Model Knows What Now to Learn: Learner-Centric SFT Data Selection via Self-Rectify

Anonymous ACL submission

Abstract

LLM supervised fine-tuning (SFT) data have become abundant, mainly driven by synthetic generation. However, sheer scale does not guarantee effective SFT: large datasets often exhibit pathologies such as redundancy, imbalance, and poor learnability, making data selection critical for improving SFT data efficiency and efficacy. Existing selection methods typically rely on static or handcrafted criteria, which can be subjective, biased, and lack transferability. To address this, we propose LERS (**Le**arner Self-**R**ectify **S**election), a learner-centric data selection framework. It enables the learner model, or its compact homologue, to identify learning-worthy samples via self-feedback signals, and dynamically rectify the training subset distribution throughout the learning process. By focusing on the learner's evolving needs rather than static metrics, LERS surfaces suitably challenging and unmastered samples that are otherwise overlooked. Experiments show that LERS boosts data efficiency and efficacy : using only 10% of the data, it matches the SFT performance of $5\times$ randomly sampled data, and yields consistent gains over full-dataset training in multi-source scenarios. Our findings reveal that prioritizing high-utility data that dynamically address the learner's needs is the key to "more with less" in SFT.[1]

1 Introduction

Large language models (LLMs), pretrained and fine-tuned on large-scale corpora (Xu et al., 2024), have achieved strong performance across natural language understanding, question answering, and reasoning tasks (OpenAI et al., 2024; Shao et al., 2024). In supervised fine-tuning, LLM-generated data (Wang et al., 2023; Taori et al., 2023) have become a main source as model capabilities scale, enabling compact models to distill knowledge from state-of-the-art generators and achieve competitive performance (Radosavovic et al., 2018; Yang et al., 2024b; DeepSeek-AI et al., 2025).

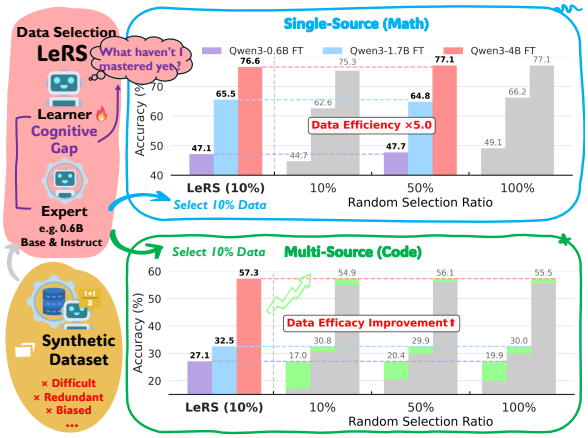


Figure 1: Workflow of LERS, an learner-centric SFT data selection tool. By exploiting the evolving learner state with respect to an expert reference, and rectifying selected data distribution, LERS achieves higher (1) data efficiency: reaching target performance with minimal samples, where LERS matches the performance of $5\times$ random data training and achieves $50\%/10\% = 5\times$ efficiency improvement; (2) data efficacy: the enhancement of available data effectiveness, where 10% LERS-selected data surpasses full-dataset SFT performance.

However, challenges persist in ensuring efficient and effective utilization of synthetic SFT data, which often contains (i) redundancy that inflates training costs; (ii) corrupt samples or imbalanced feature distributions that harm generalization (Yang et al., 2023; Chen et al., 2024). Additionally, a pronounced capability gap between SOTA LLMs and small-scale models creates learning obstacles (Cai et al., 2025). The lack of guarantees on data efficiency and efficacy leads to excessive training overhead and degraded performance, especially burdening individual trainers who rely on paid third-party training services. This highlights the need to identify and select learning-worthy data.

For data selection, efforts largely focused on designing hand-crafted semantic criteria (Penedo et al., 2023; Weber et al., 2024), or assigning static

[1]The code and data will be released as open-source.

quality scores (Chen et al., 2023; Liu et al., 2024a; Li et al., 2024b; Wettig et al., 2024). However, these criteria reflect implicit human assumptions of what benefits model learning (Li et al., 2025), while overlooking the role of model capacity and cognitive state in shaping data utility (Harada et al., 2025). For example, smaller models fail to learn from complex, multi-step reasoning data that is well-suited for SOTA LLMs (Cai et al., 2025). Consequently, these methods introduce inductive biases that generalize poorly across models and datasets. Although model-related perplexity- and gradient-based methods are proposed (Killamsetty et al., 2021; Xia et al., 2024a; Zhang et al., 2025; Morris et al., 2025), in practice, they are restricted to being calculated from the initial learner model, hindering adaptation to its evolving needs in training.

To address these challenges, we select data from a learner-centric perspective, and outline a conceptual data selection workflow in which the learner's current learning state directly informs what data should be presented next. As illustrated in Figure 1, the workflow tracks the gaps between a compact learner and an expert counterpart, indicating which samples surface the learner's unmastered capabilities, enabling the learner to actively rectify its own deficiencies. Notably, compact learners serve as efficient proxies, enabling scalable and transferable signal extraction for larger homologue.

Building on this principle, we propose LERS (**L**earner **S**elf-**R**ectify **S**election), a learner-centric data selection framework that implements the workflow. During learning, LERS tracks a learner-aligned utility metric, termed the **H**omologous **C**orrected **C**ognitive **G**aps (HCCG), which reflects sample-specific discrepancies that reveal the current learner's data-induced learning biases. LERS further partitions the selection timeline into segments, and performs real-time HCCG interpolation to capture such discrepancy changes as the learner evolves. Our theoretical and empirical analyses reveal a self-rectification mechanism: when the selected subset or the learner deviates from the ideal state, samples that reduce the learner's cognitive gap receive higher HCCG scores, implicitly steering selection back toward the desired distribution. Across math and code domains and multiple models, LERS-selected subsets consistently yield superior fine-tuning performance.

Our contributions are summarized as follows:

- We introduce a learner-centric data selection workflow that leverages compact learners as efficient proxies to identify learning-worthy samples for larger homologous models.
- We propose a learner-aligned utility metric, termed HCCG, to evaluate sample learning value at intermediate training stages, and build LERS, a learner-centric SFT data selection framework self-rectifying selected training set throughout learning.
- Extensive experiments show that LERS surpasses all data selection baselines, achieving $5\times$ data efficiency on single-source datasets and improving data efficacy on multi-source datasets by up to $+10.1\%$ over full-data fine-tuning, demonstrating that learner-aligned data utility outweigh sheer volume in SFT.

2 Related Work

2.1 Synthetic Data Generation

Synthetic data generation involves artificially generating training samples using automatic systems rather than manually collecting (Nadăș et al., 2025). Works leverage LLMs to synthesize data for diverse NLP and programming tasks. Synthetic corpora are widely used for instruction tuning (Liu et al., 2022; Honovich et al., 2023; Taori et al., 2023; Bian et al., 2023), reasoning (Zeng et al., 2024), and dialog generation (Zhang et al., 2024). In domain-specific tasks such as programming, works (Zheng et al., 2025; Sun et al., 2025) confirmed that synthetic coding data can substantially improve reasoning capability. Recent advances in DeepSeek-R1 distillation have shown effectiveness in improving reasoning capabilities via synthetic data (DeepSeek-AI et al., 2025), which led to the creation of open synthetic reasoning datasets, including OpenThoughts-114k (Guha et al., 2025) and OpenR1-Math-220k (Hugging Face, 2025).

Although preliminary filtering has been applied to synthetic datasets in terms of correctness and style, learner-centric selection is still required to enhance data utility and learnability.

2.2 SFT Data Selection

Studies indicate that improving data quality drives substantial gains in SFT (Penedo et al., 2023; Zhou et al., 2023; Ye et al., 2025). Data selection aims at identifying the optimal partitions of the dataset to improve SFT efficiency and performance. Prior works mainly concentrate on three directions: (i) Quality or Diversity Assessment and Ranking.

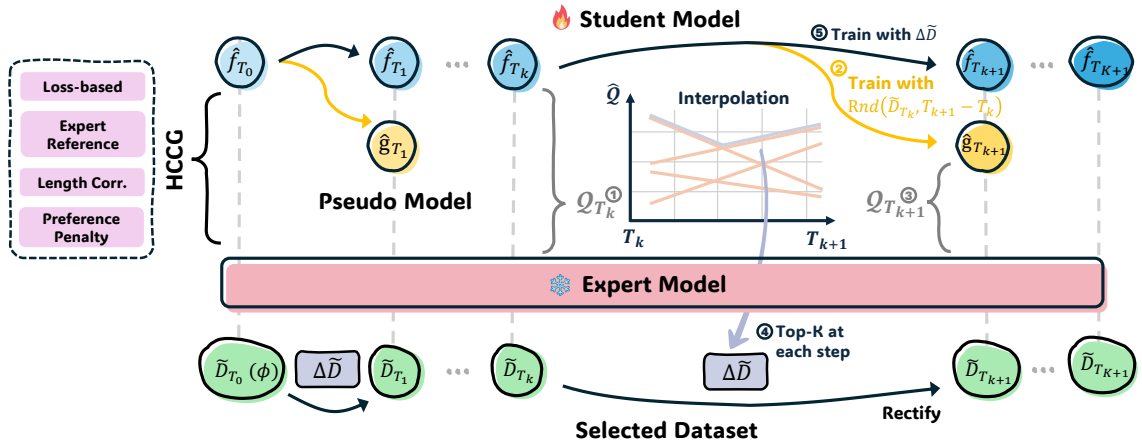


Figure 2: Overview of the proposed Learner Self-Rectify Selection, illustrated using the segment from T_k to T_{k+1}. It incrementally adds supplementary samples $\Delta\tilde{D}$ for learning rectification and outputs the selected dataset $\tilde{D}_{T_{K+1}}$.

These studies design static quality metrics to evaluate instruction complexity (Liu et al., 2024a; Zhao et al., 2024; Chen et al., 2023) and model perplexity (Li et al., 2024a,b), thereby enhancing data efficiency for instruction tuning. (ii) Dataset Coverage. To capture diverse features in data distribution, techniques such as clustering (Ge et al., 2024), embedding distance (Liu et al., 2024b), and subset compression ratio (Yin et al., 2024) are employed. (iii) Gradient and Trajectory Similarity. These approaches select samples aligned with the gradient optimization direction, such as matching gradients of validation subsets (Xia et al., 2024b) or expert trajectories (Morris et al., 2025).

However, these methods mainly rely on data-centric heuristics, which exhibit limited transferability , neglect the learner's evolving data needs, and induce biases inherent in scoring mechanisms.

3 Method

LERS implements learner-centric data selection via a segment-wise training set rectification mechanism, guided by data utility with respect to the current learner. At each segment, the framework measures the discrepancies between the learner and an expert reference, thereby surfacing unmastered samples that target these gaps and alleviating biases induced by training the learner on the data subset. This self-rectification process progressively steers the evolving data distribution toward a more coherent and well-structured learning manifold.

3.1 Subset Bias in Half-Way Learning States

Specifically, we seek to determine: *which remaining samples are the most useful for data subset rectification and learning enhancement currently?*

Definitions. Let the entire dataset be denoted by $\mathcal{D}$, the loss function by $\mathcal{L}$, the input by $X \in \mathbb{R}^n$, and the corresponding ground truth by $Y \in \mathbb{R}^m$. The model defines a mapping from $\mathbb{R}^n \to \mathbb{R}^m$.

Consider an intermediate selection state where a subset $\tilde{\mathcal{D}} \subset \mathcal{D}$ has already been chosen and used in model training. We assume that $\mathcal{D}$ comprises k disjoint feature-consistent categories, such that

$$\mathcal{D} \setminus \tilde{\mathcal{D}} = \bigcup_{i=1}^{k} \mathcal{D}_i, \quad \text{s.t.} \quad \mathcal{D}_i \cap \mathcal{D}_j = \varnothing, \forall i \neq j.$$

After training on subset $\tilde{\mathcal{D}}$, the resulting model $\hat{f}$ is an empirical approximation of the ideal predictor. Let f denotes the bayes-optimal model under the empirical data distribution $P_{\tilde{\mathcal{D}}}$:

$$f = \arg\min_{\hat{f}} \mathbb{E}_{(X,Y)\sim P_{\tilde{\mathcal{D}}}} \mathcal{L}(\hat{f}(X), Y). \quad (1)$$

For analytical tractability, we initially formulate our derivation using Mean Squared Error (noting that an analogous bias–variance decomposition holds for cross-entropy, see Appendix A.1) to show that loss of a sample on the current model is tied to the bias of the selected dataset relative to that sample, which yields a learner-centric utility metric.

Modeling Predictions and Labels. Because the loss decomposes coordinate-wise, we simplify it to the scalar prediction case ($m = 1$) for clarity.

For any training sample $(x, y) \notin \tilde{\mathcal{D}}$, a model trained on biased subset $\tilde{\mathcal{D}}$ injects an additive prediction bias term $\varepsilon_{\text{bias}} \sim \mathcal{N}(\mu, \sigma_b^2)$, while the optimal prediction $f(x)$ deviates from the true label y due to inherent noise $\varepsilon_{\text{noise}} \sim \mathcal{N}(0, \sigma_n^2)$. Thus,

$$y = f(x) + \varepsilon_{\text{noise}} + \varepsilon_{\text{bias}} = f(x) + \varepsilon, \quad (2)$$

where $\varepsilon \sim \mathcal{N}(\mu, \sigma_b^2 + \sigma_n^2) = \mathcal{N}(\mu, \sigma^2)$.

Expected Loss Decomposition. The expected mean squared error can be expanded as

$$\begin{aligned}\mathbb{E}[(\hat{f}-y)^2] &= \mathbb{E}[(\hat{f}-f-\varepsilon)^2] \\ &= \mathbb{E}[(\hat{f}-f)^2] + \mathbb{E}[\varepsilon^2] - 2\,\mathbb{E}[\varepsilon(\hat{f}-f)] \\ &= \mathbb{E}[(\hat{f}-f)^2] + \sigma^2 + \mu^2 - 2\,\mathbb{E}[\varepsilon(\hat{f}-f)]\end{aligned} \tag{3}$$

The first term represents the empirical deviation between model $\hat{f}$ and f, which typically vanishes as training converges. To approximate the last correlation term, we follow Ghojogh and Crowley (2023) and invoke *Stein's lemma* (Stein, 1981).

Lemma 3.1 (Stein, 1981) *If $z \sim \mathcal{N}(\mu, \sigma^2)$ and $g : \mathbb{R} \to \mathbb{R}$ are differentiable, then*

$$\mathbb{E}[(z-\mu)g(z)] = \sigma^2\mathbb{E}[g'(z)]. \tag{4}$$

Employing Stein's lemma, by identifying $\varepsilon = z$ and $g(\varepsilon) = \hat{f}(x,y) - f(x) = \hat{f}(x, f(x)+\varepsilon) - f(x)$, results in the following:

$$\begin{aligned}\mathbb{E}[\varepsilon(\hat{f}-f)] &= \mathbb{E}[(\varepsilon-\mu)(\hat{f}-f)] + \mu\,\mathbb{E}[\hat{f}-f] \\ &= \sigma^2\mathbb{E}\left[\frac{\partial(\hat{f}-f)}{\partial\varepsilon}\right] + \mu\,\mathbb{E}[\hat{f}-f] \\ &= \sigma^2\mathbb{E}[\hat{f}'(y)] + \mu\,\mathbb{E}[\hat{f}-f]\end{aligned} \tag{5}$$

The last term is neglected because the expectation is anticipated to become sufficiently small as training progresses. Substituting Eq. 3 yields:

$$\mathrm{MSE}(\hat{f}) \approx \mathbb{E}[(\hat{f}-f)^2] + \sigma^2 + \mu^2 - 2\sigma^2\mathbb{E}[\hat{f}'(y)] \tag{6}$$

Category-wise Analysis. Applying Monte Carlo estimation within category $\mathcal{D}_t$, we obtain:

$$\begin{aligned}\mathrm{MSE}_t(\hat{f}) \approx \frac{1}{|\mathcal{D}_t|}\sum_i (\hat{f}_i - f_i)^2 + \sigma_t^2 \\ +\mu_t^2 - \frac{2\sigma_t^2}{|\mathcal{D}_t|}\sum_{i\in\mathcal{D}_t}\frac{\partial\hat{f}}{\partial y_i}.\end{aligned} \tag{7}$$

In the most extreme case $\hat{f}(x) \equiv y$, the term reduces to $\mathbb{E}[(\hat{f}-f)^2] = \sigma_t^2$, $\frac{\partial\hat{f}}{\partial y_i} = 1$ and derives $\mathrm{MSE}_t(\hat{f}) \approx \mu_t^2$. Therefore, in general situations, we have $\mathrm{MSE}_t(\hat{f}) - \mu_t^2 \approx k\sigma_t^2, k > 0$, implying

$$\mathrm{MSE}_t(\hat{f}) \approx k\sigma_t^2 + \mu_t^2, \qquad k > 0. \tag{8}$$

Implications. This shows that, within each category, higher systematic bias μ_t (capturing distributional deviation between $\tilde{\mathcal{D}}, \mathcal{D}_t$) and a greater diversity σ_t jointly result in an elevated loss. The bias of $\tilde{\mathcal{D}}$ shifts over training, and the high utility samples are those capable of mitigating it—i.e., those from categories with the largest μ_t and σ_t.

Although the interpretation for cross-entropy is less direct (Appendix A.1), the core correlation persists: a higher loss $\mathcal{L}(x)$ identifies samples that are both informative and diverse, making them ideal targets for subset distributional rectification.

3.2 Homologous Corrected Cognitive Gaps

The preceding derivation establishes a theoretical link between the loss magnitude on the learner model and the utility of subset rectification. Yet, the challenge stems from the presence of low-quality outliers, which exhibit substantial divergence with respect to the dataset $\tilde{\mathcal{D}}$ but should be excluded.

To mitigate this issue, we incorporate the loss from an expert model as an auxiliary anchoring signal. Fine-tuning compact models on high-quality prior data can serve this purpose effectively. However, INSTRUCT models, which are typically released alongside their BASE counterparts, generally provide more robust downstream performance, positioning them as natural experts.

We introduce a metric for learner-aligned training utility, termed ***H**omologous **C**orrected **C**ognitive **G**aps* (HCCG). Given the base model f_b and a fixed expert model f_e as anchor, the HCCG score $\mathcal{Q}$ for a sample (x, y) is defined as:

$$\mathcal{Q}(x,y) = \ln|y|\big(\mathcal{L}(f_b,x,y) - \alpha\mathcal{L}(f_e,x,y)\big),\ \alpha \geq 1 \tag{9}$$

The metric penalizes samples that confound the expert, capturing cognitive discrepancies between the learner and the expert model. The formulation also includes two critical corrections:

Length Correction The standard cross-entropy loss is averaged over tokens, which often leads to higher variance in shorter sequences. To compensate for the effect, we introduce correction term $\ln|y|$, that neutralizes length-induced distortions in the loss distribution.

Learner Preference Penalty The preference penalty $\alpha \geq 1$ penalizes samples that diverge from the preferences of the expert model, which otherwise tend to impede effective learning. This penalty is motivated by an intriguing observation: *synthetic data generated by a homogeneous, smaller-scale expert model yields greater improvements in compact learner performance than data crafted by a much stronger but heterogeneous model*, as illustrated in Section 4.4. This underscores that aligning data selection with the learner's native distribution is more pivotal than adhering to seemingly objective metrics, such as absolute correctness.

The HCCG metric is also computationally efficient, as using vLLM (Kwon et al., 2023) accelerates loss computation without sacrificing precision.

3.3 Segment-wise Subset Rectification

LERS conditions select decisions on the evolving state of the learner rather than anchoring to the initial model or a static, potentially skewed yardstick. We introduce a segment-wise rectification scheme: the selection timeline is divided into segments that progressively rectify the previously selected subset. For each segment, LERS constructs pseudo-checkpoints and interpolates the HCCG, dynamically approximating learning utility.

Specifically, LERS approximates the evolution of the HCCG score $\mathcal{Q}_t(x, y)$ via piecewise linear interpolation. It uniformly partitions the entire selection horizon of $\tilde{N}$ samples using K dividing points $T_{k\in\{1,2,\ldots,K\}}$, and computes $\mathcal{Q}_t$ at segment boundaries. Within the segment interval $[T_k, T_{k+1}]$, the estimated HCCG $\hat{\mathcal{Q}}_t$ at time step t is as follows:

$$T_k = k\frac{\tilde{N}}{K+1}, \quad k = 0, \ldots, K+1, \tag{10}$$

$$\hat{\mathcal{Q}}_t = \mathcal{Q}_{T_k} + \frac{t - T_k}{T_{k+1} - T_k}\left(\mathcal{Q}_{T_{k+1}} - \mathcal{Q}_{T_k}\right). \tag{11}$$

Drawing upon the power laws in data scaling (Hoffmann et al., 2022; Luo et al., 2025) and assuming the influence of the learning rate schedule is marginal, the training loss trajectory can be modeled as:

$$\mathcal{L}_T \approx \mathcal{L}_0 + \frac{B}{T^\beta} \tag{12}$$

where $\mathcal{L}_0$, B, and β are constant coefficients.

By applying the Mean Value Theorem, the interpolation error can be uniformly bounded as:

$$\max_{t\in[T_k,T_{k+1}]} \frac{|\hat{\mathcal{Q}}_t - \mathcal{Q}_t|}{\ln|y_t|} \leq B\frac{\beta(\beta+1)}{8K^2T_k^\beta} \tag{13}$$

Thus, the absolute interpolation error is uniformly controlled by K, decaying at a rate of $O(K^{-2})$.

In practice, at step T_k, the model $\hat{f}_{T_k}$ has been trained on subset $\tilde{\mathcal{D}}_{T_k}$. To estimate the HCCG trajectory for the subsequent interval $(T_k, T_{k+1}]$, we further train a pseudo model $\hat{g}_{T_{k+1}}$ inheriting from $\hat{f}_{T_k}$ and training on $T_{k+1} - T_k$ samples randomly drawn from the selected dataset $\tilde{\mathcal{D}}_{T_k}$. HCCG is evaluated at the endpoints T_k, T_{k+1} with the model $\hat{f}_{T_k}$ and $\hat{g}_{T_{k+1}}$, respectively, and linearly interpolated across intermediate training steps. At each step, the top-`batch_size` samples are selected and removed from the candidate pool. The general selection procedure is summarized in Algorithm 1.

4 Experiment

4.1 Experimental Setup

Data. We use DEEPSEEK-R1 to generate a large-scale reasoning dataset with <THINK> labels from pools of mathematical and programming prompts. We perform preliminary cleaning and retain samples with lengths less than 3,500 tokens, resulting in 100K candidate datasets. They consist of: (1) Math: contains verifiable AoPS (Mahdavi et al., 2025) problems, exhibiting relatively homogeneous sources. (2) Code: a programming dataset composed of diverse sources and prompt types drawn from KODCODE (Xu et al., 2025).

Model. We employ open-source QWEN3 family (Yang et al., 2025a), instantiating the learner model f_b and the expert model f_e in LERS with the 0.6 B BASE and 0.6 B INSTRUCT checkpoints, respectively. After selection, QWEN3 0.6B, 1.7B, and 4B BASE are fine-tuned with the selected data.

Metric. The performance of the post-trained model directly manifests the effectiveness of selected datasets. Mathematical reasoning capability is evaluated on MATH-500, GSM8K, SVAMP, and related benchmarks, while code reasoning capability is assessed on HumanEval (+), MBPP (+), and LiveCodeBench. All results are reported using accuracy (ACC). See Appendix A.3.2 for details.

Baselines. Early studies focused predominantly on crafting static filtering metrics, assigning fixed quality scores and integrating dataset diversity, from which we select one representative method per category serving as baselines.

- **Deita** (Liu et al., 2024b) automatically selects high-value instruction samples by jointly maximizing evolved complexity and quality scores while enforcing embedding-based diversity.
- **Superfiltering** (Li et al., 2024a) employs a small model to compute instruction-following difficulty (IFD) and select top-ranked data.
- **CaR** (Ge et al., 2024) ranks instruction pairs with a lightweight scorer and clusters them to obtain high-quality and diverse subset.

We also benchmark the learner-centric, expert-anchored twin-model selection workflow, using spectrum of static filtering metrics as baselines, including loss-difference ($\Delta\mathcal{L}$) (Cao et al., 2024), KL divergence (KL), gradient-cosine similarity (GCS)

Table 1: Overall Downstream Fine-Tuning Performance of Selected Datasets on QWEN3-0.6B BASE. Each dataset consists of 10K (10%) samples filtered from an synthetic 100K data pool in mathematics or programming. CM.BaI stands for the Beginning_and_Intermediate_Algebra split of CollegeMath. For HumanEval and MBPP, we report both the original and "+" (augmented test case) versions. All results are presented as accuracy scores.

Method		MATH								CODE		
		MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.	HumanEval (+)	MBPP (+)	Avg (+).
RANDOM	10%	37.0	63.1	79.8	8.1	51.2	35.6	38.3	44.7	12.8 (11.6)	21.1 (18.8)	17.0 (15.2)
	50%	42.4	68.2	83.8	9.2	52.0	38.7	39.5	47.7	15.9 (13.4)	24.8 (21.6)	20.4 (17.5)
	100%	41.6	68.8	84.3	8.8	56.7	41.6	41.9	49.1	14.6 (13.4)	25.2 (22.4)	19.9 (17.9)
UNITARY	Deita	38.2	62.5	77.1	8.8	50.4	37.1	35.5	44.2	15.9 (13.4)	28.1 (23.8)	22.0 (18.6)
	CaR	37.4	63.2	79.2	6.6	44.9	36.9	39.5	44.0	18.3 (15.9)	28.1 (25.1)	23.2 (20.5)
	SupF	40.0	61.5	69.9	9.2	51.6	**39.2**	36.4	44.0	12.2 (10.4)	18.3 (16.0)	15.3 (13.2)
TWINED	$\Delta\mathcal{L}$	36.2	64.4	**83.8**	7.4	53.9	37.1	36.3	45.6	17.1 (16.5)	25.6 (23.3)	21.4 (19.9)
	KL	35.0	61.9	82.0	9.9	49.2	34.3	35.9	44.0	12.2 (11.0)	22.8 (18.8)	17.5 (14.9)
	GCS	39.0	**64.7**	80.3	10.7	49.6	36.4	38.6	45.6	17.1 (15.9)	23.1 (20.8)	20.1 (18.4)
	GP	37.6	64.6	80.5	9.2	47.2	36.6	**40.6**	45.2	12.8 (11.0)	24.1 (21.3)	18.5 (16.2)
	LERS	**40.8**	**64.7**	77.5	**12.1**	**57.9**	38.2	38.7	**47.1**	**23.8** (**20.7**)	**30.3** (**25.1**)	**27.1** (**22.9**)

(Morris et al., 2025), and gradient projection (GP), from which $\tilde{N}$ instances with the highest scores are retained (Appendix A.3.1).

Implementation. We select $\tilde{N} = 10K$ samples from the synthetic dataset of size $N = 100$K. For training QWEN3-0.6B, 1.7B, and 4B BASE, we conduct a grid search over learning rates $\{7e-5, 5e-5, 3e-5, 1e-5\}$ and adopt $7e-5$, $3e-5$, and $1e-5$ for each model respectively, with a total batch size of 16, weight decay of 0.01 and warm-up ratio of 0.05 in a cosine scheduler. We fix the random seed for all runs. In LERS, number of segment dividing points is set to $K = 8$. For preference penalty, we use $\alpha = 1.5$ for mathematical tasks and $\alpha = 1.0$ for code tasks, based on the fact that QWEN3-0.6B INSTRUCT demonstrates stronger expertise in mathematics than in programming.

4.2 Main Results

General Comparison Table 1 reports supervised fine-tuning results on QWEN3-0.6B BASE, across the 10K selected subsets of various strategies.

On the code data pool, a heterogeneous collection encompassing tasks such as Code Completion, Codeforces, LeetCode, Prefill and other sources, LERS demonstrates a dominant advantage, achieving up to a +10.1% improvement over random selection and +3.9% over the best baseline.

In the mathematics domain, which is composed exclusively of synthetic verifiable AoPS problems, the limited categorical diversity renders selection intrinsically harder than for typical instruction or code mixture corpora. Nevertheless, LERS still achieves outstanding performance, surpassing the best baseline by +1.5% on average. The results highlight the superior generalizability of LERS across datasets with varying features.

Unitary scorer yields marginal gains over random on multi-source pools yet collapses on more homogeneous dataset, revealing its fixation on surface structure at the expense of learner utility. Twined selection, leveraging learner-expert model pairs, consistently outperforms unitary baselines on single-source data, but its static inductive bias curbs generalization. Details are in Appendix A.2.

Generalization Across Model Scales. Table 2 evaluates the transferability of LERS-selected datasets to larger homologous models. The empirical results demonstrate that the performance gains from LERS scale consistently, outperforming random selection by an average of 2.3% on QWEN3-1.7B and 1.85% on 4B variant. This suggests that data utility demonstrates transferability across nearby model scales, and that the high-utility data remain stable within the size range. The gains also hold for training heterogeneous models with similar scales of the learner (Appendix A.3.4). LERS achieves strong performance on easier tasks and substantial gains on challenging benchmarks (e.g., MATH-500, MINERVA_MATH, GAOKAO, CM.BaI, LiveCodeBench), effectively identifies suitable challenging samples on the fly.

Data Efficiency and Efficacy To quantify the data efficiency and efficacy gains conferred by LERS, we trained baseline models on 10%, 50%, and 100% of the available data, and compared their fine-tuning performance against LERS using only 10% of the samples. The results are illustrated in Figure 1 and Table 9. On the mathematics corpus, where prompts exhibit higher similarity, LERS consistently achieves a $5\times$ improvement

Table 2: Transferability of Data Selected by LERS across Model Scales. The selected datasets are evaluated in fine-tuning of QWEN3-1.7B BASE and 4B BASE. LCBv1 denotes the LiveCodeBench v1 version.

Model	Method	MATH								CODE			
		MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.	HumanEval (+)	MBPP (+)	LCBv1	Avg (+).
QWEN3-1.7B BASE	Random	61.2	82.7	92.5	20.6	70.1	54.5	56.6	62.6	37.2 (32.9)	45.9 (41.1)	**9.3**	30.8 (27.8)
	Deita	64.0	83.5	**93.6**	21.0	73.2	**57.1**	56.2	64.1	40.9 (36.0)	51.4 (44.0)	8.0	**33.4 (29.3)**
	CaR	62.0	82.8	92.3	19.5	75.2	51.2	56.0	62.7	**42.1 (36.6)**	**52.4** (44.4)	4.0	32.8 (28.3)
	$\Delta\mathcal{L}$	54.0	83.2	92.5	**25.0**	74.4	49.4	52.5	61.6	32.9 (28.0)	42.9 (37.6)	2.8	26.2 (22.8)
	GCS	64.6	84.8	**93.6**	23.2	74.4	54.8	56.6	64.6	40.2 (32.3)	52.1 (**44.9**)	0.0	30.8 (25.7)
	LERS	**66.2**	**85.1**	92.4	22.8	**76.8**	56.1	**59.2**	**65.5**	40.2 (**36.6**)	50.4 (43.6)	7.0	32.5 (29.1)
QWEN3-4B BASE	Random	82.0	93.4	95.7	36.8	78.3	73.2	**67.8**	75.3	64.6 (59.8)	69.4 (59.1)	30.8	54.9 (49.9)
	Deita	85.0	**93.9**	95.7	41.2	77.2	74.8	66.4	76.3	65.9 (62.8)	69.4 (59.6)	33.0	56.1 (51.8)
	CaR	81.4	92.9	95.8	37.5	69.3	73.8	65.8	73.8	62.2 (58.5)	69.2 (59.4)	23.8	51.7 (47.2)
	$\Delta\mathcal{L}$	79.2	92.3	**96.0**	37.5	**81.5**	68.6	64.4	74.2	65.9 (60.4)	63.9 (55.1)	37.0	55.6 (50.8)
	GCS	83.4	92.3	94.7	38.2	76.8	71.9	66.9	74.9	65.9 (61.0)	**71.2 (60.9)**	30.5	55.8 (50.8)
	LERS	**85.6**	93.4	95.6	**42.6**	75.6	**75.8**	67.3	**76.6**	**66.5 (62.8)**	68.2 (58.9)	**37.3**	**57.3 (53.0)**

in data efficiency, reaching the level obtained by 50% data with one-fifth of the cost. Within the heterogeneous, multi-source code dataset, simply enlarging the training pool yields negligible gains, whereas LERS-selected samples outperform the model trained on the entire set. The result shows that LERS not only substantially enhances data efficiency but also effectively filters out samples that are uninformative or even detrimental to training.

4.3 Ablation Study

To analyze and disentangle the contributions of the components in LERS, we systematically ablate LERS on QWEN3-0.6B, where the wide capacity gap between the learner and the data generator accentuates the impact. Results are shown in Table 3.

Table 3: Ablation Study of LERS on QWEN3-0.6B

Benchmark	LERS	w/o Length Corr.	w/o Segments	w/o Est.	w/o Expert
MATH-500	**40.8**	37.0	39.8	40.2	29.2
GSM8K	64.7	58.2	63.2	**65.7**	59.1
SVAMP	77.5	72.1	**79.9**	77.5	80.1
MINERVA	**12.1**	8.8	10.3	8.8	5.1
AQUA	**57.9**	49.2	51.6	49.2	51.2
GAOKAO	38.2	39.2	36.9	**40.5**	28.8
CM.BaI	38.7	39.5	40.9	**41.7**	31.9
Avg.	**47.1**	43.4	46.0	46.2	40.8

w/o Length Corr. Removing length correction biases the selection toward shorter samples, as token-level loss noise disproportionately affects longer sequences, leading to weak chain-of-thought representations and degraded performance.

w/o Segments Omitting segment-wise rectification skews performance toward over-represented domains. The learner-centric rectification, by replenishing underrepresented features on the fly, enforces a more diverse data stream.

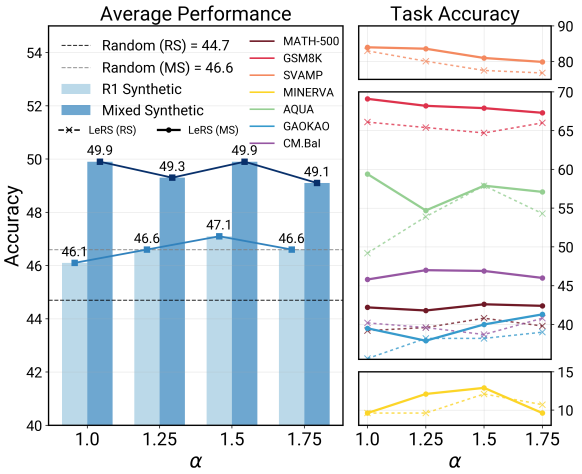


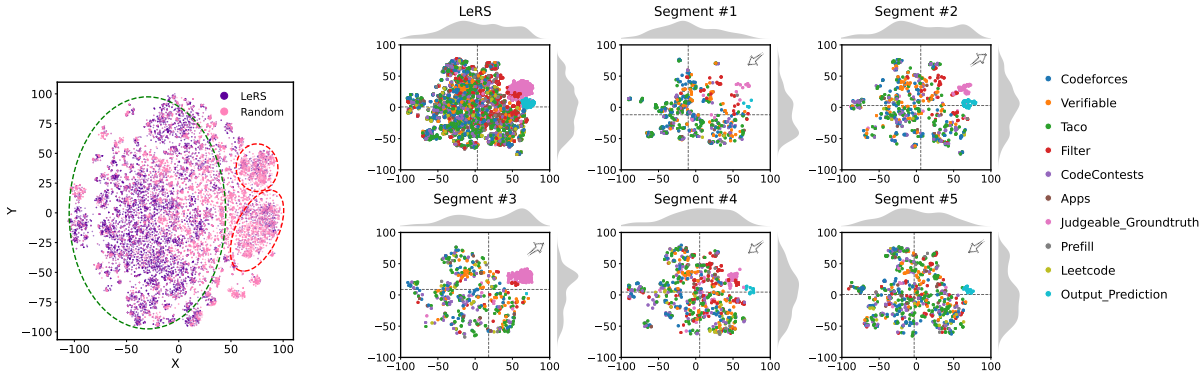
Figure 3: LERS Performance on Mathematical Tasks under Varying Penalty α. We evaluate two distinct configurations: (i) R1 Synthetic (RS), containing responses exclusively from DEEPSEEK-R1; and (ii) Mixed Synthetic (MS), a balanced 1:1 mixture of responses from QWEN3-1.7B INSTRUCT and DEEPSEEK-R1.

w/o Est. Solely disabling linear interpolation causes a modest decline, but the performance is still unbalanced, indicating that linear interpolation yields more precise and adaptive quality estimates.

w/o Expert Excluding the expert model substantially lowers the data quality, causing a notable performance degradation, and highlighting the necessity of the expert-referenced setup. Relying solely on the learner model's judgment grossly misrates low-quality data, which are outliers that can in turn mislead even the expert.

4.4 Learner Preferences and Penalty

In this section, we present counterintuitive evidence to highlight the profound impact of the learner's intrinsic token preferences, and provide comprehensive insights into the role of preference penalty.



(a) Random and LERS-selected data distribution comparison.

(b) Distribution of selected samples in each rectification segment of LERS, divided in source categories. Horizon and Vertical lines represents the average of points.

Figure 4: Embedding t-SNE visualization of code training samples. (a) LERS yields a more uniform distribution than random sampling. (b) LERS dynamically corrects distributional biases through segment-wise updates.

Alongside the verifiable mathematical data pool synthesized by DEEPSEEK-R1 (noted as "R1 Synthetic"), we generate an additional split by prompting QWEN3-1.7B INSTRUCT with identically distributed questions. These are merged into a 100K "Mixed Synthetic" (MS) dataset with a 1:1 ratio.

Although R1 significantly surpasses QWEN3-1.7B INSTRUCT in math reasoning, a 0.6B learner model trained on the latter's synthetic data outperforms its counterpart trained exclusively on DEEPSEEK-R1 answers by approximately +3%. This indicates that the learner's endogenous token priors exert a decisive influence. Preference-congruent samples are assimilated more readily than robust yet complex ones. The penalty α is introduced to mitigate such intrinsic learning bias.

Our ablation experiments on both pools (Figure 3) reveal that training on MS consistently outperforms RS across all tasks and selection methods. In pool RS, where learner preference is less polarized, penalizing with α recuperates a subset of pedagogically valuable samples. Yet in pool MS, where the proportion of preference-aligned samples is enriched, performance gains remain stable even at a minimal penalty ($\alpha = 1.0$). This confirms the effectiveness of penalizing the selection of hard-to-grasp samples using α. Notably, LERS exceeds random selection across the entire range of α.

4.5 Analysis

The mechanism behind LERS remains unclear. Why does a selected subset consistently outperform full-data fine-tuning? To elucidate this, Figure 4 visualizes (a) the embedding of samples selected via random sampling versus LERS, and (b) the evolution of LERS selections across the initial five rectification segments. In Figure 4a, random selection, which is expected to inherit the inherent distributional skew of the full dataset, leads to a high-density cluster (red ellipse) that triggers local over-fitting. In contrast, LERS's cohort (green ellipse) uniformly populates the manifold. The segment-wise rectification depicted in Figure 4b reveals a bias-correcting feedback loop driven by learner-centric HCCG signal: samples in segment #1 over-represent the third-quadrant mass. Consequently, segments #2–#3 compensate by recruiting first-quadrant instances. Once the drift becomes positive, the sample distribution in segment #4 re-orients the selection toward negative x- and y-coordinates, and the process stabilizes at segment #5. The dynamic validates LERS's ability to escape the "majority-bias" trap.

5 Conclusion

In this paper, we introduce LERS, a learner-centric data selection framework that improves data efficiency and efficacy by addressing redundancy, distributional imbalance, and poor learnability issues in synthetic SFT datasets. LERS estimates the Homologous Corrected Cognitive Gap (HCCG) metric, enabling the learner itself to adaptively identify current high-utility samples that rectify distributional biases and complement the selected subset with suitable challenging samples during learning. Extensive experiments demonstrate that LERS consistently outperforms all baselines, achieves $5\times$ data efficiency gains, and surpasses full-data training in multi-source settings with highly improved data efficacy. These gains further generalize across models at similar scales. Our findings demonstrate that prioritizing data aligned with the learner's evolving needs is fundamental to improving data efficiency and efficacy in supervised fine-tuning.

6 Limiation

We demonstrate that data quality is learner-centric and validate LERS's ability to enhance data efficiency and efficacy, especially for frequently used small-scale models whose capacity diverges most from synthetic-data generators. While LERS provides an effective selection strategy for resource-constrained model customization, its scalability to larger models requires further investigation. Such generalization may depend on a complex interplay between data sources and learner capabilities. Furthermore, several important phenomena identified in this study, such as the impact of the learner's endogenous preferences on learning and their trade-off with sample robustness, warrant future systematic analysis across models and scenarios. Finally, although we determined an optimal segment count via ablation, future work should explore whether dynamic or non-uniform segmentation, tailored to the data pool, could better fit the HCCG curve while reducing computational overhead.

This work also serves as a caution against the indiscriminate migration of training data generated from SOTA LLMs, which may lead to training degradation and diminished returns. Further explorations are encouraged into more customized resources and data governance strategies tailored to the specific status of the learner model.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ning Bian, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Ben He. 2023. Chatalpaca: A multi-turn dialogue corpus based on alpaca instructions. https://github.com/cascip/ChatAlpaca.

Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025. Enhancing reasoning abilities of small LLMs with cognitive alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7434–7449, Suzhou, China. Association for Computational Linguistics.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. *Preprint*, arXiv:2307.06290.

Jie Chen, Yupeng Zhang, Bingning Wang, Wayne Xin Zhao, Ji-Rong Wen, and Weipeng Chen. 2024. Unveiling the flaws: exploring imperfections in synthetic data and mitigation strategies for large language models. *arXiv preprint arXiv:2406.12397*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

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*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and JingBo Zhu. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 464–478, Miami, Florida, USA. Association for Computational Linguistics.

Benyamin Ghojogh and Mark Crowley. 2023. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *Preprint*, arXiv:1905.12787.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, and 31 others. 2025. Openthoughts: Data recipes for reasoning models. *Preprint*, arXiv:2506.04178.

Yuto Harada, Yusuke Yamauchi, Yusuke Oda, Yohei Oseki, Yusuke Miyao, and Yu Takagi. 2025. Massive supervised fine-tuning experiments reveal how data, layer, and training factors shape LLM alignment quality. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*,

pages 22371–22392, Suzhou, China. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. *Preprint*, arXiv:2409.12186.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint*.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference leakage: A contamination problem in llm-as-a-judge. *arXiv preprint arXiv:2502.01534*.

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 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *Preprint*, arXiv:2312.15685.

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*.

Kairong Luo, Haodong Wen, Shengding Hu, Zhenbo Sun, Zhiyuan Liu, Maosong Sun, Kaifeng Lyu, and Wenguang Chen. 2025. A multi-power law for loss curve prediction across learning rate schedules. *Preprint*, arXiv:2503.12811.

Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Christos Thrampoulidis, Leonid Sigal, and Renjie Liao. 2025. Leveraging online olympiad-level math problems for llms training and contamination-resistant evaluation. *Preprint*, arXiv:2501.14275.

John X. Morris, Junjie Oscar Yin, Woojeong Kim, Vitaly Shmatikov, and Alexander M. Rush. 2025. Approximating language model training data from weights. *Preprint*, arXiv:2506.15553.

Mihai Nadăș, Laura Dioșan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, and 1 others. 2025. Olmo 3. *arXiv preprint arXiv:2512.13961*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. 2018. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Charles M Stein. 1981. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151.

Qiushi Sun, Jinyang Gong, Lei Li, Qipeng Guo, and Fei Yuan. 2025. Codeevo: Interaction-driven synthesis of code-centric data through hybrid and iterative feedback. *Preprint*, arXiv:2507.22080.

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

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Preprint*, arXiv:2407.13690.

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.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, and 1 others. 2024. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37:116462–116492.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024a. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. Less: Selecting influential data for targeted instruction tuning. *Preprint*, arXiv:2402.04333.

Xinyi Xu, Zhaoxuan Wu, Rui Qiao, Arun Verma, Yao Shu, Jingtan Wang, Xinyuan Niu, Zhenfeng He, Jiangwei Chen, Zijian Zhou, and 1 others. 2024. Data-centric ai in the age of large language models. *arXiv preprint arXiv:2406.14473*.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. 2025. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024a. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao, and Chenghua Lin. 2024b. Effective distillation of table-based reasoning ability from llms. In *Proceedings of the 2024 Joint International Conference*

on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 5538–5550.

Chen Yang, Guangyue Peng, Jiaying Zhu, Ran Le, Ruixiang Feng, Tao Zhang, Wei Ruan, Xiaoqi Liu, Xiaoxue Cheng, Xiyun Xu, and 1 others. 2025b. Nanbeige4-3b technical report: Exploring the frontier of small language models. *arXiv preprint arXiv:2512.06266*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

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.

Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. 2024. Automatic instruction evolving for large language models. *Preprint*, arXiv:2406.00770.

Jianguo Zhang, Kun Qian, Zhiwei Liu, Shelby Heinecke, Rui Meng, Ye Liu, Zhou Yu, Huan Wang, Silvio Savarese, and Caiming Xiong. 2024. Dialogstudio: Towards richest and most diverse unified dataset collection for conversational ai. *Preprint*, arXiv:2307.10172.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. 2025. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4671–4686.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark.

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Minghao Li, Fei Huang, Nevin L. Zhang, and Yongbin Li. 2024. Tree-instruct: A preliminary study of the intrinsic relationship between complexity and alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16776–16789, Torino, Italia. ELRA and ICCL.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2025. Opencodeinterpreter: Integrating code generation with execution and refinement. *Preprint*, arXiv:2402.14658.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

A Appendix

A.1 Derivations

Proof of Stein's Lemma For a Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ and any absolutely continuous function g whose derivative is integrable, the standard integral expression is

$$\mathbb{E}[g'(X)] = \int_{-\infty}^{\infty} g'(x)\,\varphi(x)\,\mathrm{d}x$$
$$\varphi(x) = \mathcal{N}(x; \mu, \sigma^2)$$

where φ denotes the Gaussian probability density function. Due to the exponential decay of the Gaussian density, the boundary term $g(x)\varphi(x)$ vanishes as $x \to \pm\infty$, ensuring that the integration-by-parts formula applies. Thus,

$$\mathbb{E}[g'(X)] = \Big[g(x)\varphi(x)\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} g(x)\,\varphi'(x)\,\mathrm{d}x$$
$$= -\int_{-\infty}^{\infty} g(x)\Big(-\tfrac{x-\mu}{\sigma^2}\varphi(x)\Big)\,\mathrm{d}x,$$

where the second equality uses the identity $\varphi'(x) = -\frac{x-\mu}{\sigma^2}\varphi(x)$, which follows from directly differentiating the Gaussian density. Recognizing the resulting integral as an expectation obtains

$$\mathbb{E}[g'(X)] = \tfrac{1}{\sigma^2}\,\mathbb{E}[g(X)(X-\mu)]\,.$$

which completes the proof.

Bias-Variance Decomposition of Cross-Entropy Loss Let $x \in \mathcal{X}$ be an input, and the ground-truth conditional distribution be $p(\cdot \mid x)$ over labels $y \in \mathcal{Y}$. Consider a model that produces conditional distribution $q(\cdot \mid x; \theta)$. The cross-entropy of model output q under distribution p is

$$\mathcal{L}_{CE}\big(p, q(\cdot \mid x; \theta)\big) \;=\; -\sum_{y \in \mathcal{Y}} p(y \mid x)\log q(y \mid x; \theta).$$

Taking the expectation over a data category $\mathcal{D}_i$ as mentioned in 3.1, and using the decomposition $\mathcal{L}_{CE}(p, q) = H(p) + \mathrm{KL}(p\|q)$, we obtain

$$\mathbb{E}_{x\sim\mathcal{D}_i}[\mathcal{L}_{CE}(p, q(\cdot \mid x))] = \mathbb{E}_{x\sim\mathcal{D}_i}[H(p(\cdot \mid x))] + \mathbb{E}_{x\sim\mathcal{D}_i}[\mathrm{KL}(p(\cdot \mid x)\,\|\,q(\cdot \mid x))]\,, \quad (14)$$

which rewrites the expected cross-entropy loss as the sum of the conditional entropy, and the expected KL-divergence between the ground-truth distribution and the model prediction.

To further decompose the expected KL-divergence, we separate it into a *bias* term, measuring the discrepancy between the truth and the mean model prediction, and a *variance-like* term that reflects prediction variability under data sampling. Define the mean prediction over dataset D_i as

$$\bar{q}(y \mid x) = \mathbb{E}_{x\sim D_i}\big[q(y \mid x)\big].$$

By direct algebra, we obtain

$$\mathbb{E}_{x\sim\mathcal{D}_i}[\mathrm{KL}(p\|q)] = \mathrm{KL}(p\,\|\,\bar{q}) + \underbrace{\mathbb{E}_{x\sim\mathcal{D}_i}\left[\sum_y p(y \mid x)\log\frac{\bar{q}(y \mid x)}{q(y \mid x)}\right]}_{V(p,\{q\},\mathcal{D}_i)\text{: variance-like term}}, \quad (15)$$

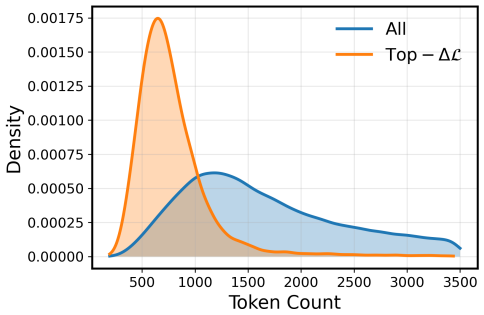
where the first term captures systematic bias and the second aggregates variability in predictions relative to the mean predictor.

A.2 Potential Issues of Other Scoring Metrics

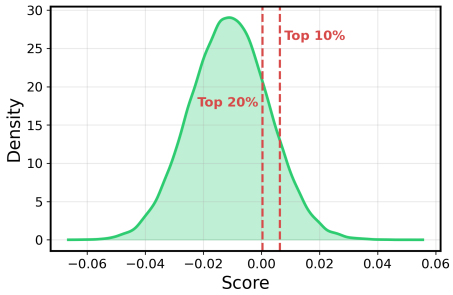
In this section, we analyze several established quality metrics typically computed using learner-expert model pairs. We examine their intrinsic limitations, and articulate how these issues motivate our proposed approach.

For the loss-gap metric $\Delta\mathcal{L}$, prior work (Cao et al., 2024) proposes similar metric suggesting that larger values indicate samples from which the model can benefit more, implying higher learning value. While this intuition is broadly correct, it introduces a critical issue in our setting: because cross-entropy loss is computed token-wise, shorter samples exhibit substantially higher variance in their scores. As illustrated in Fig. 5a, this bias causes the metric to disproportionately favor short examples, ultimately harming generalization. Consequently, HCCG adopts a length-corrected variant to mitigate this artifact.

For Gradient Cosine Similarity (GCS), a metric widely used in computer-vision data selection and recently adapted to NLP by Morris et al. (2025), the main challenge lies in its prohibitive computational cost when evaluated on full-model gradients when it comes to large language models. In practice, one must rely on aggressive approximations such as low-rank gradient compression, or single-layer extraction. These approximations discard informative gradient structure, yielding scores that resemble near-Gaussian noise with extremely small variance (Fig. 5b). Moreover, only a small fraction of samples exhibit gradient within positive direction. As a result, the metric provides limited actionable signal for data selection in LLM.



(a) Comparison of the token count distributions between the $\Delta\mathcal{L}$-selected subset and the full dataset.



(b) Gradient cosine similarity score distribution.

Figure 5: Density Distribution Plots. Judgments on $\Delta\mathcal{L}$ tends to prefer short response, while Gradient Cosine Similarity score distribution acting like gaussian noise.

A.3 Experimental Details

A.3.1 Baseline Details

For the learner-expert twined selection workflow, we adopt a series of static metrics as baselines. Let the parameters of the base model f_b and the expert model f_e be θ_b and θ_e, respectively. They are:

- **Loss Difference ($\Delta\mathcal{L}$)** the difference in cross-entropy loss, $\mathcal{Q} = \mathcal{L}(f_b, x, y) - \mathcal{L}(f_e, x, y)$, serving as a proxy for sample value.

- **KL Divergence (KL)** the KL-divergence from the likelihood distribution from the base to the expert model,

$$\mathcal{Q} = D_{\mathrm{KL}}\big(f_e(\cdot|x) \parallel f_b(\cdot|x)\big) = \sum_{y\in\mathcal{V}} f_e(y|x) \log \frac{f_e(y|x)}{f_b(y|x)}.$$

- **Gradient Cosine Similarity (GCS)** cosine similarity between the per-sample gradient and the parameter discrepancy vector,

$$\mathcal{Q} = \frac{\langle \nabla_{\theta_b}\mathcal{L},\, \theta_e - \theta_b\rangle}{\|\nabla_{\theta_b}\mathcal{L}\|\,\|\theta_e - \theta_b\|}.$$

  To retain computational feasibility, we follow Morris et al. (2025) and approximate the gradient using the `lm_head` layer under a 4096-dimensional low-rank projection.

- **Gradient Projection (GP)** the scalar projection of the per-sample gradient onto the parameter discrepancy vector,

$$\mathcal{Q} = \frac{\langle \nabla_{\theta_b}\mathcal{L},\, \theta_e - \theta_b\rangle}{\|\theta_e - \theta_b\|}.$$

A.3.2 Implement Details

The hyperparameters and configuration selections presented in Section 4.1 are systematically compiled and uniformly displayed in Table 4.

The evaluation details are enumerated in Table 5. Owing to the incomplete public release of the evaluation pipeline for QWEN3, we implemented the evaluation on top of the libraries released with Qwen2.5-Math (Yang et al., 2024a) and Qwen2.5-Coder (Hui et al., 2024). Every benchmark is executed under 0-shot mode with max_length=16,384, temperature=0.6, and top-p=0.95.

A.3.3 Ablation of Segments Count K

Table 8 reports the fine-tuning performance of QWEN3-0.6B as the number of segment cut points K growth. Across all segment counts, the performance of LERS selected dataset consistently surpasses that obtained by randomly selecting 10K training examples. Accuracy first improves and then decreases along with segments increases, indicating that moderately increasing the number of segments helps mitigate the selection bias of the training set. However, an excessive number of segments inflate computational overhead but amplifies the noise inherent in the optimization process.

A.3.4 Cross-Model Transferability of Data

To gain a deeper understanding of the generalizability of LERS-selected data, and to analyze the transferability of compact models (e.g., QWEN3-0.6B BASE/INSTRUCT) as efficient proxies, we investigate the consistency of the HCCG metric across diverse model architectures. Specifically, we randomly sample 10,000 mathematical data points and computed the Pearson correlation coefficients of HCCG scores across various learner-expert pairs. These pairs include the homogeneous QWEN3 series (0.6B, 1.7B, 4B, and 8B), as well as heterogeneous models with distinct pre-training backgrounds, such as NANBEIGE4-3B BASE & THINKING (Yang et al., 2025b) and OLMO3-7B BASE & INSTRUCT (Olmo et al., 2025).

Table 4: Hyperparameter and Implementation Details for Main Experiments.

Stage	Parameter	Value	Notes
Data	Synthetic dataset size N	100K	
	Selected subset $\tilde{N}$	10K	
	Data types	Math / Code	
	Context length	$\leq$ 3500 tokens	
Selection	Student model	QWEN3-0.6B BASE	Model f_b
	Expert model	QWEN3-0.6B INSTRUCT	Model f_e
	Segment cut points K	8	For segmentation
	Preference penalty α (Math)	1.5	Due to expert model capability
	Preference penalty α (Code)	1.0	
Fine-Tuning	Learning rate candidates	$\{7\times10^{-5}, 5\times10^{-5}, 3\times10^{-5}, 1\times10^{-5}\}$	Grid search range
	Selected LR (0.6B)	7×10^{-5}	
	Selected LR (1.7B)	3×10^{-5}	
	Selected LR (4B)	1×10^{-5}	
	Batch size	16	Total batch size
	Weight decay	0.01	
	Warmup ratio	0.05	w Cosine Scheduler
	Epoch	3	
	Seed	45	Fixed for all runs

Table 5: Details of the Benchmarks Used for Evaluation.

Task	Benchmarks	Category
Math	MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), MINERVA_MATH (Lewkowycz et al., 2022), GAOKAO2023EN (Zhang et al., 2023), CollegeMath.Beginning_and_Intermediate_Algebra (Tong et al., 2024)	Problem Solving
	AQUA (Ling et al., 2017)	Multiple-Choice
Code	HumanEval(+) (Chen et al., 2021), MBPP(+) (Austin et al., 2021), LiveCodeBenchv1 (Jain et al., 2024)	Code Generation

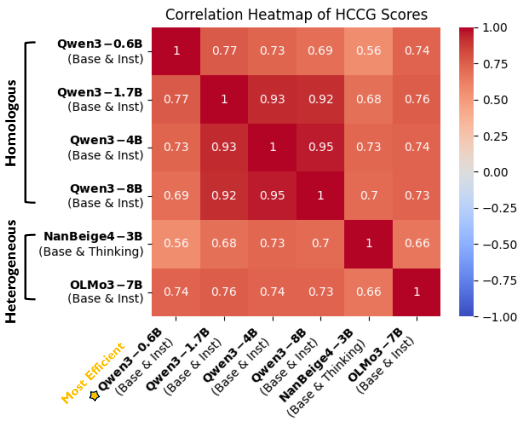


Figure 6: Cross-model Correlation of HCCG Across Diverse Learner-Expert Pairs. Despite variations in pre-training backgrounds and architectures, the observed cognitive gaps maintain high similarity. This justifies the use of compact models as computational efficient proxies, and demonstrates that the utility of selected data transfers effectively across learners at similar scale.

The resulting correlation heatmap is presented in Figure 6. The results indicate that when using QWEN3-0.6B as the cognition proxy, the derived HCCG exhibits strong correlations with both intra-family and inter-family models. It computed HCCG with compact proxies can effectively reflect the cognitive gaps in models at similar scales or capability background. However, we observe a decline in similarity as the model size increases, which underscores the necessity of learner-centric data selection - models of different scales possess distinct cognitive biases.

Furthermore, we conduct supervised fine-tuning on two heterogeneous models, NANBEIGE4-3B-BASE and OLMO3-7B-BASE, using the mathematical reasoning selected data subset in Table 1. The evaluation results are summarized in Table 7. LeRS-selected data consistently outperforms random selection downstream fine-tuning outcomes.

The findings demonstrate that, despite the change in the learner models, the data selected by QWEN3-0.6B learner-expert pair remains highly effective for other small-scale models, benefiting from the robust correlation of the HCCG scores.

A.4 Cost Measurement

Taking the experimental setup in Section 4.1 as an example, we report the end-to-end runtime of different data selection methods and the training cost of models at various scales. All measurements are obtained on NVIDIA H100 GPUs, and the results are summarized in Figure 7.

Although LERS performs segment-wise subset distribution rectification and offers a more robust data selection pipeline, the overall computational overhead remains low due to the lightweight HCCG metric computation, which only requires forward propagation to calculate, and benefit from the acceleration by vLLM. The executing time is much lower than that of LLM evaluator-based methods such as DEITA.

In our experiments, LERS achieves up to a $5\times$ improvement in data efficiency, substantially reducing the training time required to reach comparable performance. Moreover, the selected dataset from a single pass can be reused to fine-tune suites of homologous models, providing a readily reusable resource for the community.

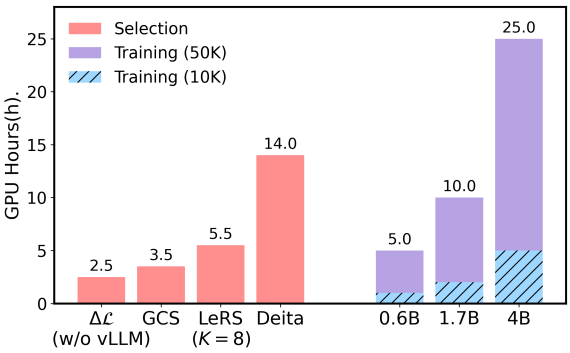


Figure 7: Execution GPU Hours Comparision. LERS achieve 5x data efficiency in mathematica tasks fine-tuning with low additional data selection costs.

Algorithm 1 LERS

Input: Candidate dataset $\mathcal{D}$, total sample size $\tilde{N}$, segment division numbers K, batch size B
for $k = 0$ to K **do**
  $\tilde{\mathcal{D}}_+ \leftarrow \text{random}(\tilde{\mathcal{D}}, T_{k+1} - T_k)$
  $\hat{g}_{T_k} \leftarrow \hat{f}_{T_k}$
  for every batch $\mathcal{B}_t$ in $\tilde{\mathcal{D}}_+$ **do**
    $\hat{g}_t \leftarrow \hat{g}_{t-B} - \eta \nabla \mathcal{L}(\mathcal{B}_t)$
  end for
  Compute $\mathcal{Q}_{T_k}$ on checkpoint $\hat{f}_{T_k}$
  Compute $\mathcal{Q}_{T_{k+1}}$ on checkpoint $\hat{g}_{T_{k+1}}$
  $\Delta\mathcal{Q} \leftarrow (\mathcal{Q}_{T_{k+1}} - \mathcal{Q}_{T_k})/(T_{k+1} - T_k)$
  $\hat{\mathcal{Q}} \leftarrow \mathcal{Q}_{T_k}$
  for each step $T \in (T_k, T_{k+1}]$ **do**
    $\mathcal{B} \leftarrow \text{Top-K Indices}\left(\hat{\mathcal{Q}}_T, \text{Mask}\right)$
    $\hat{f}_T \leftarrow \hat{f}_{T-B} - \eta \nabla \mathcal{L}(\mathcal{B})$
    $\text{Mask}[\mathcal{B}] \leftarrow 1$, $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}} \cup \mathcal{B}$
    $\hat{\mathcal{Q}}_{T+B} \leftarrow \hat{\mathcal{Q}} + \Delta\mathcal{Q}$, $T \leftarrow T + B$
  end for
end for

Table 6: Details of Learner Preferences Penalty Ablation Study in Figure 3.

Dataset	α	MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.
R1 Synthetic	1.0	39.2	**66.1**	**83.0**	9.6	49.2	35.6	40.2	46.1
	1.25	39.6	65.4	80.1	9.6	53.9	38.2	39.6	46.6
	1.5	**40.8**	64.7	77.5	**12.1**	**57.9**	38.2	38.7	**47.1**
	1.75	39.8	66.0	76.8	10.7	54.3	**39.0**	**40.8**	46.6
Mixed Synthetic	1.0	42.2	**69.1**	**84.0**	9.6	**59.4**	39.5	45.8	**49.9**
	1.25	41.8	68.2	83.6	12.1	54.7	37.9	**47.0**	49.3
	1.5	**42.6**	67.9	81.0	**12.9**	57.9	40.0	46.9	**49.9**
	1.75	42.4	67.3	79.9	9.6	57.1	**41.3**	46.0	49.1

Table 7: Experimental Results of Data Efficiency Cross-Model Transferability on Mathematical Tasks. LERS-selected data maintains high data efficiency improvement across varying model architectures and scales.

Model	Method	Percent (%)	MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.
NANBEIGE4-3B BASE	RANDOM	10%	76.2	89.2	93.3	31.2	81.1	69.9	62.6	71.9
		50%	79.4	**90.3**	**94.7**	29.0	**86.2**	71.9	65.6	73.9
	LERS	10%	**82.0**	89.5	93.0	**35.3**	83.9	**74.8**	**65.9**	**74.9**
OLMO3-7B BASE	RANDOM	10%	37.0	77.3	86.3	**9.6**	54.7	**35.3**	43.4	49.1
		50%	**37.4**	75.1	85.5	8.1	52.0	34.3	44.1	48.1
	LERS	10%	36.2	**80.7**	**88.2**	**9.6**	**60.6**	35.1	**45.3**	**50.8**

Table 8: Evaluation of the Sensitivity to LERS Rectification Segment Granularity in QWEN3-0.6B Fine-Tuning.

K	MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.
0	39.8	63.2	**79.9**	10.3	51.6	36.9	40.9	46.0
4	41.8	64.3	76.3	10.3	54.3	36.1	42.0	46.4
8	40.8	64.7	77.5	12.1	**57.9**	**38.2**	38.7	**47.1**
12	**42.4**	65.0	77.3	10.7	52.0	37.4	40.0	46.4
16	40.2	**66.3**	77.5	8.5	52.8	37.7	**42.8**	46.5

Table 9: Overall Comparison between Random-Selected and LERS-Selected Data Fine-Tuning Performance.

Model	Method	Percent(%)	MATH-500	GSM8K	SVAMP	MINERVA	AQUA	GAOKAO	CM.BaI	Avg.	HumanEval (+)	MBPP (+)	LCBv1	Avg (+).
QWEN3-0.6B BASE	RANDOM	10%	37.0	63.1	79.8	8.1	51.2	35.6	38.3	44.7	12.8 (11.6)	21.1 (18.8)	-	17.0 (15.2)
		50%	**42.4**	68.2	83.8	9.2	52.0	38.7	39.5	47.7	15.9 (13.4)	24.8 (21.6)	-	20.4 (17.5)
		100%	41.6	**68.8**	**84.3**	8.8	56.7	**41.6**	**41.9**	**49.1**	14.6 (13.4)	25.2 (22.4)	-	19.9 (17.9)
	LERS	10%	40.8	64.7	77.5	**12.1**	**57.9**	38.2	38.7	47.1	**23.8 (20.7)**	**30.3 (25.1)**	-	**27.1 (22.9)**
	vs. RDM $\Delta_{10\%}$		*+3.8*	*+1.6*	*-2.3*	*+4.0*	*+6.7*	*+2.6*	*+0.4*	*+2.4*	*+11.0 (+9.1)*	*+9.2 (+6.3)*	-	*+10.1 (+7.7)*
	vs. RDM $\Delta_{100\%}$		*-0.8*	*-4.1*	*-6.8*	*+3.3*	*+1.2*	*-3.4*	*-3.2*	*-2.0*	*+9.2 (+7.3)*	*+5.1 (+2.7)*	-	*+7.2 (+5.0)*
QWEN3-1.7B BASE	RANDOM	10%	61.2	82.7	92.5	20.6	70.1	54.5	56.6	62.6	37.2 (32.9)	45.9 (41.1)	9.3	30.8 (27.8)
		50%	63.8	83.7	93.2	**24.6**	73.6	56.6	58.3	64.8	37.8 (31.7)	**50.4** (42.4)	1.5	29.9 (25.2)
		100%	64.8	**86.2**	**95.0**	23.2	73.2	**57.7**	**63.1**	**66.2**	36.6 (32.3)	43.6 (37.3)	**9.5**	30.0 (26.4)
	LERS	10%	**66.2**	85.1	92.4	22.8	**76.8**	56.1	59.2	65.5	**40.2 (36.6)**	**50.4 (43.6)**	7.0	**32.5 (29.1)**
	vs. RDM $\Delta_{10\%}$		*+5.0*	*+2.4*	*-0.1*	*+2.2*	*+6.7*	*+1.6*	*+2.6*	*+2.9*	*+3.0 (+3.7)*	*+4.5 (+2.5)*	*-2.3*	*+1.7 (+1.3)*
	vs. RDM $\Delta_{100\%}$		*+1.4*	*-1.1*	*-2.6*	*-0.4*	*+3.6*	*-1.6*	*-3.9*	*-0.7*	*+3.6 (+4.3)*	*+6.8 (+6.3)*	*-2.5*	*+2.5 (+2.7)*
QWEN3-4B BASE	RANDOM	10%	82.0	93.4	**95.7**	36.8	78.3	73.2	67.8	75.3	64.6 (59.8)	69.4 (59.1)	30.8	54.9 (49.9)
		50%	**85.6**	93.9	**95.7**	39.3	**83.1**	74.0	**68.2**	**77.1**	63.4 (57.9)	**71.4** (60.2)	33.5	56.1 (50.5)
		100%	84.0	**95.2**	95.5	40.8	80.7	75.3	68.0	**77.1**	65.0 (59.7)	67.2 (57.4)	34.3	55.5 (50.5)
	LERS	10%	**85.6**	93.4	95.6	**42.6**	75.6	**75.8**	67.3	76.6	**66.5 (62.8)**	68.2 (58.9)	**37.3**	**57.3 (53.0)**
	vs. RDM $\Delta_{10\%}$		*+3.6*	*+0.0*	*-0.1*	*+5.8*	*-2.7*	*+2.6*	*-0.5*	*+1.3*	*+1.9 (+3.0)*	*-1.2 (-0.2)*	*+6.5*	*+2.4 (+3.1)*
	vs. RDM $\Delta_{100\%}$		*+1.4*	*-1.8*	*+0.1*	*+1.8*	*-5.1*	*+0.5*	*-0.7*	*-0.5*	*+1.5 (+3.1)*	*+1.0 (+1.5)*	*+3.0*	*+1.8 (+2.5)*