.96%** over state-of-the-art selection methods on knowledge-

Table 1: Comparison with data selection methods on the Tulu3 pool. Avg_{obj} and Avg_{sub} represent the average of the normalized knowledge-based and human-preference benchmark scores, respectively. AVG is the mean of Avg_{obj} and Avg_{sub}. MIG achieves the best performance on Avg_{obj}, Avg_{sub}, and AVG on both Llama3.1-8B and Mistral-7B.

Base Model	Method	Data Size	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
Llama3.1-8B	Pool	939K	69.15	63.88	83.40	63.41	65.77	67.10	68.79	8.94	6.86	-24.66	38.40	53.59
	Random	50K	74.24	64.80	70.36	51.22	63.86	61.00	64.25	8.57	7.06	-22.15	39.36	51.81
	ZIP	50K	77.63	63.00	52.54	35.98	65.00	61.00	59.19	6.71	6.64	-32.10	35.69	47.44
	IFD	50K	75.93	63.56	61.03	49.39	64.39	53.60	61.32	12.3	7.03	-20.20	40.83	51.08
	#InsTag	50K	72.54	64.80	69.83	48.17	63.50	**65.99**	64.14	6.58	6.84	-20.70	38.21	51.17
	DEITA	50K	78.98	66.11	**74.07**	49.39	64.00	64.33	66.15	10.19	6.83	-19.95	39.50	52.83
	CaR	50K	78.98	**69.04**	71.42	52.44	**65.15**	56.75	65.63	12.55	6.95	-20.67	40.57	53.10
	QDIT	50K	79.66	65.42	70.74	53.05	65.06	57.30	65.21	**15.78**	6.76	-20.56	41.03	53.12
	MIG	50K	**80.00**	66.39	72.02	**57.93**	64.44	65.06	**67.64**	14.66	**7.32**	**-17.77**	**42.99**	**55.32**
Mistral-7B	Random	50K	67.80	56.90	66.34	42.07	60.34	**65.43**	59.81	5.84	6.84	-25.20	37.21	48.51
	ZIP	50K	72.88	56.73	33.21	3.05	61.68	63.03	48.43	5.34	6.57	-36.17	34.32	41.37
	#InsTag	50K	**76.27**	57.15	66.34	40.85	61.80	63.22	60.94	8.20	6.91	-21.66	38.82	49.88
	DEITA	50K	75.93	57.72	64.82	11.59	61.41	64.51	56.00	8.82	6.96	-20.51	39.39	47.69
	CaR	50K	64.41	58.65	63.76	9.15	61.95	55.64	52.26	11.93	7.03	-17.82	41.11	46.58
	QDIT	50K	54.92	**58.68**	59.97	42.68	**62.46**	58.23	56.16	**15.03**	6.84	**-17.74**	41.52	48.84
	MIG	50K	75.25	56.19	**66.94**	**45.12**	60.23	64.70	**61.41**	13.66	**7.17**	-18.39	**42.05**	**51.73**



(a) (b) (c)

Figure 2: (a) Derivative of Information Score Functions. (b) Avg_{obj} on Different Information Score Functions. (c) Avg_{sub} on Different Quality Scores.

Table 2: Results on different data pools, Openhermes2.5 and X_{sota}, based on Llama3.1-8B. MIG outperforms all baselines across both data pools. Please refer to Table 5 6 in Appx. A.3 for detailed scores on all benchmarks.

	Openhermes2.5				X_{sota}			
	Data Size	Avg_{sub}	Avg_{obj}	AVG	Data Size	Avg_{sub}	Avg_{obj}	AVG
All	1M	36.91	61.49	49.20	300K	31.51	52.88	42.19
Random	50K	32.99	55.69	44.34	6K	29.94	49.69	39.81
#InsTag	50K	36.23	54.12	45.17	6K	31.89	46.19	39.04
DEITA	50K	36.80	57.36	47.08	6K	31.60	48.70	40.15
CaR	50K	37.51	55.57	46.54	6K	31.86	48.43	40.15
QDIT	50K	37.90	57.71	47.80	6K	32.52	49.10	40.81
MIG	50K	**38.12**	**58.30**	**48.21**	6K	**32.98**	**50.63**	**41.80**

Table 3: Grid search of appropriate data size and training epochs on the Tulu3 pool. We report the AVG score here.

	Random			MIG		
	Epoch2	Epoch3	Epoch4	Epoch2	Epoch3	Epoch4
10K	46.76	49.42	50.39	49.23	51.54	52.50
20K	48.23	50.36	51.08	50.98	52.87	52.84
50K	49.78	**51.81**	50.68	52.69	**54.22**	54.02

based and human-preference evaluations, respectively. When considering both knowledge-based and human-preference evaluations combined, MIG surpasses the second-best method, QDIT, by **+2.20%** on AVG score, highlighting the high quality and diversity of its sampled data. Notably, MIG is the only method that outperforms the model trained on the full Tulu3 pool, a comprehensive and high-quality SFT training dataset directly applicable to real-world scenarios, on AVG, despite utilizing only 50K samples. Specifically, MIG delivers substantial gains in human-preference performance, with an average improvement of **+4.59%** across three benchmarks while maintaining comparable performance on knowledge-based benchmarks. Previous research (Yuan et al., 2023; Dong et al., 2024) indicates that mathematical capability improves with the increasing training dataset size without plateauing. This explains the underperformance of 50K sample model compared to the model trained on the full pool on the GSM8K benchmark. Additionally, among methods that balance quality and diversity, MIG demonstrates superior efficiency on large pools. It is more efficient than DEITA and QDIT, as it eliminates the need for iterative pairwise similarity calculations in the embedding space. For detailed sampling times and efficiency analysis, please refer to Table 4 in

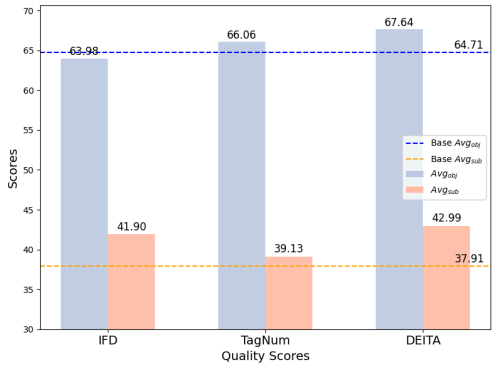


Figure 3: Quantitative results on different quality metrics. DEITA scores achieve the best performance on both human-preference and knowledge-based evaluations.

Appx. A.2.

Transferability on Models. To assess the generalizability of MIG, we additionally conduct experiments on Mistral-7B. As shown in Table 1, MIG outperforms all baseline methods with an improvement of **+1.85%** on AVG. Notably, the second-best selection method varies among different base models. Some strong baselines from Llama3.1-8B experience performance degradation when applied to Mistral-7B, further demonstrating the robustness of MIG.

Transferability on Data Pools. We conduct experiments on the comprehensive Openhermes2.5 pool and the relatively small, high-quality pool X_{sota} (Lu et al., 2024; Liu et al., 2024b) to further evaluate the robustness of MIG. Results in Table 2 show that MIG outperforms all baseline selection methods across different data pools and sample data sizes, Specifically, MIG improves AVG by **+3.87%** and **+1.99%** compared to random selection on the two data pools, and by **+0.41%** and **+0.99%** compared to previous SOTA methods. Notably, on the X_{sota}, all baseline methods exhibit performance degradation on knowledge-based evaluations, consistent with the findings in (Xia et al., 2024b). We hypothesize that quality metrics, such as DEITA scores and the number of tags, are biased toward multi-round, long samples, which tend to enhance subjective dialogue abilities and general knowledge. However, samples in specific domains, such as math and code, are typically single-turn. MIG mitigates this bias through its upper-convex information score function, providing a more effective selection for such domain-specific tasks.

4.3 Analysis

Study of Information Score Function Φ. The information score function Φ is crucial in MIG sampling as it balances quality and diversity. Based on the principles outlined in Sec. 3.2, Φ is expected to be monotonically increasing with a diminishing rate of increase. In our experiments, we evaluate two candidate functions:

$$\Phi(x) = 1 - e^{-\alpha x} \quad (\alpha > 0) \tag{12}$$

$$\Phi(x) = x^{\alpha x} \quad (0 < \alpha < 1) \tag{13}$$

Fig. 2(a) compares the decreasing rate in the derivative of these functions under varying parameter settings. Functions that decay rapidly tend to favor diverse label distributions as the information on any given label converges quickly. In contrast, slower decaying functions prioritize high-quality samples. Fig. 2(b)(c) present the quantitative performance on various benchmarks, with $\Phi(x) = x^{0.8}$ achieving the best results across both human-preference and knowledge-based evaluations, effectively balancing quality and diversity.

Study of Quality Measurement. Given the large scale of data pools in our experiments, reproducing methods from (Chen et al., 2024; Bukharin et al., 2024), which rely on computationally expensive external models such as ChatGPT for quality scoring, is impractical. Therefore, we implement three alternative quality measurement approaches: the number of tags (Lu et al., 2024), the IFD score (Li et al., 2024b), and the DEITA score (Liu et al., 2024b), to investigate their impact on information measurement. Fig. 3 compares these three quality metrics with a baseline score that assigns a constant value to all samples. The DEITA scores consistently outperform the other quality metrics in both evaluation settings. Therefore, we adopt the DEITA scores as the default quality measurements for MIG and other baseline selection methods.

Study of Label Graph. An essential question in MIG is how to determine an appropriate label graph, including its nodes (label set) and edges (label relationships). Increasing the number of nodes leads to a more granular label set, thereby providing broader coverage of knowledge topics. However, excessively large label sets inevitably include outliers or low-quality labels. Similarly, increasing edge density between labels enhances the comprehensiveness of label relationships, but overly dense graphs may result in computational inefficiencies and noises from the embedding model. There is

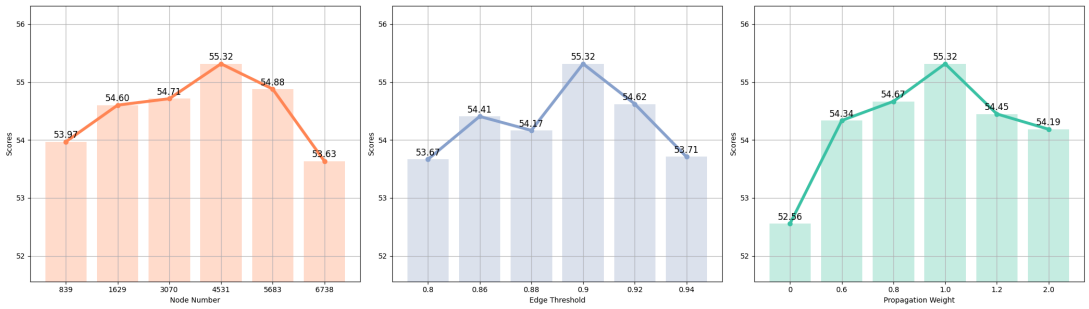


Figure 4: Analysis of Parameters in the Label Graph. The reported score is the average of Avg_{sub} and Avg_{obj}. (a) Comparison of various node counts (label set size) in the label graph. (b) Comparison of different edge thresholds, with a lower threshold indicating a dense graph. (c) Comparison of different propagation weights, where a smaller weight corresponds to weak propagation.

no universally optimal solution, as the ideal label graph depends on the characteristics of the data pool and potentially other parameters in MIG. To explore the relationship between the label graph and the downstream performance of trained models, we conduct an empirical experiment on the Tulu3 pool. Fig. 4(a) shows the downstream performance from a set of node counts in the label graph, ranging from 839 to 6738, while Fig. 4(b) presents performance across varying edge densities, with thresholds between 0.8 and 0.94. The observed trends align with our initial analysis, showing an unimodal performance curve in both experiments. For the Tulu3 pool, the optimal label graph is achieved with a label set size of 4531 and an edge similarity threshold of 0.9.

Study of Information Propagation. We conduct a series of experiments to study the impact of information propagation intensity in MIG sampling. Appropriate information propagation results in accurate information measurement over the label set. Specifically, we experiment with various values of α in Eq. 6, where a higher α corresponds to stronger information propagation and no propagation occurs when $\alpha = 0$. Fig 4(c) shows that $\alpha = 1.0$ yields the best performance, with an average improvement of 2.76 over the non-propagation. Notably, results with information propagation significantly outperform those without, indicating that information propagation effectively captures the relationship between labels, thereby improving the accuracy of information measurement on the label graph.

Grid Search for Data Size. To identify an appropriate data bucket within the Tulu3 pool for the main comparison and investigate the data scaling effects of MIG, we perform a grid search using various data budgets and training epochs. Table 3 shows that MIG consistently outperforms random selection across different data volumes, demonstrating its effectiveness. For the default setting in the Tulu3 pool, we select 50K samples with three training epochs, as both random and MIG sampling achieve the best performance under this configuration.

5 Conclusion

In this paper, we introduce a novel instruction-tuning dataset measurement method in semantic space. It models both the quality and diversity of the dataset information and balances these aspects through an upper-convex information score function. Additionally, it accurately captures the information distribution over label graph through information propagation. Accordingly, we propose an efficient sampling algorithm, MIG, that iteratively selects samples that maximize the information gain on the label graph. MIG demonstrates effectiveness across various data pools and base models, showcasing its robustness and adaptability. Our research bridges the gap between instance-level quality assessment and global dataset evaluation, offering a unified approach to dataset measurement. We hope our results can inspire dataset measurement design in the future.

Limitation. Currently, the parameters in MIG are static and depend on grid search to identify the optimal values, which can not be extensively explored. Future work could focus on developing methods to automatically determine the parameters in MIG, such as customizing the information score function for each label, to enhance the flexibility and scalability of MIG.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NIPS*.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. 2024. Data diversity matters for robust instruction tuning. In *EMNLP*.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. In *COLM*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *ICLR*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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

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

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Gérard Cornuéjols, George Nemhauser, and Laurence Wolsey. 1983. The uncapicitated facility location problem. Technical report, Cornell University Operations Research and Industrial Engineering.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *EMNLP*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *ACL*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and JingBo Zhu. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *EMNLP*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem.

2023a. CAMEL: Communicative agents for "mind" exploration of large language model society. In *NIPS*.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *ACL*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *NAACL*.

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*.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024a. SelectIT: Selective instruction tuning for LLMs via uncertainty-aware self-reflection. In *NIPS*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *ICLR*.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *ICLR*.

Michel Minoux. 2005. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. 2024. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. 2023. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024a. LESS: Selecting influential data for targeted instruction tuning. In *ICML*.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. 2024b. Rethinking data selection at scale: Random selection is almost all you need. *arXiv preprint arXiv:2410.09335*.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2024. Entropy Law: The Story Behind Data Compression and LLM Performance. *arXiv preprint arXiv:2407.06645*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024a. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*.

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

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: Rank responses to align language models with human feedback. In *NIPS*.

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Minghao Li, Fei Huang, Nevin L. Zhang, and Yongbin Li. 2024. Tree-instruct: A preliminary study of the intrinsic relationship between complexity and alignment. In *COLING*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NIPS*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *ACL*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. In *NIPS*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

A.1 Training and Evaluation Setup

Training Recipes. For experiments on X_{sota}, we follow the default settings in (Liu et al., 2024b). Specifically, we set the batch size to 128, learning rates at 2e-5, a warm ratio of 0.1, and a maximum input length of 2048. For experiments on the Tulu3 pool, we follow the settings in (Lambert et al., 2024). We set the batch size to 128, learning rates at 5e-6, a warm ratio of 0.03, and a maximum input length of 4096. For experiments on the Openhermes2.5 pool, we follow the settings in (Xia et al., 2024b). We set the batch size to 128, learning rates at 7e-6, a warm ratio of 0.01, and a maximum input length of 4096.

Evaluation Setup. The evaluation of our experiments is implemented on OpenCompass (Contributors, 2023).

Table 4: Efficiency comparison between different methods on 50K sampling from the Tulu3 pool. The timing reported here is measured by a single NVIDIA-L20Y.

Method	Time
InsTag	2.33
DEITA	81.56
QDIT	86.17
CaR	0.85
MIG	0.45

A.2 Efficiency Analysis

Table 4 presents the time used for 50K sampling on the Tulu3 pool. Among methods that balance quality and diversity, MIG achieves the highest efficiency. Notably, MIG is much more efficient compared to QDIT (Bukharin et al., 2024) and DEITA (Liu et al., 2024b), as it saves iterative pairwise similarity computation in the embedding space.

A.3 Detailed Results on Benchmarks

We provide detailed scores on full benchmarks in Table 5 6.

Table 5: Full Results on the Openhermes2.5 Pool.

Method	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
Pool	72.88	60.53	70.51	51.22	64.99	48.80	61.49	5.47	7.10	-31.51	36.91	49.20
Random	75.25	60.20	51.40	50.00	51.23	46.03	55.69	4.72	6.63	-44.12	32.99	44.34
InsTag	70.85	68.64	56.25	43.90	45.70	49.35	54.12	5.09	7.14	-35.60	36.23	45.17
DEITA	69.83	61.85	60.96	46.95	58.01	46.58	57.36	7.83	6.94	-33.69	36.80	47.08
CaR	62.71	63.73	55.42	44.51	64.37	42.70	55.57	7.33	7.09	-31.43	37.51	46.54
QDIT	66.44	62.45	58.61	50.00	63.64	45.10	57.71	9.19	6.99	-30.78	37.90	47.80
MIG	78.98	63.33	51.55	45.73	63.81	46.40	58.30	7.83	7.17	-30.34	38.12	48.21

Table 6: Full Results on the X_{sota} Pool.

Method	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
Pool	73.22	54.12	40.49	45.12	61.05	43.25	52.88	3.85	6.78	-54.21	31.51	42.19
Random	61.02	58.12	32.07	42.69	62.31	41.96	49.69	3.60	6.34	-54.39	29.94	39.81
InsTag	64.07	51.82	36.62	28.66	55.11	40.85	46.19	5.22	6.56	-50.28	31.89	39.04
DEITA	71.86	50.82	27.67	40.24	63.36	38.26	48.70	4.22	6.48	-48.44	31.60	40.15
CaR	72.88	48.90	20.92	46.95	62.68	38.26	48.43	5.22	6.51	-49.46	31.86	40.15
QDIT	71.53	51.48	29.95	41.46	63.22	36.97	49.10	5.09	6.55	-46.05	32.52	40.81
MIG	74.58	51.93	31.54	43.90	62.24	39.56	50.63	5.34	6.72	-47.18	32.98	41.80