What Makes Good Instruction-Tuning Data? An In-Context Learning Perspective

Guangzeng Han
University of Memphis
ghan@memphis.edu

Xiaolei Huang
University of Memphis
xiaolei.huang@memphis.edu

Abstract

Instruction-tuning datasets often contain substantial redundancy and low-quality samples, necessitating effective data selection methods. We propose an instruction data selection framework based on weighted in-context influence (wICI), which measures how effectively each candidate example reduces instruction-following difficulty for semantically related peers. Through systematic experiments, we address three key questions: what constitutes effective instruction tuning data from an in-context perspective, whether sample difficulty correlates with in-context influence, and how in-context influence translates to instruction tuning effectiveness. Experiments across multiple models and benchmarks demonstrate that our method consistently outperforms existing baselines under constrained data budgets, while empirically showing that sample difficulty negatively correlates with in-context influence.[1]

1 Introduction

In-context learning (ICL) (Dong et al., 2024) enables large language models (LLMs) to adapt its behavior at inference time by conditioning on a small set of demonstration examples. By prepending a handful of instruction–response pairs to the input prompt, the model leverages its pre-trained knowledge to generalize to new tasks without any parameter updates. Previous work has explored how to select effective demonstrations, using criteria such as semantic similarity (Li and Qiu, 2023; Dong et al., 2024), diversity, or model feedback (Wang et al., 2024a; Ye et al., 2023), to maximize performance on a certain task. Instruction tuning (Lou et al., 2024b), by contrast, updates model parameters through fine-tuning on large collections of instruction–response pairs (Wang et al., 2023; Xu et al., 2023). This paradigm has proven effective at improving instruction-following ability, but assembling high-quality tuning datasets is costly. Data selection methods like Superfiltering (Li et al., 2024a) use model perplexity to prune simple examples, and DEITA (Liu et al., 2024b) apply learned reward models to rank and filter instruction–response pairs based on human-aligned quality signals.

Despite the successes of each paradigm in isolation, the relationship between examples that perform well in in-context learning and those that constitute valuable instruction-tuning data remains underexplored. Recognizing this gap, (Li et al., 2024c) made a pioneering attempt in NUGGETS to bridge in-context learning and instruction tuning data curation paradigms. NUGGETS evaluates data quality by using each candidate instruction as a one-shot demonstration and measuring its impact on performance across a fixed global anchor set, establishing the first connection between ICL and instruction tuning. While NUGGETS made significant progress in this direction, we explore how this paradigm can be further advanced. Specifically, NUGGETS evaluates all candidates using the same fixed global anchor set regardless of semantic relevance, employs simple binary scoring without considering improvement magnitude or task difficulty, and requires substantial computational resources by evaluating each candidate on large anchor sets (typically 1,000 samples). We hypothesize that by introducing dynamic, semantically relevant probe sets and generalization-based weighting mechanisms, we can more precisely measure the influence of samples while significantly improving computational efficiency. We pose three key research questions to validate this hypothesis. **RQ1:** *From the perspective of in-context learning, what kind of data is good instruction tuning data?* **RQ2:** *Are samples that are difficult for a model necessarily strong demonstrations or tuning examples?* **RQ3:** *Do examples that yield high in-context influence also lead to superior instruction-following*

[1]The code is available at: https://github.com/trust-nlp/SyntheticData-Curator

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1013–1027
July 2-7, 2026 ©2026 Association for Computational Linguistics

performance when used for fine-tuning?

To answer these questions, we introduce a weighted in-context influence framework for data selection. For each candidate example we build a probe set in three stages that ensures semantic relatedness, diversity, and challenge. We then measure single-probe influence as the absolute reduction in instruction-following difficulty when the candidate is used as a one-shot demonstration. These influences are aggregated with a normalized cosine-distance weight to emphasize transferable gains. Finally, we rank by the aggregated score and select examples greedily under a diversity constraint on cosine similarity. By quantifying each example's influence on peers, we select a subset of examples that maximally improves overall instruction-following under a constrained data budget.

Our contributions are threefold. First, we introduce a weighted in-context influence framework that evaluates samples through dynamic probe sets and difficulty weighting rather than fixed global evaluation. Second, we demonstrate that local peer influence outperforms global assessment and that sample difficulty negatively correlates with teaching effectiveness. Third, extensive experiments show our method consistently outperforms existing baselines across multiple benchmarks while achieving substantial computational efficiency.

2 Preliminary

In this section, we formalize our problem setup of data selection for instruction tuning, and introduce the concepts of perplexity and its ratio-based form, instruction–following difficulty (IFD), as intrinsic measures of sample difficulty.

Data Selection of Instruction Tuning Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ be an instruction–response corpus, where $x_i = T(\textit{Instruction}, [\textit{Input}])$ denotes the full prompt produced by applying a prompt template T to the instruction (and optional input) and the reference response y_i. The average negative log-likelihood $\mathcal{L}$ of an LLM $f_{\boldsymbol{\theta}}$ on D is defined as:

$$\mathcal{L}(\boldsymbol{\theta}; D) = -\frac{1}{n}\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i \mid x_i). \tag{1}$$

A lower $\mathcal{L}$ indicates stronger instruction-following ability. Given the data budget k , our goal is to select a subset $Q \subseteq D$ such that $|Q| = k$, after instruction tuning on Q, the resulting model achieves the strongest instruction-following ability:

$$Q^{\star} = \underset{Q \subseteq D,\, |Q|=k}{\arg\min}\ \mathcal{L}(\boldsymbol{\theta}_Q; D_{\text{test}}). \tag{2}$$

Perplexity and Instruction–Following Difficulty (IFD). For a sample of response length N, perplexity is defined as:

$$\text{PPL}(y_i|x_i) = \exp\left(-\frac{1}{N}\sum_{j=1}^{N}\log p(y_{i,j}|x_i, y_{i,1}, ..., y_{i,j-1})\right) \tag{3}$$

The $\mathcal{L}$ refers to instruction-following ability and its exponential form, perplexity, an ideal method for measuring the difficulty of generating the response. Following (Li et al., 2024b), we measure *instruction–following difficulty* (IFD) as the benefit the instruction provides over unconditional generation: where a larger IFD indicates that the model gains less from the instruction.

$$\text{IFD}(y \mid x) = \frac{\text{PPL}(y \mid x)}{\text{PPL}(y)}, \tag{4}$$

3 Method

In this section, we present the proposed methodology in Figure 1. The key idea follows our hypothesis: *In-context demonstrations that substantially reduce the instruction-following difficulty of challenging tasks are likely to be high-quality samples for instruction tuning.* Our method consists of four major stages: 1) diversity-aware in-context probe retrieval, 2) in-context influence evaluation, 3) diversity-constraint data selection and 4) model training.

3.1 Diversity-aware In-context Probe Retrieval

To assess how much a candidate demonstration helps related examples in in-context learning, we first construct a high-quality probe set. A naive retrieval of random or nearest-neighbor samples often fails to capture a demonstration's true utility: unrelated probes introduce noise, redundant probes provide limited additional signal, trivially simple probes obscure the model's genuine capability to follow complex instructions. To overcome these limitations, we design a three-stage retrieval process consisting of Semantic Retrieval, Semantic Clustering, and Complexity-guided Selection, which ensures that the probe set is semantically relevant, diverse in coverage, and sufficiently challenging for evaluation.

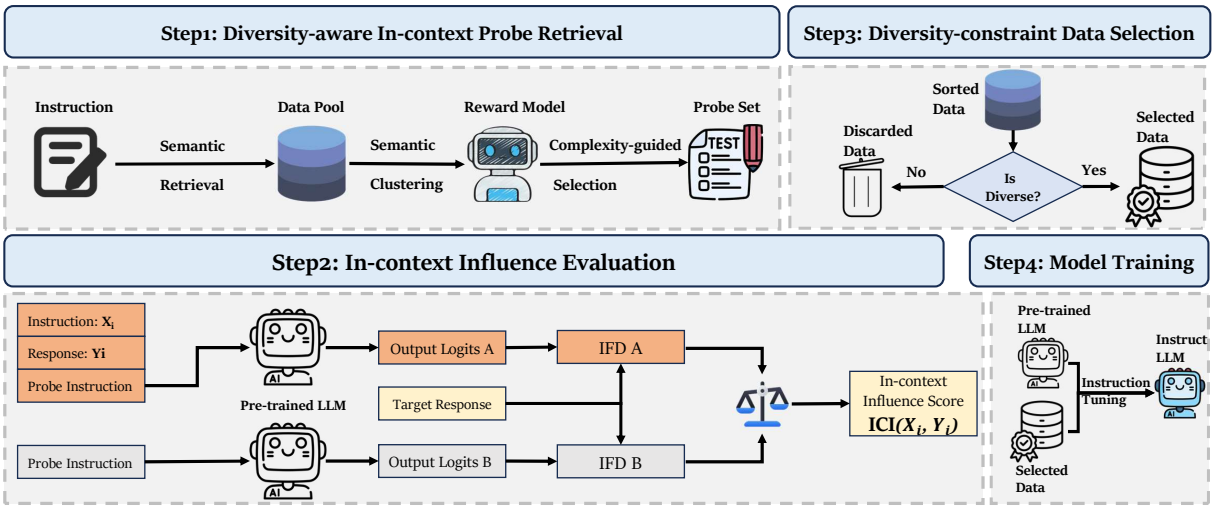


Figure 1: Overview of our framework.

Semantic Retrieval. In-context influence can only be measured meaningfully when the probes share a coherent topical space with the instruction; otherwise, their responses may reflect task mismatch rather than demonstrational assistance. For each instruction x_i, we retrieve its N nearest neighbors in the embedding space under the Euclidean distance:

$$\mathcal{N}_i^N = \underset{j \neq i}{\arg\,\mathrm{topN}} \big(-\ell_2(f(x_i), f(x_j))\big). \quad (5)$$

where $f(\cdot)$ denotes the sentence encoder.

Semantic Clustering. While semantic retrieval enforces relevance, it tends to produce highly redundant neighbors concentrated in a narrow region of the embedding space. Such redundancy weakens the generality of influence estimation and biases evaluation toward repetitive behaviors. To mitigate this, we apply K-means clustering to partition $\mathcal{N}_i^N$ into K semantically distinct groups. Each cluster corresponds to a local semantic mode, thereby ensuring that the probe set explores diverse contexts related to x_i.

Complexity-guided Selection. Finally, the probe set must include examples that are sufficiently challenging to reveal genuine improvements in instruction following. As observed by Liu et al. (2022), trivially simple probes provide little information about a demonstration's true capacity, whereas complex probes expose whether it can genuinely lower instruction-following difficulty. To operationalize this insight, we employ a reward model $R(\cdot)$ [2] that estimates instruction complexity. Within each cluster, we rank candidates by $R(x_j)$ and select the highest-scoring instance:

$$\mathcal{B}_i = \bigcup_{c=1}^{K} \underset{x_j \in \mathrm{cluster}_c}{\arg\max}\, R(x_j). \quad (6)$$

The resulting probe set $\mathcal{B}_i$ integrates semantic relevance, diversity, and challenge, forming a robust basis for subsequent in-context influence evaluation.

3.2 In-context Influence Evaluation

While instruction–following difficulty (IFD) serves as an intrinsic indicator of how hard an instruction is for a model to complete, it does not capture how one example can *influence* the model's behavior on other tasks during in-context learning. To address this limitation, we propose a new metric, the **In-context Influence (ICI)**, which quantifies how much a candidate demonstration improves the model's instruction-following performance on its probes. Given a candidate demonstration $a_i = (x_i, y_i)$ and its probe set $\mathcal{B}_i = \{(x_b, y_b)\}$ obtained from Step 1, we define the in-context influence on a single probe $b \in \mathcal{B}_i$ as the absolute reduction in instruction–following difficulty (IFD) when a_i is provided as an in-context example:

$$\mathrm{ICI}_{i \to b} = \mathrm{IFD}(y_b \mid x_b) - \mathrm{IFD}(y_b \mid a_i, x_b). \quad (7)$$

A positive $\mathrm{ICI}_{i \to b}$ indicates that the demonstration a_i reduces the difficulty of probe b, while a negative value reflects detrimental influence. To

[2] https://huggingface.co/hkust-nlp/deita-complexity-scorer

explicitly encourage generalization beyond trivial semantic similarity, we weight each single-probe influence by a normalized cosine distance that is *monotonically increasing* with semantic distance. We then define the overall weightd in-context influence *(wICI)* as:

$$\mathrm{wICI}(a_i) = \sum_{b \in \mathcal{B}_i} \frac{1 - \cos\big(f(x_i), f(x_b)\big)}{2\,|\mathcal{B}_i|}\ \mathrm{ICI}_{i \to b}. \quad (8)$$

3.3 Diversity-constraint Data Selection and Model Training

To ensure that the selected data not only have high informativeness but also cover diverse semantic patterns of target samples, we introduce a diversity-constraint mechanism during data selection. Once every data candidate has a wICI score, we sort candidates in descending order and greedily build the coreset by adding a candidate $a_i = (x_i, y_i)$ only if it is not overly similar to any already selected item; otherwise it is skipped. Let $f(\cdot)$ be the sentence encoder and τ a cosine-similarity threshold; the admission test is

$$\max_{a_j \in \mathcal{S}} \cos\big(f(x_i), f(x_j)\big) < \tau, \quad (9)$$

and we continue until $|\mathcal{S}| = k$. The resulting coreset is then used, without any additional weighting, to fine-tune the pre-trained LLM to improve the instruction following ability.

4 Experimental Settings

4.1 Training Datasets

We use two open-source instruction-tuning datasets: Alpaca-GPT4 (Peng et al., 2023), which is synthesized through the Self-Instruct (Wang et al., 2023) paradigm and provides diverse, higher quality instruction–response pairs, and WizardLM (Xu et al., 2023), which applies instruction evolution to expand basic prompts into harder multi step tasks to strengthen compositional reasoning. We adopt Wizard-70K and follow Li et al. (2024b) to filter low quality samples. For fairness, we cap the training data at 10% for our method and all baselines.

4.2 Baselines

Superfiltering (IFD) (Li et al., 2024a) observes a strong consistency in the Influence Function Direction (IFD) between small and large language models when applied to the same data. Based on this insight, it adopts a weak-to-strong paradigm: a smaller model is used to compute IFD scores and select data, which is then used to fine-tune a larger model.

DEITA (Liu et al., 2024b) identifies three key factors in data selection: instruction complexity, instruction-response quality, and data diversity. It employs two reward models distilled from proprietary LLMs to assess instruction complexity and instruction-response quality, respectively. Additionally, it incorporates a diversity constraint during the data selection process to ensure a more varied dataset.

NUGGETS (Li et al., 2024c) evaluates each instruction example's potential to serve as an effective one-shot demonstration by measuring its impact on model performance across a diverse anchor set of tasks. Specifically, NUGGETS computes a "golden score" for each instruction by comparing the model's zero-shot performance against its one-shot performance when using that instruction as context.

SelectIT (Liu et al., 2024a) exploits intrinsic uncertainty in LLMs to select high-quality instruction tuning data through three levels of self-reflection: token-level confidence analysis, sentence-level prompt variance reduction, and model-level collaborative assessment.

4.3 Evaluation Setup

4.3.1 Pair-wise Comparison

To assess the effectiveness of different data selection strategies, we perform pair-wise comparisons between models fine-tuned on selected 10% subsets of data and a reference model fine-tuned on the full data. We use five publicly available test sets for evaluation: WizardLM (Xu et al., 2023), Self-Instruct (Wang et al., 2023), Vicuna (Chiang et al., 2023), Koala (Vu et al., 2023), and LIMA (Zhou et al., 2023a). For each instruction in these datasets, both the full-data model and the subset-trained model are prompted to generate responses. These outputs are then evaluated by a strong LLM judge [3], which compares the two responses and assigns a preference score.

To reduce position bias, each response pair is presented to the judge in both possible orders. A model is considered the winner for a given instruction if its response is preferred or not worse in both orders. Based on this, we compute a *winning*

[3]We use GPT-4.1-mini as the judge. For the prompt used for evaluation, please refer to the Appendix.

score for each method to quantify its relative performance against the full-data baseline. The score is defined as:

$$\text{winning\_score}(\mathcal{D}_t) = \frac{\text{num(wins)} - \text{num(loses)}}{\text{num}(\mathcal{D}_t)} + 1 \quad (10)$$

4.3.2 Benchmark Evaluation

In addition to pair-wise comparison, we evaluate models on a set of widely used benchmark datasets including ARC-Challenge (Clark et al., 2018), HELLASWAG (HS) (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021), MT-bench (Zheng et al., 2023) and 2Wikimqa, HotpotQA, Samsum from LongBench (Bai et al., 2025). Additionally, we report results on the AlpacaEval 2.0 (Dubois et al., 2024) leaderboard using both Win Rate (WR) and Length-Controlled Win Rate (LC) metrics.

Dataset	Strategy	Llama3.1-8B	Mistral-7B
Alpaca-GPT4	Full	1.000	1.000
	IFD	1.198	1.248
	DEITA	1.076	1.099
	NUGGETS	1.133	1.201
	SelectIT	1.146	1.227
	Ours	**1.215**	**1.261**
WizardLM	Full	1.000	1.000
	IFD	**1.186**	1.294
	DEITA	1.114	1.140
	NUGGETS	1.133	1.249
	SelectIT	1.176	1.281
	Ours	1.169	**1.308**

Table 1: Pairwise evaluation performance using Llama3.1-8b and Mistral-7B-v0.3 respectively.

5 Main Experimental Results (RQ1)

5.1 Pairwise Evaluation Results

The pairwise evaluation in Table 1 shows that our data selection yields consistent and substantial gains, achieving competitive or superior performance compared to existing baselines across all three models and datasets. Relative to "Full" our method improves the winning score by +21.5% and +26.1% on Alpaca-GPT4 when fine-tuning Llama3.1-8B and Mistral, respectively. On WizardLM, our method improves the winning score by +16.9% and +30.8% with Llama3.1-8B and Mistral, respectively. Overall, these results indicate that our data selection is highly effective across different models and datasets.

5.2 Benchmark Results

Table 2 presents the zero-shot benchmark performance of models trained on the full dataset versus those fine-tuned on just 10% of the data using baselines and our method.

First, the full-data models underperform the 10% data-selection methods across nearly every evaluation, including both zero-shot benchmarks and the Alpaca-Eval 2.0 leaderboard, indicating substantial redundancy and noise in the original training set. Second, different data selection baselines exhibit task-specific strengths: on Alpaca-Eval 2.0, IFD outperforms DEITA, whereas on BBH, DEITA surpasses IFD. This indicates that different selection criteria steer model improvements along different capability dimensions rather than in a single unified direction. Third, our budget-constrained selection matches or surpasses strong baselines and full-data models across zero-shot benchmarks and Alpaca-Eval 2.0. Although some baselines peak on individual metrics, our selection remains top two in nearly all settings, showing robust and broad effectiveness under a strict budget. The final finding answers our **RQ1**: *In-context demonstrations that substantially reduce the instruction-following difficulty of diverse challenging tasks are high-quality samples for instruction tuning.*

6 In-Depth Analysis

In this section we present a series of experiments and analyses to answer the rest two of our three research questions: **RQ2**: *Are difficult data necessarily good demonstrations of in-context learning?* and **RQ3**: *Are good in-context learning demonstrations good examples of instruction tuning?*

6.1 Data Consistency Analysis (RQ2)

To test whether samples that are difficult for the model are also effective in context demonstrations, we ranked every dataset twice, first by the instruction following difficulty (IFD) and then by unweighted in-context influence (ICI), and compared the two rankings. We measured the overlap between the lists at the top 10%, 30%, and 50% cutoffs and computed the Spearman correlation over the full lists. As shown in Table 3, top-10% overlap is limited at 10.1% on Alpaca-GPT4 and 14.4% on WizardLM, rising to about 38–39% at the top 30% and 59–65% at the top 50%. Spearman correlations are positive but moderate at 0.39 and 0.26. These results indicate a meaningful yet incomplete

	ARC-C Acc	HS Acc	MMLU Acc	BBH EM	GSM8k EM	2Wikimqa F1	HotpotQA F1	Samsum ROUGE-L	MT-Bench Score	AE 2.0 WR	AE 2.0 LC
Llama3.1-8B w/ Alpaca-GPT4											
Full	52.99	79.78	61.81	40.90	47.46	39.44	41.73	36.03	4.30	5.65	13.19
IFD	53.50	80.95	63.37	35.52	53.42	**43.52**	44.28	36.21	4.84	**7.58**	**14.85**
DEITA	58.21	80.46	62.77	**40.92**	52.05	42.99	42.12	35.11	4.68	5.27	12.14
NUGGETS	57.42	80.76	62.89	39.91	52.77	43.35	42.92	34.81	4.55	5.85	13.03
SelectIT	58.16	81.24	62.56	39.93	53.90	42.03	44.48	35.10	4.73	6.22	13.90
Ours	**58.98**	**81.52**	**63.45**	39.33	**55.17**	43.47	**44.74**	**36.60**	**4.88**	7.50	14.42
Llama3.1-8B w/ Wizard											
Full	54.61	78.36	61.32	42.20	55.42	45.71	56.10	30.09	4.75	6.02	14.75
IFD	56.40	80.42	63.65	40.67	50.64	47.06	55.03	30.05	**5.41**	**6.34**	12.82
DEITA	57.29	79.19	63.52	38.70	51.83	46.37	56.02	31.63	5.23	5.90	11.42
NUGGETS	56.50	79.08	64.04	39.71	50.22	43.35	53.43	**32.68**	4.91	5.74	11.71
SelectIT	57.17	80.47	64.11	40.54	51.03	45.99	56.19	30.55	5.02	5.57	11.43
Ours	**57.79**	**81.02**	**64.90**	**41.21**	**52.84**	**47.06**	**57.14**	32.06	5.28	6.11	**13.13**
Mistral-7B-v0.3 w/ Alpaca-GPT4											
Full	44.03	73.01	51.40	31.18	18.73	37.65	37.00	21.91	3.80	5.65	13.19
IFD	48.81	78.42	**55.50**	33.68	26.80	42.11	34.54	33.16	3.98	**6.50**	**11.92**
DEITA	48.33	80.52	54.14	33.70	27.43	**42.33**	34.54	33.91	4.14	5.92	11.02
NUGGETS	48.25	80.33	53.29	33.52	27.99	39.89	31.05	31.41	4.22	5.10	10.84
SelectIT	49.01	80.21	54.04	33.44	28.26	40.54	32.72	32.49	**4.32**	6.12	11.35
Ours	**49.43**	**81.14**	54.73	**34.24**	**28.53**	41.57	**34.92**	**34.50**	4.18	6.26	11.35
Mistral-7B-v0.3 w/ Wizard											
Full	46.25	73.57	51.15	31.84	32.37	44.76	48.85	21.16	3.97	4.60	10.77
IFD	50.08	78.39	55.99	36.55	28.89	**42.88**	**47.40**	29.08	4.15	5.84	9.91
DEITA	47.76	76.25	54.81	**37.18**	28.85	41.91	46.80	28.66	4.35	5.09	11.32
NUGGETS	50.23	77.36	55.69	36.37	27.64	40.43	44.77	28.02	4.32	4.88	10.51
SelectIT	51.02	78.43	55.10	37.09	28.01	41.27	47.33	28.78	4.10	5.25	11.18
Ours	**51.27**	**78.51**	**56.31**	36.99	**29.44**	42.47	46.53	**29.22**	**4.40**	**5.91**	**11.36**

Table 2: Benchmark zero-shot evaluation of data-selection methods using 10% of the training data. "Full" denotes performance on the entire dataset for reference. WR and LC stand for Winning Rate and Length-Controlled winning rate, respectively.

alignment between instruction–following difficulty and in-context influence; thus, difficult samples are not reliably strong in-context demonstrations, answering **RQ2** in the negative.

Further, we adopt the approach of (Lou et al., 2024a) by using the Berkeley Neural Parser to extract root verbs and their direct objects from our extracted instruction subset. We then visualize the 20 most frequent root verbs alongside their four most common direct-object nouns under three conditions: (1) High Instruction Following Difficulty, (2) High In-context Influence, and (3) Random. As shown in Figure 2, the distributions differ substantially across these criteria, reinforcing our conclusion that sample difficulty and in-context learning utility capture distinct properties.

Dataset	**Overlap Ratios** 10%	30%	50%	**Rank Corr.**
Alpaca-GPT4	0.1006	0.3874	0.6476	0.3947
WizardLM	0.1442	0.3650	0.5942	0.2568

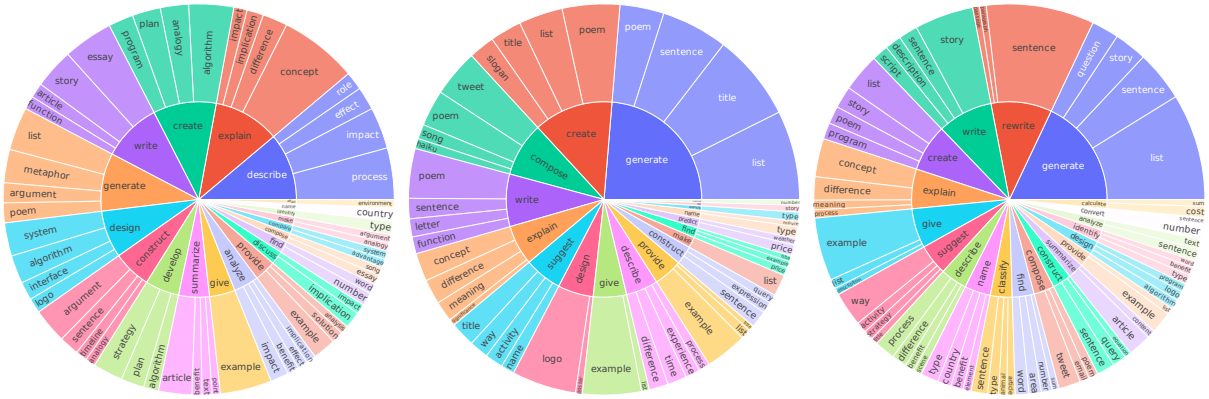
Table 3: Relation of high IFD data and high ICI data.

6.2 Ablation Study (RQ3)

We ablate two diversity components to quantify their contributions and answer RQ3. The variant *w/o DA* removes Semantic Clustering in Step 1 (no probe-side diversity during influence estimation), and *w/o DS* keeps influence scoring but drops the cosine-similarity constraint in selection (no demonstration-side diversity). As shown in Table 4, both variants underperform our full method across datasets and models, yet both still achieve winning scores greater than 1.0, which indicates they outperform training on the full data. Therefore, for **RQ3**: *Are good in-context demonstrations good examples of instruction tuning?* the answer is **yes**, even without diversity modules they provide gains over the full-data baseline, however, they are most effective when coupled with diversity on both the probe side and the demonstration side.

6.3 Performance with Varied Budgets

We next study how budget size influences each data selection method. We progressively raise the bud-



(a) High Instruction Following Difficulty. (b) High In-context Influence. (c) Random.

Figure 2: Visualization of the verb–noun structures in instructions selected by the three data-selection strategies on Alpaca-GPT4. The inner ring shows the predominant verbs, the outer ring displays the nouns that co-occur directly with those verbs.

Dataset	Strategy	Llama3.1-8B	Mistral-7B
Alpaca-GPT4	w/o DA	1.140	1.181
	w/o DS	1.155	1.198
	Ours	1.215	1.261
WizardLM	w/o DA	1.132	1.204
	w/o DS	1.154	1.239
	Ours	1.169	1.308

Table 4: Ablation study on two major sample-selection components. *w/o DA* removes Semantic Clustering in Step 1, *w/o DS* drops the cosine-similarity constraint in selection.

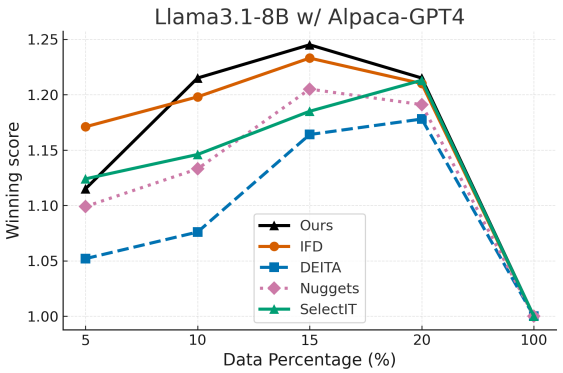


Figure 3: Winning-score curves of LLAMA3.1-8B trained on Alpaca-GPT4.

get from five to twenty percent of the training pool and record the winning score of LLAMA3.1-8B. As shown in Figure 3, our weighted-ICI method shows a steep performance climb between five and ten percent, reaches its apex at fifteen percent, and then drifts slightly downward as more data are added. Most baseline methods, including DEITA, NUGGETS, and SelectIT, follow similar curves but consistently lag behind our approach at every budget level, with IFD showing the closest performance to our method. This behavior reveals two distinct phases of data addition. In the low-budget regime, carefully selected demonstrations introduce novel and complementary information, so increasing the data volume leads to better models. However, beyond a certain saturation point, additional examples tend to overlap with existing information or introduce noise, resulting in diminishing or even negative returns. Because our strategy prioritizes demonstrations based on their ability to aid the most challenging probes, it reaches the saturation threshold with far fewer samples, thereby achieving superior sample efficiency.

6.4 Domain Generalization Analysis

To examine whether our data selection method generalizes beyond the Alpaca-GPT4 and WizardLM settings, we additionally conduct experiments in a medical domain using the MedQuAD (Ben Abacha and Demner-Fushman, 2019) dataset. Specifically, we select 30% of the training data with our method, fine-tune LLMs on the resulting subset, and evaluate them on three medical benchmarks: MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2021), and MMLU-med. We compare our method against Random selection and Full-data fine-tuning. This setting provides a targeted test of whether the proposed wICI-based selection strategy remains effective when transferred to a domain that differs substantially from the general-purpose instruction-tuning corpora used in our main experiments.

Table 5 reports the results. Overall, our method

	MedMCQA	MedQA	MMLU-med
Llama-3.1-8B			
Full	**40.53**	44.22	**71.33**
Random	32.32	29.53	45.67
Ours	39.63	**46.03**	65.33
Mistral-7B-v0.3			
Full	36.55	34.32	**51.67**
Random	33.11	37.00	36.00
Ours	**37.05**	**39.54**	50.00

Table 5: Results on the medical domain. Our method and the Random baseline use 30% of the training data, while Full uses the entire dataset. The best performance in each column is shown in bold.

consistently outperforms Random selection across both model backbones on most evaluation metrics, showing that the selected subset remains substantially more effective than random 30% sample even in the medical domain. Compared with Full-data fine-tuning, the results are more mixed: our method achieves the best performance on MedQA for Llama-3.1-8B and on both MedMCQA and MedQA for Mistral, while trailing the Full baseline on MMLU-med and on MedMCQA for Llama-3.1-8B. These results suggest that wICI generalizes beyond the domains considered in the main experiments and can still identify highly useful training subsets in a specialized domain. At the same time, the remaining gap on some settings indicates that full-data coverage may still be particularly valuable for certain types of domain-specific knowledge.

7 Related Work

In-Context Learning for Instruction Tuning In-context learning (ICL) (Brown et al., 2020) and instruction tuning (IT) (Lou et al., 2024b) represent two fundamental paradigms for adapting large language models to downstream tasks. ICL enables models to perform tasks by conditioning on demonstration examples provided in the prompt, without updating model parameters. In contrast, instruction tuning updates model parameters through supervised fine-tuning on instruction-response pairs to improve instruction-following capabilities.

Recent theoretical work has revealed deep connections between these paradigms. (Dai et al., 2023) provided a groundbreaking theoretical explanation, demonstrating that Transformer attention has a dual form of gradient descent and that ICL can be understood as implicit fine-tuning where GPT produces meta-gradients according to demonstration examples. (Von Oswald et al., 2023) further demonstrated that transformers can implement learning algorithms implicitly, showing that this process corresponds to gradient-based optimization of principled objective functions. (Duan et al., 2024) empirically validated these theoretical insights, finding that ICL changes an LLM's hidden states as if the demonstrations were used to instructionally tune the model, establishing that "ICL is implicit IT." Building on these theoretical foundations, several works have explored practical applications. (Li et al., 2024c) made a pioneering contribution by proposing NUGGETS, which leverages the ICL-IT connection for practical data selection. NUGGETS evaluates instruction tuning candidates by measuring their effectiveness as one-shot demonstrations across a global anchor set, computing "golden scores" based on performance improvements. (Mosbach et al., 2023) conducted a comprehensive comparison between few-shot fine-tuning and ICL, finding that both approaches achieve similar generalization performance when controlling for model size and training examples. In addition, related work has used ICL to synthesize instruction-tuning data. (Han et al., 2025; Xu et al., 2023)

Our work introduces In-Context Influence (ICI) to measure how effectively a candidate example reduces instruction-following difficulty for semantically related peers. Through local semantic relevance and difficulty weighting mechanisms, we achieve more precise and computationally efficient data selection than existing global evaluation approaches.

Instruction Data Selection Instruction-tuning corpora, whether in general-purpose settings or domain-specific scenarios (Rao et al., 2024; Wei et al., 2025; Rao et al., 2026), often contain redundancy and noise, motivating the selection of compact, high-value subsets. Quality-based approaches score or mine examples, including AlpaGasus (Chen et al., 2024), DEITA (Liu et al., 2024b), Superfiltering (Li et al., 2024a), CherryLLM (Li et al., 2024b), Instruction Mining (Cao et al., 2024), and the style-consistency method SCAR (Li et al., 2025). Alignment-oriented filtering reduces hallucination, for example NOVA (Si et al., 2025), and a call for rigor (Moon et al., 2025) highlights confounds from inconsistent training setups. Diversity-driven methods such as QDIT (Bukharin et al., 2024) and DPP (Wang et al., 2024b) enhance cov-

erage and minimize redundancy through greedy or determinantal objectives. Information-theoretic and multi-signal approaches include MIG (Chen et al., 2025) for information gain, GRAPE (Zhang et al., 2025) for response fitness, ZIP (Yin et al., 2024) for entropy-based compression, and NICE (Wang et al., 2025) for task-level selection. A recent survey (Liu et al., 2025) synthesizes these directions and proposes a unified framework.

Our work complements these lines by selecting examples that maximize in-context influence on related and challenging probes, measured as absolute reductions in instruction–following difficulty, and by enforcing diversity during probe construction and in the final coreset. By selecting examples according to their measured in-context influence, we align data selection with the actual inference-time benefit of demonstrations, producing compact yet transferable subsets while controlling redundancy through diversity.

8 Conclusion

We presented a data selection framework for instruction tuning based on a new influence metric. The metric, In-context Influence, measures the reduction in instruction-following difficulty caused by a candidate demonstration, and its generalization-weighted aggregate guides selection. Our pipeline builds probe sets that are related, diverse, and challenging, and applies a diversity constraint during selection to avoid redundancy. Under a 10% budget, the method matches or surpasses strong baselines and full-data models across different benchmarks. We also answered three research questions. **RQ1**: demonstrations that substantially reduce difficulty on diverse challenging probes are high-quality samples for instruction tuning. **RQ2**: difficult samples are not reliably strong in-context demonstrations. **RQ3**: good in-context demonstrations are effective instruction-tuning examples, and they work best when both probe diversity and demonstration diversity are enforced.

Limitations

Our study has two primary limitations. First, due to computational constraints, we did not evaluate larger backbones such as Llama3-70B (Dubey et al., 2024), nor did we test on substantially larger instruction corpora such as Tulu3 (Lambert et al., 2024). Second, we focused on supervised instruction tuning and did not evaluate other post-training methods such as DPO (Rafailov et al., 2023), PPO (Schulman et al., 2017) and their variants. We expect our selection framework to transfer to larger models and alternative post-training methods, which we leave to future work.

Acknowledgment

The authors thank anonymous reviewers for their insightful feedback. The project was partially supported by the National Science Foundation (NSF) under awards CNS-2318210 and TI-2434589 (OpenAI API expenses). We thank the computing resources provided by the iTiger GPU cluster (Sharif et al., 2025) supported by the NSF MRI program under the award CNS-2318210.

References

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *BMC bioinformatics*, 20(1):511.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexander Bukharin, 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 *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3411–3425, Miami, Florida, USA. Association for Computational Linguistics.

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

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 *The Twelfth International Conference on Learning Representations*.

Yicheng Chen, Yining Li, Kai Hu, Ma Zerun, HaochenYe HaochenYe, and Kai Chen. 2025. MIG:

Automatic data selection for instruction tuning by maximizing information gain in semantic space. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9902–9915, Vienna, Austria. Association for Computational Linguistics.

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:1803.05457v1*.

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

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Hanyu Duan, Yixuan Tang, Yi Yang, Ahmed Abbasi, and Kar Yan Tam. 2024. Exploring the relationship between in-context learning and instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3197–3210, Miami, Florida, USA. Association for Computational Linguistics.

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

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*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Guangzeng Han, Weisi Liu, and Xiaolei Huang. 2025. Attributes as textual genes: Leveraging LLMs as genetic algorithm simulators for conditional synthetic data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19367–19389, Suzhou, China. Association for Computational Linguistics.

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

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 *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7595–7628, Mexico City, Mexico. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024c. One-shot learning as instruction data prospector for large language models. In *Proceedings of the 62nd Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 4586–4601, Bangkok, Thailand. Association for Computational Linguistics.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. 2025. SCAR: Data selection via style consistency-aware response ranking for efficient instruction-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12790, Vienna, Austria. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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 *The Twelfth International Conference on Learning Representations*.

Ziche Liu, Rui Ke, Yajiao Liu, Feng Jiang, and Haizhou Li. 2025. Take the essence and discard the dross: A rethinking on data selection for fine-tuning large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6595–6611, Albuquerque, New Mexico. Association for Computational Linguistics.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu Su, and Wenpeng Yin. 2024a. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2024b. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095.

Hyeonseok Moon, Jaehyung Seo, and Heuiseok Lim. 2025. Call for rigor in reporting quality of instruction tuning data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 100–109, Vienna, Austria. Association for Computational Linguistics.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Hanshu Rao, Weisi Liu, Haohan Wang, I-Chan Huang, Zhe He, and Xiaolei Huang. 2026. A scoping review of synthetic data generation by language models in biomedical research and application: Data utility and quality perspectives. *Journal of Healthcare Informatics Research*, pages 1–26.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10083, Miami, Florida, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mayira Sharif, Guangzeng Han, Weisi Liu, and Xiaolei Huang. 2025. Cultivating multidisciplinary research and education on gpu infrastructure for mid-south institutions at the university of memphis: Practice and challenge. *Preprint*, arXiv:2504.14786.

Shuzheng Si, Haozhe Zhao, Gang Chen, Cheng Gao, Yuzhuo Bai, Zhitong Wang, Kaikai An, Kangyang Luo, Chen Qian, Fanchao Qi, Baobao Chang, and Maosong Sun. 2025. Aligning large language models to follow instructions and hallucinate less via effective data filtering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 16469–16488, Vienna, Austria. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Thuy-Trang Vu, Xuanli He, Gholamreza Haffari, and Ehsan Shareghi. 2023. Koala: An index for quantifying overlaps with pre-training corpora. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 90–98, Singapore. Association for Computational Linguistics.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Pang Wei Koh, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2025. NICE data selection for instruction tuning in LLMs with non-differentiable evaluation metric. In *Forty-second International Conference on Machine Learning*.

Liang Wang, Nan Yang, and Furu Wei. 2024a. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Xiaolong Wei, Bo Lu, Xingyu Zhang, Zhejun Zhao, Dongdong Shen, Long Xia, and Dawei Yin. 2025. Igniting creative writing in small language models: Llm-as-a-judge versus multi-agent refined rewards. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17171–17197.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

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. *Preprint*, arXiv:2407.06645.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Dylan Zhang, Qirun Dai, and Hao Peng. 2025. The best instruction-tuning data are those that fit. *Preprint*, arXiv:2502.04194.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

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 Implementation Details

A.1 Hyperparameters

We use $k = 32$ nearest neighbors for semantic retrieval, $K = 5$ clusters for semantic clustering (k-means on ℓ_2-normalized embeddings), and a cosine-similarity threshold $\tau = 0.9$ to enforce diversity during demonstration selection.

All experiments in this paper adopt a unified configuration. We use Llama Factory (Zheng et al., 2024) to fully fine-tune Llama3.1-8B, and Mistral-7B-v0.3. Fine-tuning is performed with DeepSpeed ZeRO-3 for memory optimization and bf16 mixed precision, using input sequences truncated to 2,048 tokens. Each model is trained for three epochs with the AdamW optimizer, an initial learning rate of 1×10^{-5}, a cosine-annealing learning rate schedule with linear warmup at a rate of 0.1, and a total batch size of 64.

The prompt template of the reward model is shown in Figure 4.

A.2 Hardware and Software

We conducted all experiments on a machine equipped with 8x 6000Ada GPUs, 2x EPYC Genoa 9334 CPUs, and 768GB of RAM. The system runs on Linux kernel 5.14.

For the Sentence Transformer library (Reimers and Gurevych, 2019) used in our experiments, we utilize the `stsb-roberta-large` checkpoint as the embedding model.

B Evaluation Details

B.1 Benchmark Evaluation

For evaluation on standard NLP benchmarks, we use the lm-evaluation-harness (Gao et al., 2024), following their default zero-shot settings.

B.2 Pairwise Evaluation

We employ GPT-4.1-mini as a judge for pairwise evaluation using the prompt template shown in Figure 5.

C Efficiency Analysis

To better understand the practical implications of our approach, we analyze the computational efficiency of different data selection methods. Table 6 provides a comprehensive comparison across several dimensions: backward pass requirements, dependency on teacher models, external knowledge requirements, and the number of forward passes needed per sample.

Our analysis reveals significant differences in computational overhead across methods. DEITA incurs the highest cost, requiring 36,000 forward passes to train reward models plus 2 forward passes per sample for scoring. This substantial training overhead makes DEITA particularly expensive for large-scale data selection scenarios. NUGGETS, while avoiding the training cost, requires 2,000 forward passes per sample to evaluate each candidate against its fixed anchor set, resulting in prohibitive computational costs when processing large datasets. In contrast, uncertainty-based methods like SelectIT (15 forward passes per sample) and RECOST (2 forward passes per sample) offer more reasonable computational requirements. However, SelectIT's multi-prompt and multi-LLM evaluation strategy still introduces considerable overhead, while RECOST's dependency on external knowledge bases may limit its applicability in certain domains.

Our method strikes a balance between evaluation precision and computational efficiency. With 16 forward passes per sample (5 probes × 3 evaluations + 1 for the candidate itself), our approach requires significantly fewer computations than NUGGETS while maintaining the benefits of contextual evaluation. Unlike DEITA, our method requires no training phase, and unlike RECOST, it operates without external dependencies. The dynamic nature of our probe selection ensures that computational resources are focused on the most relevant semantic neighbors, maximizing the informativeness of each evaluation. This efficiency analysis demonstrates that our weighted in-context influence framework achieves superior performance gains while maintaining practical computational requirements, making it suitable for real-world data selection scenarios at scale.

```
System Prompt:
You are a helpful assistant. Please identify the complexity score
of the following user query.

User Prompt:
##Query:
{instruction}

##Complexity:
```

Figure 4: Prompt template used to elicit a discrete complexity level (1–6) from the DEITA complexity scorer. At inference, we read the logits for the six numeral tokens and compute the expected score $\sum_{k=1}^{6} k \cdot \mathrm{softmax}(\ell_k)$, where ℓ_k is the logit of token k.

```
System Prompt:
You are a helpful and precise assistant for checking the quality of the answer.

User Prompt:
[Question]
{question}

[The Start of Assistant 1's Answer]
{ans1}
[The End of Assistant 1's Answer]

[The Start of Assistant 2's Answer]
{ans2}
[The End of Assistant 2's Answer]

[System]
We would like to request your feedback on the performance of two AI
assistants in response to the user question displayed above. Please
rate the helpfulness, relevance, accuracy, level of details of their
responses. Each assistant receives an overall score on a scale of 1
to 10, where a higher score indicates better overall performance.
Please first output a single line containing only two values indicating
the scores for Assistant 1 and 2, respectively. The two scores are
separated by a space. In the subsequent line, please provide a
comprehensive explanation of your evaluation, avoiding any potential
bias and ensuring that the order in which the responses were presented
does not affect your judgment.
```

Figure 5: Prompt template used for pairwise evaluation with GPT-4.1-mini as judge.

Method	Backward	Teacher Model	External Knowledge	Number of Forward Passes
IFD	✗	✗	✗	2/sample (with/without instruction)
DEITA	✓	✓	✗	36,000 (reward model training) + 2/sample (model scoring)
NUGGETS	✗	✗	✗	2,000/sample (anchor set evaluation)
SelectIT	✗	✓	✗	15/sample (multi-prompt, multi-LLM)
RECOST	✗	✗	✓	2/sample (with/without external KB)
Ours	✗	✗	✗	16/sample (5 probes × 3 + 1)

Table 6: Computational Efficiency Analysis of Data Selection Methods

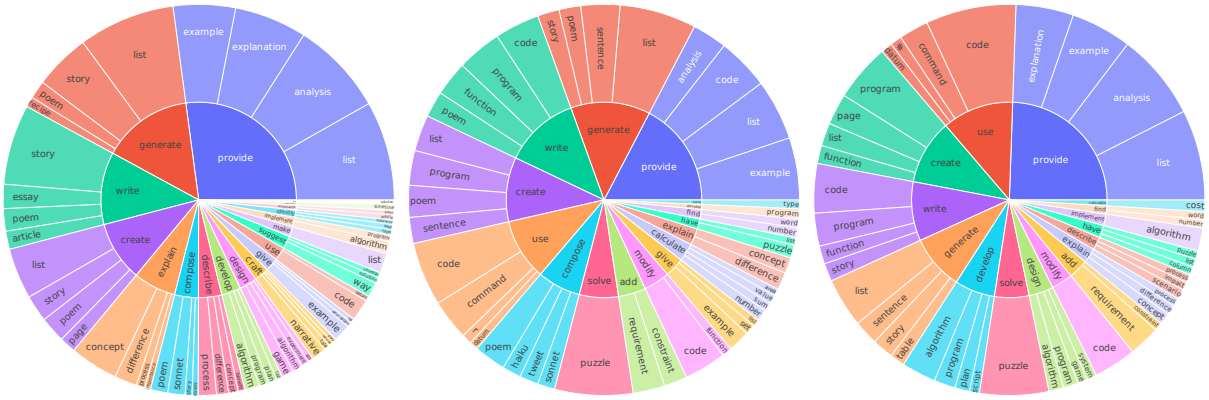
D Additional Results

D.1 Additional Benchmark: IFEval

To further assess whether selected instruction-tuning data improves instruction-following ability beyond general knowledge and reasoning benchmarks, we additionally evaluate all methods on IFEval (Zhou et al., 2023b). IFEval measures how well language models follow *verifiable* instructions, such as output format constraints, keyword requirements, and length constraints, and thus provides a more targeted test of instruction following than broad capability benchmarks. Following the standard evaluation protocol, we report the average over four metrics: prompt-level strict (Pr(S)), prompt-level loose (Pr(L)), instruction-level strict (Ins(S)), and instruction-level loose (Ins(L)). As

Model / Dataset	Full	IFD	DEITA	NUGGETS	SelectIT	Ours
Llama3.1-8B / Alpaca-GPT4	**48.83**	42.31	36.87	38.55	34.11	42.37
Llama3.1-8B / WizardLM	**53.01**	49.60	49.05	50.53	52.11	51.19
Mistral-7B / Alpaca-GPT4	**41.74**	35.19	33.70	32.80	32.77	38.35
Mistral-7B / WizardLM	**50.92**	48.48	48.13	48.15	47.75	48.43

Table 7: IFEval benchmark evaluation. We report the average of four metrics: Pr(S), Pr(L), Ins(S), and Ins(L). The best performance in each row is shown in bold.



(a) High Instruction Following Difficulty. (b) High In-context Influence. (c) Random.

Figure 6: Visualization of the verb–noun structures in instructions selected by the three data-selection strategies on Wizard. The inner ring shows the predominant verbs, the outer ring displays the nouns that co-occur directly with those verbs.

shown in Table 7, IFEval complements our main benchmarks by directly evaluating whether models can precisely comply with explicit user requirements.

The results reveal a clear pattern: in all four settings, the model trained on the full dataset achieves the best performance. Although data selection methods improve performance on knowledge- and answer-quality-oriented benchmarks, these gains do not carry over to stronger instruction-following ability on IFEval. This suggests that instruction following depends more on sufficient data coverage, while being relatively less sensitive to data quality than other capabilities. Among the selection-based methods, our method remains competitive and achieves the best result in the Llama3.1-8B / Alpaca-GPT4 and Mistral-7B / Alpaca-GPT4 settings. Still, the consistent advantage of the full-data baseline indicates that, for instruction-following evaluation, maintaining training scale is more important than selecting a smaller but higher-quality subset. Overall, these findings complement our main results by showing that data selection is especially beneficial for knowledge-intensive and response-quality benchmarks, whereas instruction-following ability appears to benefit more from scale.

D.2 Additional Data Visualization

We provide root verbs and their direct objects from our extracted instruction subset of Wizard70k in Figure 6.