D²S: DUAL-PERSPECTIVE DYNAMIC DATA SELECTION FOR SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective data selection is crucial for supervised fine-tuning under limited training budgets. Existing data selection methods can be broadly categorized into data-centric and model-centric approaches. Data-centric methods rely on reusable static scores but do not directly capture the evolving learning needs of the model, whereas model-centric methods reflect the current learning needs but incur substantial costs when repeatedly evaluating large data pools. To address these issues, we introduce **D**ual-perspective **D**ynamic Data **S**election (D^2S), a framework that first performs coarse selection from the data perspective and then refines the candidates from the model perspective. The resulting model-perspective feedback is used to guide subsequent data-perspective selection, forming an iterative selection process which adapts to the evolving learning needs of the model while restricting expensive model evaluation to a small candidate set. To provide fine-grained model-perspective feedback, we further introduce **M**odel-**a**ware **S**coring (MaS), which distinguishes between important and unimportant response tokens based on the model's prediction probabilities, applies different evaluation criteria to each group, and aggregates the resulting scores into a sample-level score. Experiments on three data pools and two base models show that our framework achieves an average relative improvement of 3.6% over the best baseline and 16.4% over full-data training, while reducing average runtime by 6.1% and 66.2%, respectively.

1 INTRODUCTION

Supervised Fine-Tuning (SFT) is an important approach to improving the instruction-following capabilities and downstream performance of Large Language Models (LLMs). However, with large instruction datasets of varying quality, more training examples do not necessarily yield better performance, while low-quality or noisy examples even hinder learning(Zhou et al., 2023). Effective data selection is therefore essential for improving SFT under a limited training budget.

Existing data selection methods for SFT can be grouped into data-centric and model-centric methods. Data-centric methods assess potential learning value through static attributes of examples and accordingly select training data. They use auxiliary models to score the quality or difficulty of data (Chen et al., 2024; Liu et al., 2024b), or consider diversity to construct representative subsets (Chen et al., 2025; Yin et al., 2024). Model-centric methods guide data selection according to the model's learning needs. They assess sample utility through model predictions or gradients (Li et al., 2024b; Xia et al., 2024; Chen et al., 2026). Some dynamic approaches further update sample priorities during training by model feedback (Yang et al., 2025; Zou et al., 2026).

However, relying on either perspective alone has limitations. Data-perspective attributes can be precomputed and reused but do not directly reflect the model's evolving learning needs. Model-perspective feedback reflects the model's learning needs, but may overvalue noisy or incorrect examples and is costly to repeatedly evaluate the large data pools with the evolving model. These limitations motivate a central question: *How can we integrate the data and model perspectives to balance effectiveness and computational efficiency in data selection?*

To address this challenge, we propose D^2S, a **D**ual-perspective **D**ynamic data **S**election framework that first performs coarse selection from the data perspective and then refines the candidates from the model perspective. From the data perspective, we combine precomputed attribute scores

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

through adaptive weighting to shortlist candidate examples. From the model perspective, we introduce **M**odel-**a**ware **S**coring (MaS), which first distinguishes between important and unimportant response tokens by comparing the model's prediction probabilities with and without the instruction. It then applies informativeness-oriented scoring to important tokens and fluency-oriented scoring to unimportant tokens, combining both into a sample-level score. MaS not only selects final data from candidates for training, but also guides the update of the data-perspective attribute weights, adapting subsequent shortlisting to the model's evolving learning needs while limiting model-perspective evaluation to a small candidate set. Experiments on three datasets and two base models demonstrate the effectiveness and efficiency of our framework. It improves downstream performance by 3.6% over the best data selection method and by 16.4% over full-data training, while reducing average runtime by 6.1% and 66.2%, respectively. In summary, our main contributions are as follows:

- We propose a dual-perspective dynamic data selection framework $\mathrm{D}^2\mathrm{S}$, which connects the data and model perspectives through iterative feedback, adapting selection to the model's evolving learning needs while reducing model-perspective evaluation overhead.
- We introduce a model-aware scoring mechanism MaS, which accounts for differences in token importance to provide fine-grained model-aware assessments of sample utility, supporting both model-perspective selection and data-perspective weights adaptation.
- Experiments on three data pools and two base models show that $\mathrm{D}^2\mathrm{S}$ achieves the highest average benchmark performance. Further ablation studies show that both perspectives are essential for $\mathrm{D}^2\mathrm{S}$, while efficiency analysis demonstrates its reduced computational cost.

2 Preliminary and Related Work

This section formulates the data selection problem for SFT and reviews data-centric and model-centric methods, highlighting their limitations to motivate the design of $\mathrm{D}^2\mathrm{S}$.

Data selection for SFT Given an instruction dataset $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}$, where x_i and y_i denote an instruction and its response, and a model with parameters θ, data selection for SFT seeks an optimal subset $\mathcal{S}^\star(\theta)$ of $b < N$ examples that maximizes the model's downstream performance after SFT:

$$\mathcal{S}^\star(\theta) = \arg\max_{\substack{\mathcal{S}\subseteq\mathcal{D} \\ |\mathcal{S}|=b}} \mathcal{U}\big(\mathcal{A}(\theta, \mathcal{S})\big), \tag{1}$$

where $\mathcal{A}(\theta, \mathcal{S})$ denotes the model parameters obtained by fine-tuning the model on the subset $\mathcal{S}$ under a fixed training configuration, and $\mathcal{U}$ measures downstream task performance.

Data-centric selection. Data-centric methods assess training examples through attributes such as quality, complexity, and diversity (Chen et al., 2024; Liu et al., 2024b). Rethinking (Xia et al., 2025) and SCAR (Li et al., 2025) select examples based on token length and response-style consistency, respectively. InsTag (Lu et al., 2024) uses semantic tags to quantify instruction complexity and diversity. To improve diversity and coverage, MIG (Chen et al., 2025) maximizes semantic information gain, while ZIP (Yin et al., 2024) uses compression-based selection to reduce redundancy. CaR (Ge et al., 2024) combines quality ranking with cluster-based selection, while LASER (Mirza et al., 2025) jointly considers diversity, quality, and difficulty to guide data selection.

Model-centric selection. Model-centric methods assess sample utility through model predictions or gradients (Xia et al., 2024; Chen et al., 2026). IFD (Li et al., 2024b) measures instruction-following difficulty using the ratio of response losses with and without the instruction, while Super-Filtering (Li et al., 2024a) and T-SHIRT (Fu et al., 2025) refine difficulty-based selection through smaller reference models and informative-token scoring, respectively. SelectIT (Liu et al., 2024a) and ADG (Li et al., 2026) use uncertainty-related signals to guide data selection, relying on self-reflection and the geometric dispersion of generated answers. However, these methods adopt a one-time model-based assessment before SFT, which only captures the model's initial learning needs. Some dynamic methods further update sample priorities during training to reflect the model's changing learning needs (Yang et al., 2025; Lin et al., 2025). For example, UDS (Zou et al., 2026) and DSP (Yang et al., 2025) combine model feedback with diversity control to adapt training samples and reduce redundancy, but still incur substantial overhead from repeatedly evaluating data pools.

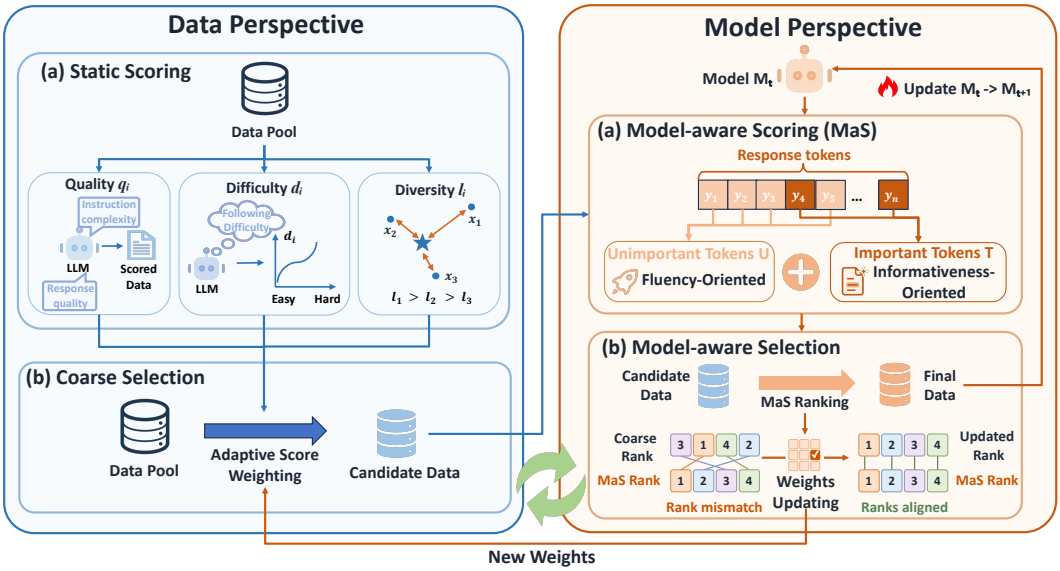


Figure 1: Overview of the proposed D^{2}S framework.

Summarization of related work. In summary, data-centric methods provide reusable assessments of data attributes but do not directly track the model's evolving learning needs, potentially prioritizing high-quality examples whose content the model has already mastered. Model-centric methods assess sample utility through model feedback, with dynamic variants tracking their evolution through repeated evaluation. However, commonly used signals such as loss or uncertainty may also reflect noise or errors rather than useful learning signals. Moreover, repeatedly evaluating large data pools with the evolving model incurs substantial computational overhead.

3 D^{2}S

To address the limited adaptability of static data scoring and the high cost of repeated model-aware evaluation, we propose D^{2}S, a dual-perspective dynamic data selection framework. As depicted in Figure 1, it combines coarse selection from the data perspective with model-aware selection from the model perspective through iterative updates, balancing selection effectiveness and efficiency.

3.1 Data Perspective

In the data perspective, we combine precomputed static quality, difficulty, and diversity scores through adaptive weighting to coarsely select a candidate set $\mathcal{C}_t$.

Static Scoring. We assess static scores of training examples along three complementary dimensions: quality, difficulty, and diversity, consistent with recent work (Hu et al., 2026). Specifically, for each sample z_i, the quality score q_i assesses its reliability and informativeness, the difficulty score d_i measures the challenge of predicting the response given the instruction, and the diversity score l_i captures semantic differences among examples. Together, they form the static scoring vector $\boldsymbol{\phi}_i = [q_i, d_i, l_i]^\top$, with each metric computed as follows.

Quality. The quality score combines response quality and instruction complexity:

$$q_i = Q(z_i) + C(z_i), \tag{2}$$

where Q and C are the response quality and instruction complexity scorers from DEITA (Liu et al., 2024b), respectively.

Difficulty. The difficulty score is computed using the Instruction-Following Difficulty (IFD) (Li et al., 2024b), the ratio of response losses with and without the instruction under a fixed reference

model p_{ref}:

$$d_i = \frac{\ell_{\text{ref}}(y_i \mid x_i)}{\ell_{\text{ref}}(y_i)}, \tag{3}$$

where ℓ_{ref} denotes the mean token-level negative log-likelihood, and a higher difficulty score d_i indicates greater relative instruction-following difficulty.

Diversity. The diversity score measures a sample's distance from the previously selected set, with larger values indicating less semantic redundancy. In practice, the score is computed based on cosine similarity and updated as the selected set $\mathcal{B}_k$ grows during coarse selection within category k:

$$l_i(\mathcal{B}_k) = 1 - \max\left(\{0\} \cup \{\mathbf{z_i}^\top \mathbf{z_j} : z_j \in \mathcal{B}_k\}\right), \tag{4}$$

where $\mathbf{z_i}$ is the ℓ_2-normalized semantic embedding of z_i, and $\mathcal{B}_k$ contains the examples already selected within category k in the current round. Category labels are assigned by a trained LLM classifier f_{cls}, with $k_i = f_{\text{cls}}(z_i)$.

The static scores are normalized to $[0, 1]$ using min–max normalization to ensure comparable scales.

Coarse Selection. Coarse selection aims to construct a small and high-quality candidate set. A global ranking may favor categories with consistently higher attribute scores, leaving other task types overlooked. To preserve category coverage, we first score examples using category-specific adaptive weights, then greedily select candidates within each category's allocated budget.

For scoring, let $\mathcal{D}_k$ denote the examples in category k. Each example $z_i \in \mathcal{D}_k$ receives a weighted attribute score, discounted according to its selection history to discourage repetition:

$$s_t(z_i; \mathcal{B}_k) = \gamma^{n_{i,t}}\, \mathbf{w}_{t,k}^\top [q_i, d_i, l_i(\mathcal{B}_k)]^\top, \qquad \mathbf{w}_{t,k} \geq 0, \quad \mathbf{1}^\top \mathbf{w}_{t,k} = 1, \tag{5}$$

where t denotes the current round, $\mathbf{w}_{t,k}$ denotes the adaptive attribute weights, $n_{i,t}$ counts previous selections of z_i, and $\gamma \in [0, 1]$ is the decay coefficient. For selection, the category proportion $\pi_{t,k}$ determines the candidate budget $m_{t,k}$. Starting from $\mathcal{B}_k = \varnothing$, the highest-scoring example in $\mathcal{D}_k \setminus \mathcal{B}_k$ is repeatedly added until $|\mathcal{B}_k| = m_{t,k}$, with diversity scores updated as $\mathcal{B}_k$ grows. The resulting subsets form the candidate set $\mathcal{C}_t = \bigcup_k \mathcal{B}_k$.

3.2 Model Perspective

Given the candidate set $\mathcal{C}_t$ from coarse selection, the model perspective applies model-aware scoring and selection to obtain the training subset $\mathcal{S}_t$, then uses the resulting scores as feedback to adapt subsequent coarse selection.

Model-aware Scoring (MaS). This component assesses the learning value of candidates based on the model's current learning needs. Since response tokens differ in their importance to model learning (Qin et al., 2026), applying the same evaluation criterion to all tokens may be suboptimal. We therefore propose MaS, which distinguishes important and unimportant response tokens, applies different evaluation criteria to each group, and combines their scores into a sample-level score.

MaS considers strongly instruction-dependent tokens important, as they often correspond to instruction-related answer content, while treating weakly instruction-dependent tokens that may be predictable from the response prefix alone as unimportant. To distinguish these two groups, MaS quantifies instruction dependence using the log-likelihood gain:

$$\Delta_{i,j}^{(t)} = \log \frac{p_{\theta_t}(y_{i,j} \mid x_i, y_{i,<j})}{p_{\theta_t}(y_{i,j} \mid y_{i,<j})}, \tag{6}$$

where θ_t denotes the current model parameters. Tokens whose gains exceed the $(1 - \rho)$ quantile within each response form the important-token set T_i, while the remaining tokens form the unimportant-token set U_i.

For important tokens, MaS favors contents carrying more instruction-related information, while for unimportant tokens, it prioritizes fluent expression consistent with the model's current conditional distribution. Specifically, we propose an informativeness-oriented score based on the mean instruction-induced log-likelihood gain over T_i to evaluate important tokens, and propose a fluency-oriented score based on the mean conditional log-likelihood over U_i to measure unimportant tokens.

These two scores are combined into a sample-level score:

$$\mathrm{MaS}^{(t)}(z_i) = \underbrace{\frac{\lambda}{|T_i|}\sum_{j\in T_i}\Delta_{i,j}^{(t)}}_{\text{Important tokens}} + \underbrace{\frac{1-\lambda}{|U_i|}\sum_{j\in U_i}a_{i,j}^{(t)}}_{\text{Unimportant tokens}}, \tag{7}$$

where $a_{i,j}^{(t)} = \log p_{\theta_t}(y_{i,j} \mid x_i, y_{i,<j})$ is the token's conditional log-probability under the current model and $\lambda \in [0,1]$ balances the two criteria.

Model-aware Selection. Model-aware selection uses MaS to prioritize candidates that match the model's current learning needs. To account for selection history, each candidate's MaS is multiplied by a decay coefficient:

$$\mathcal{S}_t = \mathrm{Select}\Big(\mathcal{C}_t;\, \gamma^{n_{i,t}\sigma_{i,t}}\mathrm{MaS}^{(t)}(z_i),\, b\Big), \tag{8}$$

where $\sigma_{i,t}$ denotes the sign of the MaS, and Select returns the b candidates with the highest scores. The selected subset $\mathcal{S}_t$ is used to fine-tune the current model.

Feedback Update. As the model's learning needs evolve during training, previously coarse selection becomes less suitable for the current model. We use MaS as a proxy for these evolving needs and update the data-perspective weights accordingly.

Specifically, MaS feedback updates both category selection proportions and category-specific attribute weights. For category allocation, we adjust the next round's selection proportions according to the category distribution of selected training examples, while keeping these proportions within predefined lower and upper bounds. For attribute weighting, we perform grid search within each category to maximize ranking agreement between the reweighted data-perspective scores and MaS:

$$\mathbf{w}_{t+1,k} \in \arg\max_{\mathbf{w}\in\mathcal{W}_{\mathrm{grid},k}} r_{\mathrm{S}}\big(\mathbf{s}_{t,k}(\mathbf{w}), \mathbf{s}_{t,k}^{\mathrm{MaS}}\big), \tag{9}$$

where $\mathcal{W}_{\mathrm{grid},k}$ contains the weight combinations evaluated by grid search, $\mathbf{s}_{t,k}(\mathbf{w})$ is the data-perspective scores of candidates in category k under weights $\mathbf{w}$, $\mathbf{s}_{t,k}^{\mathrm{MaS}}$ is their MaS scores in the same candidate set, and the function r_{S} denotes Spearman rank correlation.

The updated proportions and attribute weights guide candidate allocation and ranking in the next round. Model-perspective feedback from a small candidate set guides data-perspective selection across the full pool, while reusing precomputed attribute scores reduces model evaluation overhead.

4 Experiments

To evaluate the effectiveness and efficiency of D^{2}S and understand the impact of its key components and hyperparameters, we conduct a series of experiments. We first present the experimental setup and main results, followed by ablation studies, efficiency analysis, and hyperparameter analysis.

4.1 Experimental Setup

Data Pools. Following ADG (Li et al., 2026), we consider three widely-used instruction-tuning data pools: (i) *Alpaca-GPT4* (52K) (Peng et al., 2023), which pairs Alpaca instructions with GPT-4-generated responses; (ii) *WizardLM* (70K) (Xu et al., 2024), which uses Evol-Instruct to generate increasingly complex instructions; and (iii) *CoT* (100K) (Wei et al., 2022), which contains examples with step-by-step reasoning.

Benchmarks. We evaluate three broad capability categories: reasoning, knowledge, and coding. For reasoning, performance is measured by accuracy on BBH (Suzgun et al., 2023) and GSM8K (Cobbe et al., 2021). For knowledge, we report accuracy on MMLU (Hendrycks et al., 2021) and the MC2 score on TruthfulQA (Lin et al., 2022). For coding, performance is evaluated using pass@1 on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Further evaluation details are provided in Appendix B.

Baselines. The data selection baselines fall into two categories: *data-centric methods*, which select examples based on data attributes, including Rethinking (Xia et al., 2025), SCAR (Li et al., 2025),

MIG (Chen et al., 2025), and ZIP (Yin et al., 2024); and *model-centric methods*, which guide selection using model feedback, including IFD (Li et al., 2024b), SuperFiltering (Li et al., 2024a), SelectIT (Liu et al., 2024a), T-SHIRT (Fu et al., 2025), ADG (Li et al., 2026), and UDS (Zou et al., 2026). We also include random selection and full-data fine-tuning as reference baselines.

Implementation Details. We use LLaMA-3.1-8B (Grattafiori et al., 2024) and Qwen2.5-7B (Yang et al., 2024) as the base models and perform full-parameter fine-tuning using LLaMA-Factory (Zheng et al., 2024). D^{2}S runs three or four iterative selection–training rounds, each shortlisting 4,500 candidates and selecting 3,000 examples for one epoch of training. This requires a smaller training budget than the data selection baselines, which train on 10k examples of each data pool for three epochs. For MaS, the mixing weight $\lambda = 0.7$ balances the informativeness-oriented and fluency-oriented criteria. The important-token ratio is set to $\rho = 0.2$, assigning tokens with gains above the 80th percentile within each response to T_i and the remaining tokens to U_i. The decay coefficient is set to $\gamma = 0.95$. Further implementation details are provided in Appendix C.

4.2 Main Results

Tables 1 and 2 report the results on Alpaca-GPT4, WizardLM, and CoT for the two base models, respectively. We have the following key observations.

D^{2}S outperforms full-data training and all baselines. As shown in Table 1, D^{2}S surpasses full-data training by 23.3% on average across the three data pools with LLaMA-3.1-8B, highlighting the importance of efficient data selection in reducing the impact of noisy or redundant samples. It also outperforms the best data-centric baseline MIG by 22.5% on average. This large gap reflects that D^{2}S better adapts data selection to the model's evolving learning needs. Compared with the best model-centric baseline ADG, D^{2}S achieves an average relative improvement of 4.5%, with gains of 6.1%, 3.2%, and 4.3% on Alpaca-GPT4, WizardLM, and CoT, respectively, suggesting the effectiveness of D^{2}S in filtering out noisy examples that model-centric methods may favor.

D^{2}S generalizes across different data pools and base models. Tables 1 and Table 2 show that D^{2}S achieves the highest average score across three data pools with different sizes and compositions on the two base models, respectively. Specifically, on LLaMA-3.1-8B, even on noisier pools such as CoT, D^{2}S outperforms full-data training and the best baseline by 32.7% and 4.3%, respectively, showing that its effectiveness is not restricted to a particular data composition. Across both base models, D^{2}S consistently surpasses the best baseline and full-data training, with average relative improvements of 4.5% and 23.3% on LLaMA-3.1-8B, and 2.6% and 9.5% on Qwen2.5-7B, respectively, showing that D^{2}S can select effective data without relying on a specific base model.

D^{2}S supports broad capability gains. D^{2}S maintains balanced performance advantages across reasoning, coding, and knowledge benchmarks, except TruthfulQA. As shown in Table 1, D^{2}S averagely improves reasoning, coding, and knowledge scores by 4.5%, 7.2%, and 2.6% over the best baseline ADG, and by 27.3%, 39.2%, and 11.1% over full-data training, respectively. These consistent gains across all three capabilities suggest that D^{2}S selects broadly beneficial training data rather than favoring a particular task category. As an exception, on TruthfulQA, D^{2}S trails SCAR on Alpaca-GPT4 and IFD on CoT by relative drops of 0.5% and 13.1%, respectively. This is because TruthfulQA requires models to recognize and avoid common misconceptions, making examples that provide explicit factual explanations particularly valuable. Although SCAR and IFD may select data better suited to this requirement, their advantages on TruthfulQA are accompanied by weaker reasoning and coding performance. D^{2}S surpasses SCAR on Alpaca-GPT4 and IFD on CoT in average score by 19.7% and 20.5%, respectively. This contrast suggests that D^{2}S selects broadly useful training data rather than favoring examples with benchmark-specific characteristics.

4.3 Ablation Study

We conduct an ablation study to evaluate the contributions of the core components of D^{2}S, i.e., the data and model perspectives. Without loss of generality, the ablation study is performed on Alpaca-GPT4 with LLaMA-3.1-8B, as shown in Table 4. Here, *w/* and *w/o* denote with and without certain module, respectively.

The effectiveness of data perspective. The data perspective consists of static scoring and coarse selection. To assess the value of static scoring, w/ random selection replaces coarse selection with

Table 1: Performance comparison on Alpaca-GPT4, WizardLM, and CoT. The best and second-best results are bolded and underlined, respectively.

Dataset	Method	Reasoning		Knowledge		Coding		Average
		BBH	GSM8K	MMLU	TruthfulQA	HumanEval	MBPP	
Alpaca-GPT4 (52K)	All Data	55.3	36.6	60.4	54.8	32.9	42.8	47.2
	Random	52.4	33.3	60.5	53.7	37.8	38.8	46.1
	Data-centric methods							
	Rethinking	53.1	36.3	60.6	56.0	37.8	38.6	47.1
	SCAR	53.7	42.5	61.4	**58.5**	33.5	38.2	48.0
	MIG	51.9	37.0	60.1	55.4	36.0	40.0	46.7
	ZIP	51.8	37.2	61.2	52.3	37.2	40.6	46.7
	Model-centric methods							
	IFD	40.4	46.0	63.2	56.1	41.5	46.4	48.9
	SuperFiltering	53.9	39.7	60.1	56.8	40.9	45.0	49.4
	SelectIT	50.8	33.7	58.6	53.9	33.5	41.6	45.4
	ADG	63.0	51.6	64.3	55.7	44.5	45.6	54.1
	UDS	62.5	51.5	64.7	57.6	43.9	47.0	54.5
	D^{2}S	**66.4**	**58.6**	**65.5**	58.2	**45.1**	**51.0**	**57.5**
WizardLM (70K)	All Data	56.7	46.5	60.5	47.9	43.3	42.4	49.6
	Random	55.1	41.7	61.5	52.3	38.4	39.2	48.0
	Data-centric methods							
	Rethinking	52.2	37.9	61.2	46.8	34.2	39.2	45.3
	SCAR	52.5	37.5	58.8	49.6	29.9	37.4	44.3
	MIG	55.4	38.5	61.1	50.4	40.9	41.8	48.0
	ZIP	54.4	35.0	61.1	52.3	32.9	40.0	45.9
	Model-centric methods							
	IFD	58.3	42.4	61.6	53.8	39.6	45.4	50.2
	SuperFiltering	58.6	43.6	61.9	53.8	39.0	45.2	50.4
	SelectIT	54.3	35.8	60.6	46.3	33.5	41.8	45.4
	ADG	64.2	55.4	65.3	53.5	44.5	49.6	55.4
	UDS	63.9	54.4	64.4	54.1	45.7	49.8	55.4
	D^{2}S	**65.5**	**57.7**	**65.5**	**54.3**	**47.0**	**53.2**	**57.2**
CoT (100K)	All Data	45.0	47.4	51.9	43.3	22.6	31.2	40.2
	Random	46.4	32.8	53.6	40.4	22.6	36.2	38.7
	Data-centric methods							
	Rethinking	47.7	37.4	57.4	43.1	26.2	39.2	41.8
	SCAR	44.9	36.6	54.8	42.7	21.3	36.8	39.5
	MIG	49.5	36.4	57.1	42.2	31.1	39.0	42.5
	ZIP	50.1	37.1	56.5	46.3	29.3	38.0	42.9
	Model-centric methods							
	IFD	39.3	34.9	60.6	**52.7**	35.4	43.0	44.3
	SuperFiltering	52.6	37.2	59.3	38.8	35.4	41.0	44.0
	SelectIT	41.3	29.7	53.1	37.7	24.4	35.2	36.9
	ADG	61.5	53.8	61.2	44.4	40.9	45.2	51.2
	UDS	53.0	52.5	58.9	44.0	36.0	47.8	48.7
	D^{2}S	**62.8**	**54.2**	**64.1**	45.8	**43.3**	**50.2**	**53.4**

random sampling, keeping the candidate-set size unchanged. To further examine the benefit of coarse selection, w/o coarse selection applies MaS directly to the entire data pool. As shown in Table 4, D^{2}S yields a 2.9% higher average score than w/ random selection, supporting the contribution of static scoring to improve candidate quality. It also outperforms w/o coarse selection by 5.5% while requiring fewer MaS evaluations. This suggests that coarse selection improves both efficiency and effectiveness by excluding noisy examples that MaS may overvalue.

The effectiveness of model perspective. The model perspective includes MaS and feedback updates. The contribution of MaS is examined through w/o MaS, which selects training examples solely using static scores. The effectiveness of dynamic selection is examined using w/ fixed MaS and w/ T-SHIRT. The former keeps the initial MaS scores fixed throughout training, while the latter replaces MaS with the strongest alternative scoring baseline, T-SHIRT (Fu et al., 2025). Besides, w/o feedback disables model feedback to coarse selection to assess its contribution. Among

Table 2: Average scores across six benchmarks for Qwen2.5-7B on three data pools.

Method	Alpaca	WizardLM	CoT
All Data	64.0	63.3	59.4
Random	66.8	64.7	60.1
Data-centric methods			
Rethinking	66.8	64.3	61.1
SCAR	67.1	65.2	60.0
MIG	67.6	66.4	61.1
ZIP	67.9	64.5	60.7
Model-centric methods			
IFD	67.2	64.1	61.8
SuperFiltering	68.1	65.6	61.2
SelectIT	65.5	63.7	60.0
ADG	69.1	67.6	62.7
UDS	68.6	67.6	62.1
D^{2}S	**70.4**	**68.7**	**65.4**

Table 3: Efficiency analysis with runtime in minutes. Preproc. denotes preprocessing.

Method	Preproc.	Training	Total
Alpaca-GPT4 (52K)			
Full Data	0.0	246.7	246.7
ADG	106.8	49.0	155.8
UDS	0.0	83.2	**83.2**
D^{2}S	56.9	27.6	84.5
w/o coarse sel.	0.0	138.4	138.4
WizardLM (70K)			
Full Data	0.0	337.9	337.9
ADG	143.6	51.0	194.6
UDS	0.0	125.8	125.8
D^{2}S	76.6	42.1	**118.7**
w/o coarse sel.	0.0	335.1	335.1
CoT (100K)			
Full Data	0.0	471.9	471.9
ADG	205.4	50.1	255.5
UDS	0.0	176.2	176.2
D^{2}S	109.4	41.7	**151.1**
w/o coarse sel.	0.0	418.8	418.8

the model-perspective ablations in Table 4, w/o MaS produces the largest decline in average score, highlighting the central role of MaS in selecting examples that meet the model's learning needs. The 4.0% advantage of D^{2}S over w/ fixed MaS further supports that updating MaS during training helps select data that better match the model's evolving learning needs. The HumanEval exception may reflect relatively stable learning needs for that particular coding task. Compared with w/ T-SHIRT, the strongest alternative model-aware scoring baseline, D^{2}S achieves a 2.0% higher average score, which supports the effectiveness of MaS within the same selection framework. Finally, D^{2}S outperforms w/o feedback by 1.6%, supporting the role of model feedback in adapting data-perspective weights to the model's evolving learning needs and combining the two perspectives in a closed loop.

Table 4: Ablation study assessing the contributions of the data and model perspectives in D^{2}S on Alpaca-GPT4 with LLaMA-3.1-8B.

Method	BBH	GSM8K	MMLU	TruthfulQA	HumanEval	MBPP	Avg.
D^{2}S	**66.4**	**58.6**	**65.5**	**58.2**	45.1	**51.0**	**57.5**
Ablation studies on the data perspective							
w/ random selection	64.7	56.4	65.3	56.2	45.1	47.4	55.9
w/o coarse selection	63.6	53.6	64.7	55.9	43.9	45.2	54.5
Ablation studies on the model perspective							
w/o MaS	63.7	54.1	64.3	56.9	44.0	46.8	55.0
w/ fixed MaS	63.5	56.9	64.4	52.6	**46.5**	48.2	55.3
w/ T-SHIRT (Fu et al., 2025)	65.0	56.3	65.1	56.2	45.0	50.2	56.3
w/o feedback	64.5	55.7	65.2	57.5	46.3	50.4	56.6

4.4 Efficiency Analysis

This section analyzes the time cost of D^{2}S. Table 3 compares the runtimes of different methods on LLaMA-3.1-8B, all evaluated under the same experimental settings on 8 NVIDIA A100 80GB GPUs. From these results, we can observe that:

D^{2}S reduces total runtime across data pools. D^{2}S reduces total runtime by 66.2% relative to full-data training and by 41.9% relative to the best model-centric baseline ADG on average, demonstrating its computational efficiency. Compared with UDS, D^{2}S is slightly slower on Alpaca-GPT4 but runs faster on the larger WizardLM and CoT data pools, suggesting an efficiency advantage at larger data scales. Furthermore, preprocessing accounts for most of D^{2}S's runtime. The result of preprocessing can be reused across base models on the same data pool, further improving efficiency by avoiding repeated preprocessing.

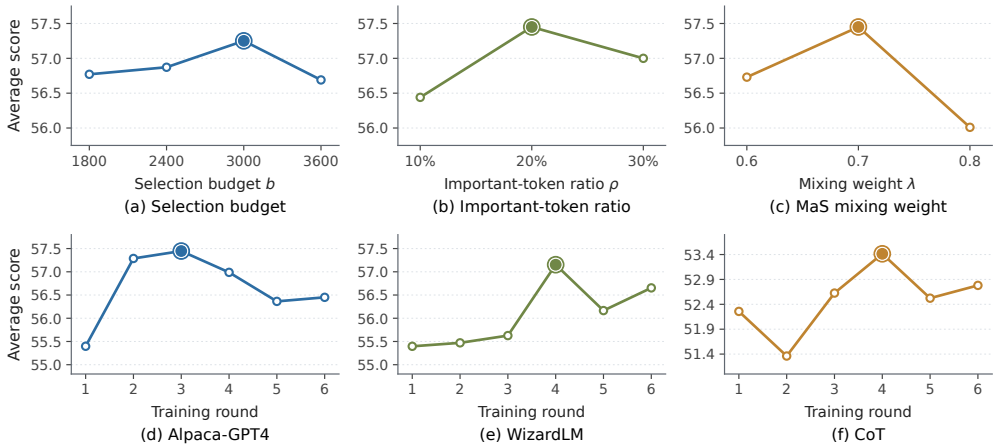


Figure 2: Hyperparameter analysis of D^{2}S.

Coarse selection improves both efficiency and effectiveness. The efficiency analysis further examines whether coarse selection reduces repeated MaS evaluation costs. For this comparison, w/o coarse selection removes coarse selection and applies MaS directly to the entire data pool in every round. As reported in Table 3, this variant incurs 63.8%, 182.3%, and 177.2% more total runtime than D^{2}S on Alpaca-GPT4, WizardLM, and CoT, respectively. Its runtime even approaches that of full-data training. Meanwhile, D^{2}S achieves a 5.5% higher average score on Alpaca-GPT4 from the result in Table 4. These results suggest that coarse selection improves both efficiency and effectiveness by limiting MaS evaluation to a smaller, higher-quality candidate set.

4.5 Hyperparameter Analysis

This section examines the main hyperparameters of D^{2}S: the per-round selection budget b, the important-token ratio ρ, the MaS mixing weight λ, and the number of selection–training rounds. They are studied on Alpaca-GPT4 with LLaMA-3.1-8B, with the selected values subsequently applied to the other data pools and base models.

From Figure 2(a), the selection budget $b = 3000$ achieves the best performance. Smaller budgets may limit the coverage of useful training examples, while increasing the budget to 3600 brings no clear benefit. This is because the additional samples introduce noisy samples, highlighting the importance of a moderate per-round training budget. For MaS, $\rho = 0.2$ and $\lambda = 0.7$ yield the best results (Figure 2(b)–(c)). A smaller ρ may exclude useful instruction-dependent tokens from the important group, whereas a larger ratio may introduce unimportant tokens. The lower scores at $\lambda = 0.6$ and 0.8 further highlight the need for an appropriate mixing weight to balance the contributions of important and unimportant tokens. Figure 2(d)–(f) further shows that the selected runs peak in round three or four, with no consistent gains from subsequent rounds. This is because additional rounds of SFT may lead to catastrophic forgetting, highlighting the importance of choosing an appropriate number of training rounds. Overall, the performance variations of different hyperparameters are limited, demonstrating the robustness of D^{2}S.

5 Conclusion

In this paper, we introduce D^{2}S, a dual-perspective dynamic data selection framework for SFT. From the data perspective, D^{2}S adaptively combines reusable static scores to shortlist candidates. From the model perspective, MaS distinguishes response tokens by importance and applies different scoring criteria to select efficient training examples. The resulting scores also update attribute weights for subsequent rounds, adapting coarse selection to evolving learning needs while limiting model-side evaluation to a small candidate set. Experiments on three data pools and two base models demonstrate consistent performance gains over competitive baselines and full-data training. Ablation studies and efficiency analysis further highlight the contributions of both perspectives.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing and improving the organization and readability of the code prepared for submission. The core ideas, algorithmic design, code implementation, and experiments were developed and carried out by the authors. We have reviewed all AI-assisted work and we take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our work. All datasets and base models used in our experiments are publicly available and cited in the main text. The methodology section describes the scoring, selection, and feedback procedures of D^2S, while the appendix provides detailed training configurations, hyperparameter settings, and evaluation protocols. Our source code, including scripts for data preprocessing, data selection, and model training, is provided in the supplementary materials to facilitate reproduction of the results.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, volume 2024, pp. 34767–34797, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Sirui Chen, Yunzhe Qi, Mengting Ai, Yifan Sun, Ruizhong Qiu, Jiaru Zou, and Jingrui He. Influence-preserving proxies for gradient-based data selection in LLM finetuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PDNpRLxDlI.

Yicheng Chen, Yining Li, Kai Hu, Ma Zerun, Haochen Ye, and Kai Chen. MIG: Automatic data selection for instruction tuning by maximizing information gain in semantic space. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9902–9915, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.515. URL https://aclanthology.org/2025.findings-acl.515/.

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

Yanjun Fu, Faisal Hamman, and Sanghamitra Dutta. T-SHIRT: Token-selective hierarchical data selection for instruction tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=oN5YVZ9JeF.

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. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 464–478, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.28. URL https://aclanthology.org/2024.emnlp-main.28/.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hang Hu, Ziyan Liu, Rujie Wen, Ruihui Hou, Xueyan Wu, Mu Zhang, Jianxing Yu, Tong Ruan, and Jingping Liu. From selection to refinement: Iterative optimization for instruction data. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40670–40684, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1889. URL https://aclanthology.org/2026.acl-long.1889/.

Bo Li, Mingda Wang, Shikun Zhang, and Wei Ye. Instruction data selection via answer divergence. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4684–4702, 2026.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 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)*, pp. 14255–14273, 2024a.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 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)*, pp. 7602–7635, 2024b.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. SCAR: Data selection via style consistency-aware response ranking for efficient instruction-tuning of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12756–12790, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.625. URL https://aclanthology.org/2025.acl-long.625/.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Xiaotian Lin, Yanlin Qi, Yizhang Zhu, Themis Palpanas, Chengliang Chai, Nan Tang, and Yuyu Luo. Lead: iterative data selection for efficient llm instruction tuning. *Proceedings of the VLDB Endowment*, 19(3):426–439, 2025.

Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. *Advances in Neural Information Processing Systems*, 37:97800–97825, 2024a.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*, volume 2024, pp. 22353–22373, 2024b.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 36456–36474, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9dae2a90bae49dc874ce1ca8fcc20879-Paper-Conference.pdf.

Paramita Mirza, Lucas Weber, and Fabian Küch. Stratified selective sampling for instruction tuning with dedicated scoring strategy. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19949–19974, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1086. URL https://aclanthology.org/2025.findings-emnlp.1086/.

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

Xiaohan Qin, Xiaoxing Wang, Ning Liao, Cancheng Zhang, Xiangdong Zhang, Mingquan Feng, Jingzhi Wang, and Junchi Yan. sstoken: Self-modulated and semantic-aware token selection for LLM fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wZ4UoRKDNa.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. Rethinking data selection at scale: Random selection is almost all you need. In *EMNLP (Findings)*, pp. 2698–2711, 2025.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

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. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Yingxuan Yang, Huayi Wang, Muning Wen, Xiaoyun Mo, Qiuying Peng, Jun Wang, and Weinan Zhang. Diversity-aware self-paced data selection for llm fine-tuning. In *ECAI 2025: 28th European Conference on Artificial Intelligence 25-30 October 2025, Bologna, Italy-Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025) Proceedings*, pp. 4330–4337. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2025.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. Entropy law: The story behind data compression and llm performance. *arXiv preprint arXiv:2407.06645*, 2024.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 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)*, pp. 400–410, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Heming Zou, Yixiu Mao, Yun Qu, Qi Wang, and Xiangyang Ji. Utility-diversity aware online batch selection for LLM supervised fine-tuning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=NWZZmXRIH5.

Algorithm 1 D^{2}S: Dual-Perspective Dynamic Data Selection

Require: Data pool $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}$; initial model θ_0; number of rounds R; candidate budget m; training budget b
Require: MaS parameters ρ, λ; decay coefficient $\gamma \in (0, 1]$ and the rule for $\sigma_{i,t}$; weight-search grid $\mathcal{W}$; category lower and upper proportion bounds
Ensure: Checkpoints $\Theta = \{\theta_1, \ldots, \theta_R\}$

Offline preprocessing
Compute and cache normalized quality and difficulty scores q_i, d_i, unit embeddings $\mathbf{e}_i$, and category labels k_i
Partition $\mathcal{D}$ into categories $\{\mathcal{D}_k\}_{k=1}^{K}$
Initialize $\mathbf{w}_{0,k} \leftarrow \mathbf{w}_{\text{init}}$ for each category k
Initialize selection counts $n_{i,0} \leftarrow 0$ for all $z_i \in \mathcal{D}$ and $\Theta \leftarrow \varnothing$
for $t = 0, \ldots, R-1$ **do**
  Allocate integer candidate budgets $\{m_{t,k}\}_{k=1}^{K}$ under the category bounds and available data, with $\sum_{k=1}^{K} m_{t,k} = m$
  $\mathcal{C}_t \leftarrow \varnothing$
  Data perspective: candidate construction
  for each category k **do**
    $\mathcal{B}_k \leftarrow \varnothing$
    while $|\mathcal{B}_k| < m_{t,k}$ **do**
      Update $\boldsymbol{\phi}_i(\mathcal{B}_k) = [q_i, d_i, l_i(\mathcal{B}_k)]^\top$ for remaining examples in $\mathcal{D}_k$
      Select $z_{i^\star} \in \arg\max_{z_i \in \mathcal{D}_k \setminus \mathcal{B}_k} \gamma^{n_{i,t}} \mathbf{w}_{t,k}^\top \boldsymbol{\phi}_i(\mathcal{B}_k)$
      Record $\bar{\boldsymbol{\phi}}_{t,i^\star} \leftarrow \boldsymbol{\phi}_{i^\star}(\mathcal{B}_k)$
      $\mathcal{B}_k \leftarrow \mathcal{B}_k \cup \{z_{i^\star}\}$
    end while
    $\mathcal{C}_{t,k} \leftarrow \mathcal{B}_k$; $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \mathcal{C}_{t,k}$
  end for
  Model perspective: fine-grained MaS
  for each $z_i \in \mathcal{C}_t$ **do**
    Compute response-token log-likelihoods $a_{i,j}$ with the instruction and $u_{i,j}$ without it using θ_t
    $\Delta_{i,j} \leftarrow a_{i,j} - u_{i,j}$ for all response positions j
    $\tau_i \leftarrow \text{Quantile}_{1-\rho}(\{\Delta_{i,j}\}_{j=1}^{L_i})$
    $T_i \leftarrow \{j : \Delta_{i,j} > \tau_i\}$; $U_i \leftarrow \{1, \ldots, L_i\} \setminus T_i$
    $\text{MaS}^{(t)}(z_i) \leftarrow \lambda \, \text{Avg}_{j \in T_i} \Delta_{i,j} + (1-\lambda) \, \text{Avg}_{j \in U_i} a_{i,j}$
  end for
  Model-aware selection
  Compute $\sigma_{i,t}$ according to the decay rule for each $z_i \in \mathcal{C}_t$
  $\mathcal{S}_t \leftarrow \text{Select}\Big(\mathcal{C}_t;\, \gamma^{n_{i,t}\sigma_{i,t}} \text{MaS}^{(t)}(z_i),\, b\Big)$
  Feedback to subsequent selection
  for each category k **do**
    $\pi_{t+1,k} \leftarrow \frac{|\mathcal{S}_t \cap \mathcal{D}_k|}{b}, \qquad \pi_k^{\min} \le \pi_{t+1,k} \le \pi_k^{\max}$
    $\mathbf{w}_{t+1,k} \leftarrow \arg\max_{\mathbf{w} \in \mathcal{W}} \text{Spearman}\Big(\{\mathbf{w}^\top \bar{\boldsymbol{\phi}}_{t,i}\}_{z_i \in \mathcal{C}_{t,k}}, \{\text{MaS}^{(t)}(z_i)\}_{z_i \in \mathcal{C}_{t,k}}\Big)$
  end for
  Model and history updates
  $\theta_{t+1} \leftarrow \mathcal{A}(\theta_t, \mathcal{S}_t)$
  $\Theta \leftarrow \Theta \cup \{\theta_{t+1}\}$
  $n_{i,t+1} \leftarrow n_{i,t} + \mathbf{1}[z_i \in \mathcal{S}_t]$ for all $z_i \in \mathcal{D}$
end for
return Θ

A Algorithmic Details of D^{2}S

Algorithm 1 summarizes D^{2}S. Static data attributes and semantic embeddings are computed once, while candidate construction, model-aware scoring, and feedback updates are repeated as training progresses. Selection history discounts previously used examples, and category quotas control the composition of both the candidate set and the training subset.

The diversity score $l_i(\mathcal{B}_k)$ is recomputed from cached embeddings as $\mathcal{B}_k$ grows. Its value at admission is stored in $\bar{\boldsymbol{\phi}}_{t,i}$ and held fixed during feedback optimization. The operator Select selects b

candidates based on a global ranking of decay-adjusted MaS scores, subject to the previously specified lower and upper bounds on category proportions. Feedback reuses the current candidate scores, while the updated model supplies MaS scores in the next round. All response tokens in selected examples remain supervised during fine-tuning. Detailed algorithm parameter settings are provided in Appendix C.

B Evaluation Details

Benchmark settings. Evaluation uses the LM Evaluation Harness (v0.4.9.2) with the vLLM inference backend. Table 5 summarizes the evaluation sizes, few-shot settings, and metrics. Each benchmark is evaluated on its complete evaluation split.

Table 5: Evaluation settings for the six benchmarks.

Benchmark	Capability	Examples	Shots	Metric
BBH	Reasoning	6,511	3	Exact match
GSM8K	Reasoning	1,319	8	Exact match
MMLU	Knowledge	14,042	4	Accuracy
TruthfulQA	Knowledge	817	0	MC2
HumanEval	Coding	164	0	pass@1
MBPP	Coding	500	1	pass@1

Score aggregation and checkpoint reporting. All benchmark scores are expressed on a scale from 0 to 100. The reported average is the unweighted arithmetic mean of the six benchmark scores, giving equal weight to reasoning, knowledge, and coding. All six scores reported for a run correspond to the same checkpoint, rather than independently selected benchmark-specific maxima.

C Training and Implementation Details

Fine-tuning configuration. D^2S uses full-parameter supervised fine-tuning in LLaMA-Factory (Zheng et al., 2024). The training configuration is summarized in Table 6. An effective batch size of 64 is obtained using n 8 NVIDIA A100 80GB GPUs, a per-device batch size of one, and eight gradient accumulation steps. The `llama3` and `qwen` training templates are used for LLaMA-3.1-8B and Qwen2.5-7B, respectively. Each round starts from the model weights produced by the previous round.

Static attribute computation. Response quality and instruction complexity are assessed using the DEITA scorers (Liu et al., 2024b). LLaMA-3.1-8B serves as the fixed reference model for both instruction-following difficulty computation and semantic embedding extraction, with a maximum input length of 2,048 tokens. Difficulty is computed from response losses with and without the instruction. Semantic embeddings are extracted from the final-layer hidden state at the last response token and ℓ_2-normalized before cosine-similarity computation. For category assignment, Qwen2.5-7B is fine-tuned on 2,000 examples annotated with category labels by Claude Opus 4.6. The resulting classifier assigns each example to one of eight categories: writing, roleplay, extraction, reasoning, math, coding, STEM, and social science. Static scores, embeddings, and predicted category labels are cached and reused across selection rounds.

Model-aware scoring. MaS is computed using the current model in inference mode, with a maximum scoring sequence length of 1,024 tokens. Each candidate is evaluated with and without its instruction, and the resulting log-likelihoods are aligned to the same response tokens before computing their differences. The important-token ratio is $\rho = 0.2$: tokens whose gains exceed the 80th percentile within each response form the important-token group. The two groups are aggregated separately, with an empty group contributing zero. Token partitioning affects sample scoring only; the selected examples retain their complete responses for supervised fine-tuning, subject to the training sequence-length limit. The mixing weight is set to $\lambda = 0.7$ by default.

Table 6: Training settings used by D^2S.

Setting	Value
Fine-tuning strategy	Full-parameter SFT
Number of GPUs	8
Numerical precision	BF16
Distributed training	DeepSpeed ZeRO-3
Effective batch size	64
Per-device batch size	1
Gradient accumulation steps	8
Learning rate	5×10^{-6}
Learning-rate schedule	Cosine decay
Warmup ratio	0.05
Maximum training sequence length	4,096 tokens
Training epochs per selection round	1

Feedback implementation. Attribute weights are updated separately within each category using a grid search over weights that sum to one, with a configured search step of 0.03. Spearman rank correlation between coarse scores and MaS scores provides the ranking objective. Each category is allocated between 5% and 30% of the selection budget to maintain coverage across categories while preventing excessive concentration. The initial category selection proportions are set according to each category's proportion in the data pool. The initial weights for quality, difficulty, and diversity are set to 0.9, 0.1, and 0, respectively. This quality-focused initialization prioritizes high-quality data in the early rounds, providing a more effective starting point than uniform weighting.

Selection and Training Budgets. D^2S selects 3,000 examples per round and trains on the selected subset for one epoch. The process runs for three or four selection–training rounds

. Following ADG (Li et al., 2026), we use a fixed selection budget of 10,000 examples for all data-selection baselines. ADG further validates this choice through a budget-scaling study over 3K, 5K, 10K, 20K, and 25K selected examples, where 10K achieves the highest average performance among the evaluated budgets, with performance improving up to 10K but not consistently benefiting from further increases. Each baseline is trained on its selected subset for three epochs, corresponding to 30,000 sample presentations. D^2S therefore achieves the reported performance with a smaller cumulative training budget.

D Detailed Results on Qwen2.5-7B

The detailed results for Qwen2.5-7B are reported in Table 7. D^2S achieves the highest average scores of 70.4, 68.7, and 65.4 on Alpaca-GPT4, WizardLM, and CoT, respectively. Its advantages are particularly consistent in reasoning, with the highest BBH and GSM8K scores across all three data pools. D^2S also ranks among the top two methods on both coding benchmarks in every data pool. Together with the LLaMA-3.1-8B results, these findings support the effectiveness of D^2S across different base models and data pools.

Table 7: Performance comparison on Alpaca-GPT4, WizardLM, and CoT with Qwen2.5-7B. Avg. Score is the unweighted mean across six benchmarks. The best and second-best results within each data pool are bolded and underlined.

Dataset	Method	Reasoning		Knowledge		Coding		Avg. Score
		BBH	GSM8K	MMLU	TruthfulQA	HumanEval	MBPP	
Alpaca-GPT4 (52K)	All Data	60.2	78.7	71.8	59.0	53.7	60.8	64.0
	Random	62.2	81.2	73.0	60.6	61.6	62.0	66.8
	Data-centric methods							
	Rethinking	64.3	81.6	73.8	59.7	59.2	62.4	66.8
	SCAR	66.9	83.6	73.5	59.4	56.2	63.2	67.1
	MIG	65.9	83.7	73.6	60.7	59.2	62.6	67.6
	ZIP	65.0	80.9	**74.0**	**60.8**	62.0	64.6	67.9
	Model-centric methods							
	IFD	63.2	83.9	73.6	59.2	61.0	62.6	67.2
	SuperFiltering	63.7	82.6	73.3	60.7	61.6	66.4	68.1
	SelectIT	63.5	79.4	73.5	54.2	62.2	60.4	65.5
	ADG	66.6	84.2	**74.0**	60.7	62.8	66.0	69.1
	UDS	58.1	81.5	72.3	58.8	**76.8**	64.4	68.6
	D^{2}S	**67.9**	**86.3**	72.4	58.8	69.7	**67.0**	**70.4**
WizardLM (70K)	All Data	62.9	78.7	72.6	55.4	63.4	46.8	63.3
	Random	63.9	80.7	73.8	54.2	58.5	57.4	64.7
	Data-centric methods							
	Rethinking	65.2	79.8	73.9	55.3	56.1	55.6	64.3
	SCAR	66.3	81.0	73.0	55.9	56.7	58.0	65.2
	MIG	67.0	81.2	73.9	57.5	58.2	60.4	66.4
	ZIP	65.4	76.9	73.5	57.0	56.7	57.8	64.5
	Model-centric methods							
	IFD	65.5	80.9	73.2	58.8	60.4	46.0	64.1
	SuperFiltering	67.0	80.9	73.1	57.3	56.1	59.0	65.6
	SelectIT	65.4	78.2	73.8	54.5	58.5	51.8	63.7
	ADG	67.7	82.5	**74.3**	**59.1**	60.4	62.0	67.6
	UDS	62.0	81.7	72.3	**59.1**	66.0	**64.2**	67.5
	D^{2}S	**68.0**	**83.1**	72.0	58.6	**66.6**	63.8	**68.7**
CoT (100K)	All Data	60.8	69.7	70.5	**48.0**	52.4	55.2	59.4
	Random	62.6	70.1	73.2	43.0	54.3	57.6	60.1
	Data-centric methods							
	Rethinking	63.3	73.3	72.7	46.6	52.4	58.4	61.1
	SCAR	59.8	72.3	73.1	41.4	56.1	57.0	60.0
	MIG	64.1	71.7	73.4	45.1	53.7	58.4	61.1
	ZIP	60.9	72.9	72.5	44.8	57.9	55.4	60.7
	Model-centric methods							
	IFD	62.0	76.4	72.9	45.9	56.1	57.8	61.8
	SuperFiltering	62.5	74.4	72.8	47.1	54.9	55.4	61.2
	SelectIT	62.2	70.7	72.5	45.3	54.3	54.8	60.0
	ADG	64.3	76.8	74.0	44.8	58.5	57.8	62.7
	UDS	57.1	73.0	71.6	45.3	**68.3**	57.6	62.1
	D^{2}S	**64.5**	**78.0**	**75.1**	47.2	67.0	**60.6**	**65.4**