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

Yicheng Chen[1,2], **Yining Li**[1†], **Kai Hu**[1,3], **Zerun Ma**[1], **Haochen Ye**[1], **Kai Chen**[1†]
[1]Shanghai AI Laboratory [2]Fudan University [3]Carnegie Mellon University

Project page: https://yichengchen24.github.io/projects/mig

Abstract

Data quality and diversity are key to the construction of 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 accurately capture the intent of complex instructions in the semantic space. To bridge this gap, we propose a unified method for quantifying the information content of datasets. 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 a 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.

1 Introduction

Large Language Models (LLMs) have shown remarkable capabilities in following human instructions in a wide range of tasks (Wang et al., 2023a). Typically, LLMs first acquire general knowledge through large-scale pretraining and are subsequently refined through instruction tuning to better align with diverse human intentions (Brown et al., 2020; Taori et al., 2023; Touvron et al., 2023). Instruction tuning utilizes instruction-response pairs to guide base models toward more accurate and contextually appropriate responses. Recent studies (Zhou et al., 2023a; Chen et al., 2024) emphasize the critical role of data engineering in instruction tuning, highlighting data quality rather than quantity as the key to effective instruction tuning. Notably, LIMA (Zhou et al., 2023a) demonstrates that just 1000 high-quality, human-curated instructions can achieve performance comparable to substantially larger datasets. However, manual curation of such datasets is inherently time-consuming and labor-intensive (Chiang et al., 2023).

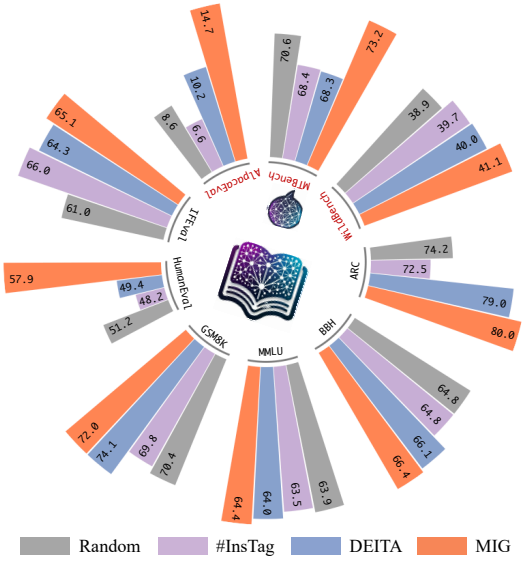


Figure 1: Comparison with different data selection methods (Lu et al., 2024; Liu et al., 2024b) on the Tulu3 (Lambert et al., 2024) pool using Llama3.1-8B (Touvron et al., 2023), evaluated on (black) knowledge-based benchmarks and (red) human-preference benchmarks. See details in Sec. 4.2.

More recently, a line of work (Chen et al., 2024; Lu et al., 2024; Liu et al., 2024b; Bukharin et al., 2024) proposes automatic selections of optimal subsets from extensive data pools by defining desirable data characteristics. These approaches (Bukharin et al., 2024; Yu et al., 2024b) posit that **quality** and

† Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 9902–9915

diversity are crucial for an effective instruction-tuning dataset. Data quality is defined from multiple perspectives, such as instruction complexity (Lu et al., 2024; Zhao et al., 2024), model perplexity and uncertainty (Li et al., 2024b), or scores assigned by advanced external models (Chen et al., 2024; Liu et al., 2024b). However, diversity remains less explicitly quantified, often addressed via heuristic methods such as maximizing label set coverage (Lu et al., 2024), reducing redundancy through diversity filters (Liu et al., 2024b), or enforcing fixed sample distributions per cluster (Ge et al., 2024; Yu et al., 2024b). This narrow focus on diversity only during later selection stages, without a comprehensive view of the entire dataset, diminishes the global diversity and representativeness of the sampled data. Alternative methods (Bukharin et al., 2024) employing embedding-based facility location functions (Cornuéjols et al., 1983) quantify diversity but require computationally intensive iterative pairwise distance calculations, making them impractical for large datasets. Additionally, distance-based clustering in the embedding space may fail to capture the semantic intent of complex instructions accurately. To solve these issues, several essential questions are raised: 1) How can we effectively quantify diversity in semantic space while balancing quality and diversity in dataset evaluation? 2) How can we efficiently select data based on such evaluations?

To this end, we propose an information-based measure for instruction-tuning datasets and introduce an efficient data selection algorithm that aims to **M**aximize the **I**nformation **G**ain (MIG). We model the semantic space as a label graph, with nodes representing labels and edges capturing semantic relationships. Information in the dataset is distributed across this graph, with the total information being the sum of each label's information. Each data point contributes to its associated labels in proportion to its quality. Thus, the information of each data point measures local data quality, while the total of all label information measures the global diversity of the dataset. To balance quality and diversity, we apply a monotonically increasing but marginally diminishing function to compute label information, thereby promoting diversity and preventing excessive data concentration on particular labels. To better model information distribution in semantic space, we propagate information along label graph edges to address semantic correlations and annotation biases. Leveraging the submodularity of our proposed information-based dataset measurement, we implement an efficient greedy algorithm that iteratively selects data points that maximize the information gain according to the current state of the label graph.

Through extensive experiments across data pools (Liu et al., 2024a; Teknium, 2023; Lambert et al., 2024) of varying quality and sizes, and LLMs of different families (Touvron et al., 2023; Jiang et al., 2023; Yang et al., 2024), MIG consistently achieves superior performance on both human-preference and knowledge-based evaluations. As shown in Fig 1, on the Tulu3 (Lambert et al., 2024) pool with Llama3.1-8B as the base model, 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). Notably, the model fine-tuned with 5% Tulu3 data sampled by MIG outperforms the official SFT model trained on the full dataset by **+1.73%** (average on nine benchmarks), with a substantial boost of **+4.59%** in human-preference evaluations. MIG also outperforms existing 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 significantly enhances sampling efficiency, reducing sampling time by over 100-fold on the Tulu3 data pool compared to embedding-based methods.

In summary, our contributions are as follows:

- We propose an information-based measurement for instruction-tuning datasets in semantic space. It quantifies quality and diversity within the information distributed across the semantic label graph.
- We introduce MIG, an efficient data selection algorithm that maximizes the information gain on the label graph iteratively.
- Extensive experiments on various data pools, base models, and benchmarks demonstrate the effectiveness and generalizability of MIG. The correlation between parameters in MIG and the attributes of sampled data is well studied.

2 Related Work

Data Selection for Instruction Tuning. Recent studies (Zhou et al., 2023a; Chen et al., 2024) indicate that increasing data quality and diversity rather than quantity effectively boosts instruction-following performance. Consequently, data selection methods aim to identify optimal subsets that meet such characteristics and generally fall into three categories: (1) **Quality-based approaches** prioritize high-quality data points, where quality is defined through various perspectives, such as instruction complexity and response quality. INSTRUCTMINING (Cao et al., 2024b) identifies natural language metrics indicative of 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 intrinsic uncertainty 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 broad coverage of the data pool. DiverseEvol (Wu et al., 2023) iteratively selects samples distant from previously selected data in the embedding space. ZIP (Yin et al., 2024) prioritizes subsets with low compression ratios, implicitly favoring diversity. **(3) Comprehensive approaches** strive to balance 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) cluster data and sample high-quality points from each cluster. However, these methods typically rely on heuristic rules rather than a unified quantitative metric to balance quality and diversity.

Submodular function for Diversity Measurement. Traditional submodular functions, such as facility location, graph cut, and log determinant, effectively quantify dataset diversity by identifying representative, 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. Although this NP-hard problem can be approximated with a greedy algorithm following submodularity (Nemhauser et al., 1978; Minoux, 2005), embedding-based methods are inefficient at scale due to the high storage and computational costs of calculating high-dimensional pairwise distances. To mitigate this issue, our custom dataset measurement is submodular, which justifies the use of a greedy strategy, while MIG samples data in a high-level semantic space, substantially reducing computational overhead.

3 Method

As shown in Fig. 2(a), we begin by annotating the raw data pool with a tagger and scorer. Next, MIG constructs a label graph to measure dataset information (Sec. 3.2) and selects a subset for subsequent SFT by maximizing the information gain (Sec. 3.3).

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,

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

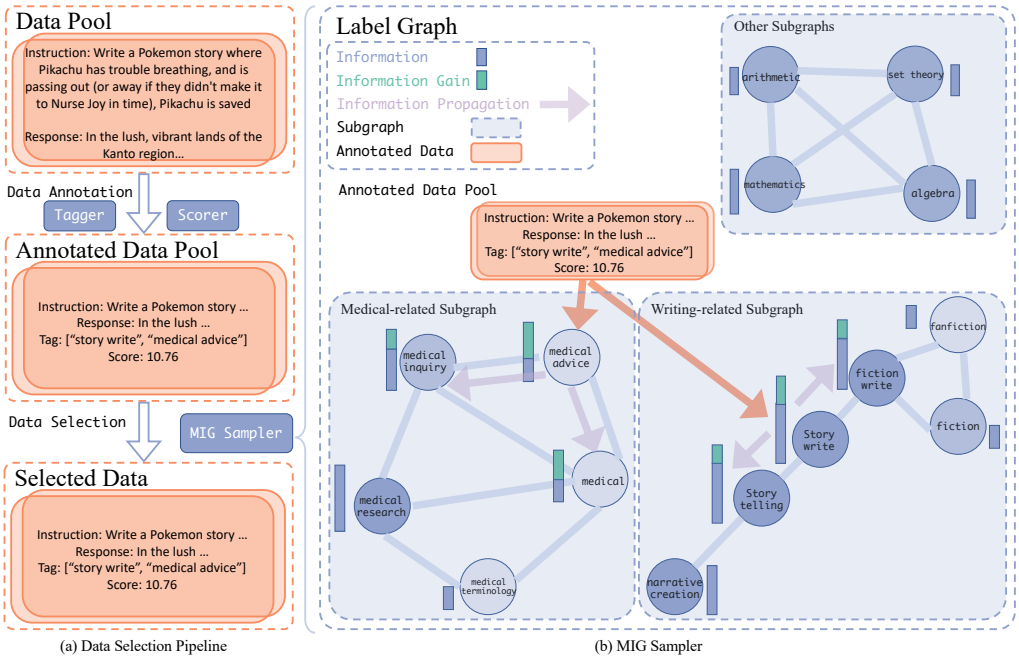
Data. Each data point is formed as:

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

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 labels (including embedding-based clusters) are independent, ignoring the semantic relationships among them. However, such label associations are crucial for accurately capturing the information distribution in semantic space. Intuitively, we can



(a) Data Selection Pipeline (b) MIG Sampler

Figure 2: 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 data. Next, MIG constructs the label graph based on the label set and iteratively selects the data point that maximizes the information gain within the graph. The selected data are used for supervised fine-tuning (SFT) of LLMs.

model labels as nodes, their associations as edges, and the intensity of associations as edge weights, thus modeling semantic space as 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 label similarities as edge weights and remove edges whose weights are below a threshold T to ensure computational efficiency. Therefore, 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 textual similarity between label l_p and l_q, and $\sigma(\cdot)$ yields 1 when the input is evaluated as True.

Data Point Information. 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:

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

Semantic overlaps between labels and annotation-induced bias can lead to inaccurate information distribution. To address this, we introduce information propagation along the edges of the label graph, enabling a more accurate modeling of information distribution across the semantic space. 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 hyperparameter controlling the intensity of information propagation. Let A be the propagation matrix, then the propagated information vector of d_i is:

$$\hat{\mathbf{e}}_i = A\mathbf{e}_i \tag{7}$$

Dataset Information. To balance quality and diversity within the label graph, we apply a monotonically increasing yet upper-convex function ϕ to compute the label information. The marginally diminishing information gain is negatively correlated with the existing label information. Thus, information gains on labels with less information are prioritized. Formally, the dataset information is:

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

where Φ is a nonlinear transformation that applies ϕ element-wise to the input and then aggregates the results by summation.

3.3 MIG Sampling

Directly selecting D_S from D_P is computationally infeasible as the combination $C_{|D_P|}^N$ grows quickly.

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**
  $G \leftarrow A\Phi'(A\sum_{k\in D_S} E_k)$;
  $d_i \leftarrow \text{argmax}_{d\in D_P} GE_d$;
  $D_S \leftarrow D_S \cup \{d_i\}$;
  $D_P \leftarrow D_P \setminus \{d_i\}$;
return D_S

Thus, as shown in Fig 2(b), we follow the submodularity of $E(D)$ (detailed in Appx. C) and 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:

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

where Φ' represents the derivative of Φ. Thus, the selection process can be formed as:

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

The sampling is detailed in Alg. 1. Refer to Appx. A.2 for more implementation details of MIG.

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 million-level records across a wide variety of subjects, including mathematics, programming, and user dialogues.

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

• X_{sota} (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 model performance comprehensively. The evaluation is conducted using OpenCompass (Contributors, 2023), with the average results reported as normalized scores on a percentage scale. Detailed evaluation settings are provided in Appx. A.4.

• Human-preference Benchmarks. We evaluate open-ended dialogue abilities using model-based evaluation metrics on three benchmarks: AlpacaEvalv2 (Dubois et al., 2024), MTBench (Zheng et al., 2023), and WildBench (Lin et al., 2024).

• Knowledge-based Benchmarks. We assess the factual knowledge, reasoning, coding, mathematical, and instruction-following abilities using automatic metrics on six benchmarks: ARC (Clark et al., 2018), Big-Bench-Hard(BBH) (Suzgun et al., 2022), MMLU (Hendrycks et al., 2021), HumanEval (Chen et al., 2021), GSM8k (Cobbe et al., 2021), and IFEval (Zhou et al., 2023b).

Baselines. We compare our methods against strong data selection approaches: random selection (Xia et al., 2024b), 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). To replicate baselines on Tulu3 and Openhermes2.5, we adjust certain parameters to fit the large-scale datasets, as detailed in Appx A.1.

Training. We use LLaMA3.1-8B (Touvron et al., 2023), Mistral-7B-v0.3 (Jiang et al., 2023), and Qwen2.5-7B (Yang et al., 2024) as our base models and fine-tune them using the Llama-Factory framework (Zheng et al., 2024). Please refer to Appx. A.3 for detailed training setup.

4.2 Main Results

Main Comparison. Table 1 presents the performance of MIG and baselines across benchmarks. All methods select 50K samples based on the grid search (Sec 4.3). With Llama3.1-8B, MIG outperforms all baselines on most tasks, with average improvements of **+1.49%** and **+1.96%** over previous state-of-the-art selection methods on knowledge-based and human-preference evaluations, respectively. MIG surpasses QDIT, the second-best

Table 1: Comparison with data selection methods on the Tulu3 pool. HE denotes HumanEval, AE denotes AlpacaEvalv2, MT denotes MTBench, and Wild denotes WildBench. 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 all base models.

Base Model	Method	Data Size	ARC	BBH	GSM	HE	MMLU	IFEval	Avg_{obj}	AE	MT	Wild	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	12.30	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-v0.3	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**
Qwen2.5-7B	Pool	939K	90.51	65.01	85.29	78.05	75.15	64.88	76.31	9.07	7.04	-23.98	39.16	57.74
	Random	50K	85.42	63.87	80.74	79.27	73.81	58.04	75.53	10.56	7.18	-18.08	41.11	57.32
	ZIP	50K	85.76	63.43	83.24	72.56	73.60	58.23	72.80	7.45	7.33	-27.83	38.94	55.87
	#InsTag	50K	88.81	63.03	84.61	**81.10**	73.50	61.00	75.34	9.07	7.52	-18.80	41.62	58.48
	DEITA	50K	89.15	63.22	86.13	79.27	74.27	58.78	75.14	10.31	7.28	-19.71	41.09	58.11
	CaR	50K	**91.86**	65.60	**87.64**	77.44	73.97	50.28	74.47	13.66	7.39	-20.77	42.39	58.43
	QDIT	50K	89.83	**69.34**	87.04	**81.10**	**74.72**	50.83	75.48	**13.79**	7.10	-20.46	41.52	58.50
	MIG	50K	90.51	67.39	84.46	79.88	73.85	**61.74**	**76.30**	11.80	**7.54**	**-14.49**	**43.32**	**59.81**

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. D for detailed scores on all benchmarks.

	Openhermes2.5				X_{sota}			
	Data Size	Avg_{sub}	Avg_{obj}	Avg	Data Size	Avg_{sub}	Avg_{obj}	Avg
Pool	1M	36.91	61.49	49.20	306K	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**

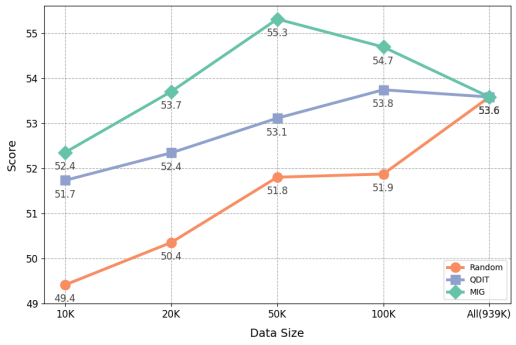


Figure 3: Data scaling experiments on Tulu3 using Llama3.1-8B. The score reported here is the Avg score.

method, by **+2.20%** on overall Avg score. Notably, the model trained on 5% data sampled by MIG outperforms the model trained on the full Tulu3 pool by **+4.59%** on human-preference benchmarks while maintaining comparable knowledge-based performance. Additionally, MIG significantly outperforms embedding-based methods in sampling efficiency due to reduced computational overhead. Please refer to Table 4 in Appx. B for detailed sampling times and efficiency analysis.

Transferability on Models. Table 1 presents results for Mistral-7B and Qwen2.5-7B. MIG consistently surpasses baselines with Avg improvements of **+1.85%** and **+1.31%**, respectively, demonstrating its robustness. Notably, the second-best selection method varies among different base models, further demonstrating the generalizability of MIG.

Transferability on Data Pools. Table 2 presents results on different data pools with varying sizes and quality. MIG consistently outperforms all baselines, achieving Avg improvements of **+0.41%** and **+0.99%** over previous best methods, further demonstrating its generalizability. Notably, on X_{sota}, all baselines 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 tag counts, are biased toward multi-round, long samples that enhance subjective chat abilities. However, samples in specific domains, such as math and code, are typically single-turn. MIG mitigates this bias by effectively balancing quality and diversity.

Data Scaling. We compare MIG with baseline methods across varying data budgets on the Tulu3 pool, using Llama3.1-8B as the base model. As

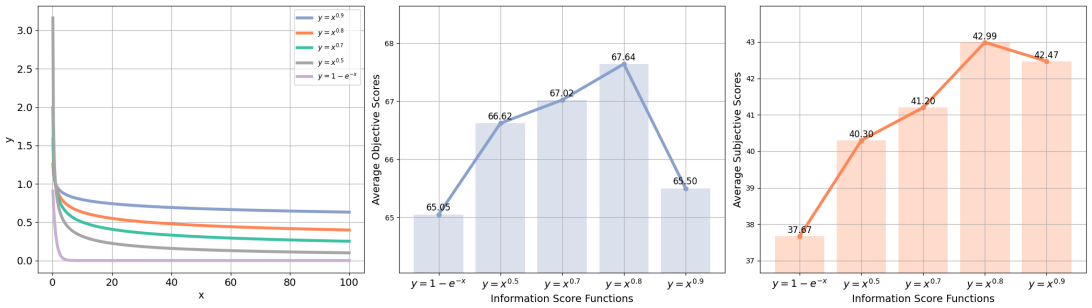


Figure 4: (a) Derivative of Information Score Functions. (b) Avg_{obj} on Different Information Score Functions. (c) Avg_{sub} on Different Quality Scores.

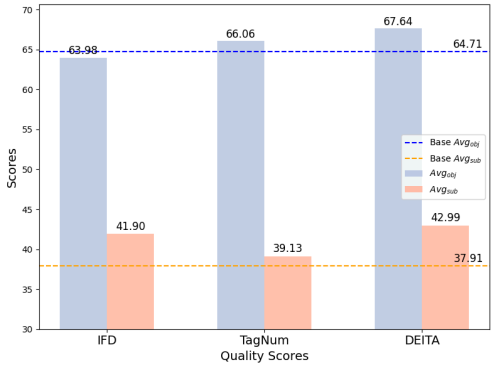


Figure 5: Quantitative results on different quality metrics. DEITA scores achieve the best performance on both human-preference and knowledge-based evaluations.

shown in Fig 3, MIG consistently delivers superior performance at each data budget, demonstrating its robust scalability. Remarkably, MIG achieves comparable performance to the full dataset with only 20K samples, underscoring its efficiency. The observed initial increase and subsequent plateau in performance align closely with findings from previous works (Li et al., 2024b; Liu et al., 2024b), highlighting the importance of data selection.

4.3 Analysis

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:

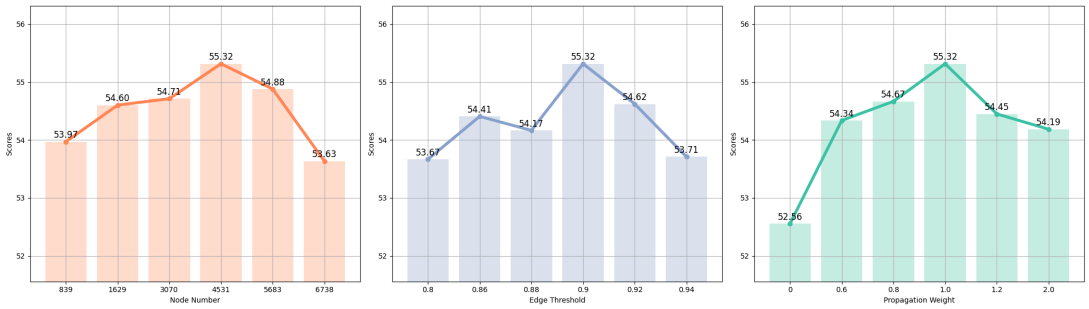
$$\Phi(x) = 1 - e^{-\alpha x} \quad (\alpha > 0) \tag{12}$$

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

Fig. 4(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. Fig. 4(b)(c) present the performance on different evaluations, with $\Phi(x) = x^{0.8}$ achieving the best results on human-preference and knowledge-based benchmarks, effectively balancing quality and diversity.

Quality Metrics. 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. 5 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.

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 noise from the embedding model. There is 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. 6(a) shows the downstream performance from a set of node counts in the label graph, ranging from 839 to 6738, while Fig. 6(b) presents performance across varying edge densities, with thresholds between 0.8 and 0.94. The observed trends align with our



(a) (b) (c)

Figure 6: Analysis of Parameters in the Label Graph. The reported score is the average of Avg_{sub} and Avg_{obj}. Please refer to Table 7 8 9 in Appx. D for detailed scores on all evaluated benchmarks. (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.

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	49.13	52.36	51.11
20K	48.23	50.36	51.08	51.18	53.71	53.14
50K	49.78	**51.81**	50.68	52.88	**55.32**	55.14

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.

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 distribution in the semantic space. Specifically, we experiment with various values of α in Eq. 6, where α is proportional to the intensity of information propagation. Fig 6(c) shows that $\alpha = 1.0$ yields the best performance, with Avg improvement of **2.76** over the non-propagation. It indicates that information propagation effectively improves the accuracy of information measurement on the label graph.

Grid Search. To identify an appropriate data bucket and training epoch on the Tulu3 pool for the main comparison, we perform a grid search. Results in Table 3 indicate 50K samples with three training epochs as optimal for Tulu3, consistently maximizing performance for MIG and random selection.

5 Conclusion

In this paper, we propose a novel method for measuring instruction-tuning datasets in semantic space. We model the semantic space as a label graph and jointly evaluate data quality and diversity. We introduce an upper-convex information score function to balance quality and diversity, and propose information propagation to capture the information distribution accurately Building on the submodularity of such measurement, we propose MIG, an efficient sampling algorithm that iteratively selects samples to maximize the information gain on the label graph. Extensive experiments across diverse data pools and base models validate the effectiveness and generalizability of MIG. Our research bridges the gap between instance-level quality assessment and global dataset-level evaluation, offering a unified approach to dataset measurement. We hope our results can inspire dataset measurement-guided data selection 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.

Acknowledgements

This work was supported by National Key R&D Program of China 2022ZD0161600.

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*.

Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. 2024a. Compassjudger-1: All-in-one judge model helps model evaluation and evolution. *arXiv preprint arXiv:2410.16256*.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024b. 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*.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. 2019. dbscan: Fast density-based clustering with r. *Journal of Statistical Software*.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

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. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

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*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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*.

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 Details of Experiments Setup

A.1 Baseline Settings

The specific baseline settings in our experiments are as follows:

• **Random**. A fixed random seed of 42 is used to ensure reproducibility.

• **IFD** (Li et al., 2024b). We follow the setting from (Li et al., 2024a) to directly compute IFD scores using base LLMs for efficiency.

• **ZIP** (Yin et al., 2024). The default setting is applied across all data pools.

• ***#InsTag*** (Lu et al., 2024). The open-released InsTagger is used to tag data from the pool, followed by tag normalization, which includes frequency filtering and semantic aggregation. The frequency threshold is set to 2, as InsTagger is fully trained on valid normalized tags, resulting in 9471 tags. For semantic aggregation, we use E5-Mistral-7B-Instruct (Wang et al., 2023b) to generate tag embeddings, and the DBSCAN algorithm (Hahsler et al., 2019) is applied with a semantic similarity threshold of 0.05, yielding 6738 tags.

• DEITA (Liu et al., 2024b). For X_{sota}, we use the released sampled dataset. For the Openhermes2.5 (Teknium, 2023) and Tulu3 (Lambert et al., 2024) pools, we reproduce its method. Quality assessment is conducted using the released quality and complexity scorers. For *Repr Filter*, we utilize Llama3.1-8B to obtain instance embeddings. A threshold of 0.9 is applied to the Tulu3 pool, and 0.95 is used for the Openhermes2.5 pool, as the 0.9 threshold results in insufficient samples for the latter.

• CaR (Ge et al., 2024). We follow the default setting, using all-mpnet-base-v2 (Reimers and Gurevych, 2019) to obtain data embeddings, PCA to retain 95% of dimensions, and k-Means clustering with the number of clusters as $k = \sqrt{n/2}$. For ranking, we use the released IQS model. We maintain the original ratio between $n1$ and kn_2 for different pools.

• QDIT (Bukharin et al., 2024). Embedding computation follows the default setting with all-mpnet-base-v2. Quality assessment uses the DEITA scores instead of ChatGPT. The hyperparameter α for balancing quality and diversity is set to 0.9 for X_{sota} and 0.7 for the Tulu3 and Openhermes2.5 pools.

A.2 Implementation Details

The label set in MIG follows *#InsTag* (Lu et al., 2024), with variations in the label graph across different data pools. Specifically, 3059 tags are used for X_{sota}, 4531 for Tulu3, and 5166 for Openhermes2.5, with label set size positively correlated to pool size. E5-Mistral-7B-Instruct is used as the embedding model to compute label similarity, with a threshold set to 0.9. Quality assessment in MIG uses the DEITA scores. The information score function is set to $\phi(x) = x^{0.8}$, and the information propagation weight α is set to 1.

A.3 Training Recipes

For experiments on X_{sota}, we follow the default settings from (Liu et al., 2024b), using a batch size of 128, a learning rate of 2e-5, a warm ratio of 0.1, and a maximum input length of 2048. For the Tulu3 pool, we adopt the settings from (Lambert et al., 2024), with a batch size of 128, a learning rate of 5e-6, a warm ratio of 0.03, and a maximum input length of 4096. For the Openhermes2.5 pool, we follow the settings from (Xia et al., 2024b), setting the batch size to 128, learning rates to 7e-6, a warm ratio of 0.01, and a maximum input length of 4096.

A.4 Evaluation Setup

The evaluation of our experiments is implemented using OpenCompass (Contributors, 2023) with greedy inference to ensure uniform evaluation across all models.

Human-preference Evaluations. We use the open-source CompassJudger-1-32B (Cao et al., 2024a) for human-preference evaluation. As different benchmarks use different scoring metrics and ranges, we normalize the scores according to the following mapping:

• AlpacaEvalv2 (Dubois et al., 2024): The score range is 0-100, requiring no special adjustment.

• MTBench (Zheng et al., 2023): The score range is 0-10, which is mapped by multiplying by 10.

• WildBench (Lin et al., 2024): The score range is -100-100, which is normalized by adding 100 and dividing by 2.

Knowledge-based Evaluations. We conduct evaluations using the following settings: three-shot evaluation on BBH (Suzgun et al., 2022), five-shot evaluation on MMLU (Hendrycks et al., 2021), and zero-shot evaluation on ARC (Clark et al., 2018) and GSM8K (Cobbe et al., 2021). For Hu-

Table 4: Efficiency comparison of different methods for 50K sampling from the Tulu3 pool, with timing measured on a single NVIDIA-L20Y.

Method	GPU	Time
Random	×	0.09
ZIP (Yin et al., 2024)	×	53.99
IFD (Li et al., 2024b)	×	0.05
#InsTag (Lu et al., 2024)	×	2.33
DEITA (Liu et al., 2024b)	✓	81.56
QDIT (Bukharin et al., 2024)	✓	86.17
CaR (Ge et al., 2024)	✓	0.85
MIG	✓	0.45

manEval (Chen et al., 2021), we report pass@1 results, and for IFEval (Zhou et al., 2023b), we provide strictly followed scores.

B Efficiency Analysis

Table 4 presents the time used for 50K sampling on the Tulu3 pool. Among methods that balance quality and diversity, MIG demonstrates the highest efficiency. Notably, MIG outperforms QDIT (Bukharin et al., 2024) and DEITA (Liu et al., 2024b) significantly, as it eliminates the need for iterative pairwise similarity computations in the embedding space.

C Theoretical Analysis of Greedy Strategy in MIG

C.1 Submodularity Analysis

We first prove that our dataset measurement function $E(D)$, defined in Eq. 8, is submodular. Specifically, for all $D \subseteq T$, and for all elements $e \notin T$, the following inequality holds:

$$E(D \cup e) - E(D) \geq E(T \cup e) - E(T) \tag{14}$$

Proof. Let $\mathbf{z}(D) = A \sum_{i \in D} s_i \mathbf{v}_i$. The marginal gain from adding an element e to D is:

$$\Delta(D, e) = \sum_{k=1}^{n} [\phi(z_k(D) + \Delta_k(e)) - \phi(z_k(D))] \tag{15}$$

where $\Delta_k(e) = s_e(A\mathbf{v}_e)_k \geq 0$ represents the incremental contribution of e to the k-th component. Since ϕ is monotonically increasing and concave, for any $\delta \geq 0$ and $z' \geq z$, we have:

$$\phi(z + \delta) - \phi(z) \geq \phi(z' + \delta) - \phi(z') \tag{16}$$

Given $D \subseteq T$, we have:

$$z_k(T) = z_k(D) + \sum_{i \in T \setminus D} s_i (A\mathbf{v}_i)_k \geq z_k(D) \tag{17}$$

Thus, by the concavity property of ϕ:

$$\begin{aligned} &\phi(z_k(D) + \Delta_k(e)) - \phi(z_k(D)) \geq \\ &\phi(z_k(T) + \Delta_k(e)) - \phi(z_k(T)) \end{aligned} \tag{18}$$

Summing over all components k, we get:

$$\Delta(D, e) \geq \Delta(T, e) \tag{19}$$

Thus, $E(D \cup e) - E(D) \geq E(T \cup e) - E(T)$, which satisfies the definition of submodularity. □

C.2 Cardinality-Constrained Submodular Maximization

Given that $E(D)$ is submodular, our data selection task, defined in Eq. 1, constitutes a cardinality-constrained submodular maximization problem that is NP-complete. However, a greedy algorithm provides a well-established approximation guarantee, ensuring that $E(D^{greedy}) \geq (1 - \frac{1}{e})E(D^*)$, where D^* represents the optimal solution (Nemhauser et al., 1978). Assuming $P \neq NP$, this guarantee is the best achievable for polynomial-time algorithms.

D Detailed Results on Benchmarks

We provide detailed scores on full benchmarks in Table 5 6 7 8 9.

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

Table 7: Full Results across Different Node Numbers.

Node Number	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
839	73.90	66.35	73.31	52.44	64.42	61.18	65.27	16.02	7.18	-19.59	42.67	53.97
1629	76.27	66.22	72.18	56.71	64.71	58.96	65.84	15.16	7.40	-18.15	43.36	54.60
3070	78.31	66.80	72.25	55.49	65.03	63.22	66.85	14.04	7.25	-17.63	42.58	54.71
4531	80.00	66.39	72.02	57.93	64.44	65.06	67.64	14.66	7.32	-17.77	42.99	55.32
5683	83.73	66.47	72.55	55.49	64.22	64.14	67.77	12.55	7.35	-20.15	41.99	54.88
6738	76.27	65.92	70.13	52.44	64.78	64.14	65.61	11.30	7.36	-19.87	41.65	53.63

Table 8: Full Results across Different Edge Densities.

Node Number	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
0.8	77.97	65.69	70.96	54.88	64.59	63.96	66.34	11.3	7.24	-21.45	40.99	53.67
0.86	80.68	65.72	71.49	55.49	64.15	62.85	66.73	14.04	7.21	-19.77	42.08	54.41
0.88	76.95	67.77	72.02	57.93	64.99	62.85	67.09	13.42	7.04	-20.17	41.25	54.17
0.9	80.00	66.39	72.02	57.93	64.44	65.06	67.64	14.66	7.32	-17.77	42.99	55.32
0.92	83.73	66.39	71.80	58.54	64.35	65.43	68.37	12.55	6.97	-19.29	40.87	54.62
0.94	78.31	65.86	70.36	56.10	64.81	64.14	66.60	10.68	7.15	-19.42	40.82	53.71

Table 9: Full Results across Different Propagation Weights.

Node Number	ARC	BBH	GSM8K	HumanEval	MMLU	IFEval	Avg_{obj}	AlpacaEval	MTbench	Wildbench	Avg_{sub}	AVG
0	74.24	65.39	71.04	49.39	64.62	63.59	64.71	10.81	7.09	-21.01	40.40	52.56
0.6	80.68	66.78	72.33	54.88	64.66	63.40	67.12	13.29	7.13	-19.87	41.55	54.34
0.8	78.98	66.46	73.54	54.88	64.83	64.14	67.14	14.04	7.14	-17.71	42.19	54.67
1.0	80.00	66.39	72.02	57.93	64.44	65.06	67.64	14.66	7.32	-17.77	42.99	55.32
1.2	80.00	67.42	73.39	52.44	64.89	64.14	67.04	12.42	7.20	-17.72	41.85	54.45
2.0	81.36	67.03	70.36	54.88	64.86	63.96	67.08	13.54	7.06	-20.49	41.30	54.19