GALA: Geometric Data Selection with Strategic Prospecting for Large Language Model Self-Training

Zhongwei Xie[1], **Ruihao Liao**[2], **Zimo Wang**[3], **Chong Chen**[4,*], **Xian-Sheng Hua**[2,5], **Xiao Luo**[6]
[1]Department of Computer Science and Engineering, HKUST
[2]Terminus Group [3]Beijing Bayi School [4]Hesicare Technology Co. Ltd
[5]Tongji University [6]University of Wisconsin–Madison
zxiebk@connect.ust.hk, chenchong@hesicare.com

Abstract

Self-training has emerged as a promising direction for autonomously improving large language models (LLMs). Existing approaches typically adopt a *generate-and-filter* paradigm based on rejection sampling, which could suffer from inefficiency and low-quality reasoning paths. Towards this end, this paper proposes a novel framework named Geometric Data Selection with Strategic Prospecting (GALA) for LLM self-training. The core of our GALA is to identify diverse and informative samples from redundant data and exploit them more strategically. In particular, our proposed GALA first conducts clustering on latent sentence embeddings and then selects an anchor sample from each cluster based on the geometric distance to reduce data redundancy. To further exploit these samples, we conduct strategic brainstorming and reflection for high-quality reasoning trajectory prospecting. In addition, we introduce a lightweight dynamic validation module to validate the reliability of mini-batches to ensure the overall quality of the data. Extensive experiments on various benchmarks validate the effectiveness of the proposed GALA against several competing baselines.

1 Introduction

The capacity for self-improvement is a crucial frontier in the evolution of large language models (LLMs) (Zhang et al., 2025a; Kumar et al., 2025), promising a path toward autonomous systems (Liu et al., 2023) that can enhance their capabilities with minimal human oversight. By requesting models to generate their own training data (Rosenberg et al., 2005; Luo et al., 2023, 2024),self-training has proven particularly effective in domains with verifiable outcomes, e.g., mathematical reasoning, where the correctness of a solution provides a clear signal for reinforcement. With numerous generated solutions and selectively training on the successful ones, LLMs can iteratively refine their problem-solving skills and reasoning abilities.

Currently, data selection and active learning (Sener and Savarese, 2017; Nguyen et al., 2022) approaches have been extended to LLM self-training (Xiao et al., 2026; Su et al., 2026; Sun et al., 2025a). These approaches usually rely on an automatic data generating process. For example, LIMO (Ye et al., 2025; Li et al., 2025) develops a high-quality dataset to improve the data efficiency of LLM post-training. However, these approaches typically focus on static dataset design, which neglects the dynamic nature of LLM self-training. Towards this end, an effective framework that can generate a high-quality data stream for LLM self-training is highly anticipated.

The majority of approaches in this line follow a *generate-and-filter* paradigm, which is often implemented via rejection sampling (Zelikman et al., 2022; Singh et al., 2023; Luong et al., 2024). To be specific, given a question, we require LLMs to generate a range of reasoning paths, and then retain only those that match the ground-truth answer. However, these approaches would generate semantically equivalent paths with binary reward signals, resulting in diminishing returns during LLM self-evolution (Du et al., 2024; Zhu et al., 2025). In addition, the generated correct solution could collapse with a limited number of dominant patterns, implying the lack of path diversity and instructional depth expected during post-training. Furthermore, the current data selection paradigm usually relies on offline evaluation (Wang et al., 2022; Singh et al., 2023), which could be out-of-date during the post-training process.

To address these challenges, we propose a new approach named GALA, which adopts a "curate-smarter" paradigm to enhance LLM self-training. The core idea of our proposed GALA is to introduce

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 10281–10293

a holistic pipeline where three important stages are integrated to handle the aforementioned bottlenecks. In particular, GALA first introduces principled geometric subset selection, which optimizes both redundancy and diversity to enhance data efficiency of LLM self-training. In addition, GALA enhances instructional depth and strategic variety during solution generation, which maximizes the utility of limited samples. Finally, we introduce a lightweight dynamic validation filter, which further mitigates potential noise with fine-grained feedback. Our experimental results on the benchmark datasets validate that our GALA can achieve competing performance with much higher data efficiency.

In sumary, our contributions can be summarized as follows:

① We propose an approach named GALA, which explores geometric relationships in the hidden space using clustering and anchoring to optimize the redundancy for LLM self-training.

② Our GALA utilizes a "curate-smarter" paradigm, which combines strategy brainstorming with reflection to make the best of selected samples.

③ Extensive experiments on multiple benchmarks show that (1) our GALA can achieve accuracy comparable to full-data baselines; and (2) our GALA completes self-training over much fewer training samples, validating its strong data-efficiency.

2 Problem Setup and Motivation

Let p_θ be a large language model (LLM) parameterized by θ. Our goal is to enhance the mathematical reasoning capabilities of p_θ through an iterative self-training process. We begin with a seed dataset $\mathcal{D}_{\text{seed}} = \{(x_i, y_i)\}_{i=1}^{N}$, where each x_i is a mathematical problem and y_i is its corresponding ground truth final answer. The objective is to enable the model to generate a correct intermediate reasoning path r_i that logically connects x_i to y_i.

2.1 The Naive Self-Training Loop

Standard self-training approaches (Zelikman et al., 2022; Singh et al., 2023; Luong et al., 2024) are typically based on rejection sampling, which follows a straightforward *generate-and-filter* procedure. This process can be considered as the *naive self-training loop*, which serves as our baseline. In particular, we start from a candidate generation process, where for each problem $x_i \in \mathcal{D}_{\text{seed}}$, the model p_θ is prompted to generate a large set of K candidate reasoning paths denoted as $\mathcal{R}_i$:

$$\mathcal{R}_i = \{\hat{r}_{i,k}\}_{k=1}^{K}, \quad \text{where} \quad \hat{r}_{i,k} \sim p_\theta(\cdot \mid x_i). \tag{1}$$

Then, in the verification and filtering stage, each generated path $\hat{r}_{i,k}$ is evaluated by a verification function $V(\hat{r}, y) \in \{0, 1\}$. The function returns 1 if the path successfully yields the ground truth answer y. All correctly verified paths are then collected to form a new training dataset $\mathcal{D}_{\text{train}}$:

$$\mathcal{D}_{\text{train}} = \Big\{(x_i, \hat{r}_{i,k}) \mid (x_i, y_i) \in \mathcal{D}_{\text{seed}}, \hat{r}_{i,k} \in \mathcal{R}_i, V(\hat{r}_{i,k}, y_i) = 1\Big\}. \tag{2}$$

Finally, the parameters of the model are updated through fine-tuning, which involves performing supervised fine-tuning (SFT) on the collected dataset $\mathcal{D}_{\text{train}}$. This is achieved by minimizing the negative log-likelihood loss:

$$\theta_{\text{new}} \leftarrow \arg\min_{\theta} \mathcal{L}_{\text{SFT}}(\theta; \mathcal{D}_{\text{train}}), \quad \text{where} \quad \mathcal{L}_{\text{SFT}} = -\,\mathbb{E}_{(x,r)\in\mathcal{D}_{\text{train}}}[\log p_\theta(r \mid x)]. \tag{3}$$

This loop is repeated iteratively in self-training by using the updated model θ_{new} for the next step.

2.2 Core Challenges: Redundancy and Lack of Diversity

Note that the naive self-training loop is highly inefficient and often results in a suboptimal training dataset (Du et al., 2024; Zhu et al., 2025) due to two primary shortcomings as follows. Let $E : s \mapsto \mathbf{e} \in \mathbb{R}^d$ be a function that maps any sample sentence s to a semantic vector embedding $\mathbf{e}$.

Data Redundancy. The original generation process often produces a range of reasoning paths that are syntactically different but semantically equivalent (Wang et al., 2026). We define the redundancy of a dataset $\mathcal{D}$ as the average pairwise cosine similarity of the sample embeddings:

$$\mathcal{J}(\mathcal{D}) \triangleq \frac{1}{|\mathcal{D}|(|\mathcal{D}|-1)} \sum_{\substack{s_i, s_j \in \mathcal{D} \\ i \neq j}} \cos(E(s_i), E(s_j)). \tag{4}$$

Intuitively, a high redundancy $\mathcal{J}(\mathcal{D})$ indicates that the dataset contains similar logic in different words, which leads to inefficiency and redundancy during self-training.

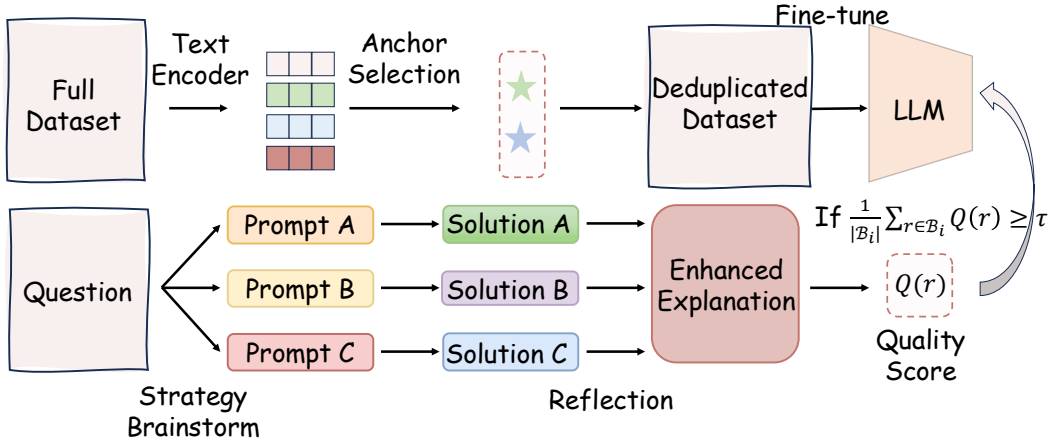


Figure 1: **An overview of our data-efficient self-training framework GALA.** The framework engineers a high-quality training set from a large raw dataset via a three-stage pipeline: **(1) Efficient Subset Selection,** where a geometric subset selection method identifies a diverse, non-redundant core set. **(2) Strategic Trajectory Exploration,** where strategic augmentation enhances the core set with high-quality reasoning paths. **(3) Dynamic Validation,** where a lightweight dynamic filter ensures the integrity of the final training data.

Lack of Strategic Diversity and Instructional Depth. Existing methods utilize the ground truth answers for filtering, which does not guarantee a strategically diverse dataset (Zelikman et al., 2022; Sachdeva et al., 2024). In this case, the model may learn to solve problems using only a few dominant reasoning patterns. To quantify and enhance the diversity of a dataset $\mathcal{D}$, we utilize a metric based on the generalized variance of its embeddings, which measures the volume occupied by sample sentences in the semantic space. Specifically, the diversity score $\mathcal{H}(\mathcal{D})$ is measured using the stabilized log-determinant of the covariance matrix (Peng et al., 2015):

$$\mathcal{H}(\mathcal{D}) \triangleq \ln\det\left(\mathrm{Cov}(\{E(s) \mid s \in \mathcal{D}\}) + \epsilon\mathbf{I}\right), \tag{5}$$

where $\epsilon\mathbf{I}$ is a small ridge term ensuring positive-definiteness. The underlying rationale for this metric is geometric. In particular, this determinant is proportional to the squared volume of the parallelepiped spanned by the embedding vectors. Maximizing $\mathcal{H}(\mathcal{D})$ forces the selection of samples that are dissimilar to one another. If two samples become collinear, the spanned volume collapses, and the metric penalizes the set. To further enhance numerical robustness, all embeddings are unit-normalized prior to computation, mitigating sensitivity to absolute scale.

Note that $\mathcal{H}(\mathcal{D})$ addresses geometric redundancy and we separately handle the qualitative issue of instructional depth, where solutions are correct but superficial. As this requires semantic refinement rather than geometric selection, it is primarily tackled in our exploration module (Section 3.3).

Directly maximizing $\mathcal{H}(\mathcal{D})$ via determinantal point processes (DPPs) incurs $O(n^3)$ cost (Kulesza and Taskar, 2012). Therefore, we leverage the Law of Total Variance to obtain an efficient approximation. For any partition of the data into C clusters, the total covariance decomposes as $\Sigma_{\text{total}} = \Sigma_W + \Sigma_B$, where Σ_W and Σ_B denote the within- and between-cluster covariance, respectively. Since K-Means minimizes $\mathrm{tr}(\Sigma_W)$, it implicitly maximizes $\mathrm{tr}(\Sigma_B)$ for a fixed Σ_{total}. By selecting anchor points that approximate these cluster centroids, our subset typically inherits this optimized between-cluster dispersion, thereby approximately optimizing $\det(\mathrm{Cov})$ and thus $\mathcal{H}(\mathcal{D})$. We would like to utilize this motivation to design our GALA.

3 The Proposed GALA

3.1 Framework Overview

To address the core challenges of data inefficiency and quality degradation inherent in naive self-training, we propose a novel self-training approach named GALA, which replaces the standard fine-tuning loop with a holistic pipeline. Instead of training on a pre-filtered dataset (Li et al., 2025; Xiao et al., 2026), the proposed GALA integrates data curation directly into the training process. In particular, we design a three-stage procedure that dynamically generates an optimal data distribution for the parameter updating. The algorithm first

selects a core set of high-impact training samples, then expands this set by exploring and synthesizing new reasoning trajectories, and finally monitors the resulting mini-batches in real-time before applying the gradient update. This unified pipeline ensures that the learning process is accompanied by a high-quality data stream.

3.2 Establishing the Strategic Scaffold (Geometric Selection)

The primary challenge in LLM self-training is identifying a compact and strategically diverse subset (Sener and Savarese, 2017; Allen-Zhu and Li, 2023) of samples from a large yet redundant pool. To solve this, we introduce a subset selection framework to identify a subset of samples whose semantic embeddings are maximally spread out. This ensures that subsequent gradient updates are informed by diverse reasoning strategies, optimizing $\mathcal{H}(\mathcal{D})$ as defined in Equation 5. To solve this efficiently within the training loop, we employ a clustering-based module, which identifies distinct clusters of reasoning strategies (Luo et al., 2025; Lin et al., 2025b; Li et al., 2026; Zhao et al., 2025) and selects a high-quality example from each to represent the primary directions for LLM self-training.

Identifying Semantic Directions via Clustering. Here, we transform each sample along with the generated reasoning paths into a vector representation using an embedding function $E(\cdot)$, i.e., the *all-mpnet-base-v2* sentence-transformer (Reimers and Gurevych, 2019). When it comes to large-scale datasets, our module acts as an efficient selection strategy, which can reduce the calculation complexity. We then apply a standard clustering algorithm (e.g., K-Means) to partition the embeddings into C distinct clusters. This acts as a powerful heuristic to identify diverse groups of learning signals. By selecting one representative from each spatially distinct cluster, we ensure the samples guiding the model update are not semantically redundant, which facilitates a high-diversity parameter update measured by $\mathcal{H}$.

Selecting Valid Representatives via Anchor Points. Centroids ($\bar{\mathbf{e}}_j$) represent the average intent of a cluster but are synthetic vectors with no textual semantics. Intuitively, training on a centroid would be like trying to learn from a blurry average of different math solutions. Instead, GALA selects the anchor point, i.e., the real data point embedding $\mathbf{e}_j^*$ within the cluster that is closest to the centroid (Diaz-Rodriguez, 2025). This selection is formalized as:

$$\mathbf{e}_j^* = \arg\min_{\mathbf{e}_k \in \mathcal{S}_j} \|\mathbf{e}_k - \bar{\mathbf{e}}_j\|_2, \quad \text{where} \quad \bar{\mathbf{e}}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\mathbf{e} \in \mathcal{S}_j} \mathbf{e}. \tag{6}$$

This ensures the model learns from human-readable logic while still capturing the core strategic direction of the cluster. In summary, our procedure is guided by the efficient max-volume approximation derived in Section 2.2, outputing the selected subset $\mathcal{D}_{\text{filter}}$ guided by both redundancy optimization ($\mathcal{J}$) and diversity optimization ($\mathcal{H}$).

3.3 Strategic Trajectory Exploration

While the selected sample set $\mathcal{D}_{\text{filter}}$ is diverse, its limited size may lead to insufficient guidance. To enrich the learning signal, GALA shifts from passive selection to active exploration (Nayab et al., 2024; Yang et al., 2025; Lu et al., 2024), enhancing each anchor point for high-quality reasoning. Intuitively, this module focuses on enhanced reasoning with the diversity of strategies and instructional depth.

Strategy Brainstorming. To encourage exploration of the solution space, we first prompt the LLM to act as an expert and brainstorm several solution strategies for a given problem (Chang and Li, 2025; Chen et al., 2025). For each proposed strategy, our model would generate a full step-by-step solution. For instance, in the Melinda dice problem, LLMs would conduct structured set-based reasoning via the Principle of Inclusion-Exclusion. We also include prompts to ensure the quality of solutions. Finally, each generated path is strategically unique and serves as a high-quality supervision signal.

Reflection. To mitigate the error propagation in self-training (Pan et al., 2023; Yang et al., 2024), we utilize a reflection mechanism (Agrawal et al., 2025). Here, the input is several potentially noisy solutions, and our reflection module aims to distill the underlying logic and then regenerate a final solution. Through this process, we effectively correct common failure modes such as incomplete casework. For example, the LLM might initially fail by ignoring reverse dice rolls (e.g., counting (1,2) but missing (2,1)) in certain problem. Through our reflection mechanism, our proposed GALA identifies this logical gap and generates an enhanced reasoning path that explicitly handles different rolls,

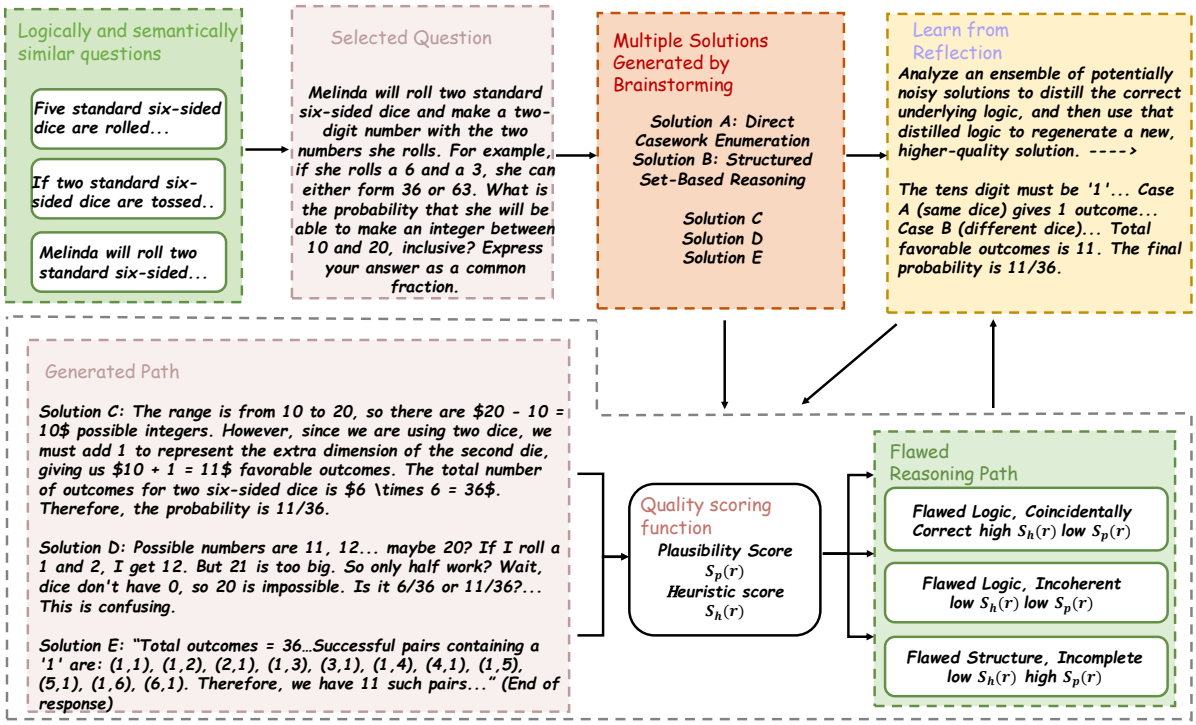


Figure 2: An illustration of strategic trajectory exploration and batch gating in our GALA framework. Given a selected question from the filtered set, we first utilize strategy brainstorming to generate multiple solutions. These potentially noisy solutions then go through an ensemble reflection process to distill the correct underlying logic and regenerate a new higher-quality enhanced solution. Finally, the generated and enhanced reasoning paths are dynamically evaluated by the quality scoring function $Q(r)$, which combines a plausibility score and a heuristic score to effectively filter out flawed logic and incomplete structures.

ensuring the depth and clarity of the reasoning process.

The resulting set $\mathcal{D}_{\text{aug}}$ makes up a strategically dense training corpus where each problem is supported by multiple deeply-reasoned trajectories. To ensure only high-quality data reaches the gradient update, $\mathcal{D}_{\text{aug}}$ is passed to the Dynamic Batch Gating module described next.

3.4 Dynamic Batch Gating and Fine-tuning

In this module, we aim to address the challenge of inefficient validation by integrating a quality control mechanism (Pan et al., 2025) directly into the fine-tuning process. This module replaces the conventional offline evaluation with a real-time mini-batch gating mechanism that provides immediate feedback. This gating mechanism ensures that the gradient update of parameters is calculated only on high-utility data, preventing parameter corruption from low-quality or noisy synthetic data generated in the exploration module.

The gating mechanism is controlled by a quality scoring function, i.e., $Q(r)$, which is a composite score balancing heuristic signals with model-based evaluation:

$$Q(r) = w_h \cdot S_h(r) + (1 - w_h) \cdot S_p(r), \quad (7)$$

where $w_h \in [0, 1]$ is a weighting hyperparameter. The two-component score include the following:

Heuristic Score ($S_h(r)$): A lightweight function that evaluates intrinsic properties (Jennings et al., 2024) of a reasoning path (e.g., length, presence of refusal phrases, numerical stability) without model inference.

Plausibility Score ($S_p(r)$): The log-probability of the reasoning path r assigned by the current state of the model (the teacher model for this iteration). Intuitively, $S_p(r)$ acts as a sanity check. In detail, if a reasoning path is nonsensical, the model's own internal probability for those tokens will be low, even if the final answer is correct. Through this, we prevent the model from learning flawed shortcuts that do not generalize. (Baral et al., 2009; Zhou et al., 2025).

This composite score allows for a robust real-time assessment of each sample. During training, the augmented data $\mathcal{D}_{\text{aug}}$ is partitioned into mini-batches $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_M\}$. The algorithm only

Table 1: Performance comparison in terms of accuracy (%) on the GSM8K, MATH, MMLU-PRO(MATH) and ASDIV datasets. The best results under SFT and DPO are shown in **bold**. Strong increase ($\geq 0.5\%$) over baselines are underlined. Overall performance is averaged on test benchmarks.

Setting	LLaMA-3.2-1B					LLaMA-3.1-8B				
	GSM8K(%)	MATH(%)	MMLU-PRO(%)	ASDIV(%)	AVG(%)	GSM8K(%)	MATH(%)	MMLU-PRO(%)	ASDIV(%)	AVG(%)
Default	40.0	19.9	9.1	54.5	30.9	84.2	40.0	49.4	81.1	63.7
SFT										
Full Data	49.1	24.0	10.5	55.8	34.9	84.6	40.3	49.7	82.6	64.3
+Random Sampling	49.0$^{\downarrow 0.1}$	18.1$^{\downarrow 5.9}$	10.4$^{\downarrow 0.1}$	53.2$^{\downarrow 2.6}$	32.7$^{\downarrow 2.2}$	84.3$^{\downarrow 0.3}$	39.7$^{\downarrow 0.6}$	49.3$^{\downarrow 0.4}$	80.3$^{\downarrow 2.3}$	63.4$^{\downarrow 0.9}$
+Uncertainty Sampling	49.2$^{\uparrow 0.1}$	18.6$^{\downarrow 5.4}$	9.4$^{\downarrow 1.1}$	52.4$^{\downarrow 3.4}$	32.4$^{\downarrow 2.5}$	84.2$^{\downarrow 0.4}$	40.0$^{\downarrow 0.3}$	49.6$^{\downarrow 0.1}$	81.4$^{\downarrow 1.2}$	63.8$^{\downarrow 0.5}$
+GALA (Ours)	**49.5**$^{\uparrow 0.4}$	**24.5**$^{\uparrow 0.5}$	10.6$^{\uparrow 0.1}$	55.6$^{\uparrow 0.2}$	35.1$^{\uparrow 0.2}$	**84.7**$^{\uparrow 0.1}$	40.6$^{\uparrow 0.3}$	50.6$^{\uparrow 0.9}$	83.3$^{\uparrow 0.7}$	64.8$^{\uparrow 0.5}$
+EAST	49.2$^{\uparrow 0.1}$	24.2$^{\uparrow 0.2}$	10.7$^{\uparrow 0.2}$	56.0$^{\uparrow 0.2}$	35.0$^{\uparrow 0.1}$	84.4$^{\downarrow 0.2}$	40.3	49.8$^{\uparrow 0.1}$	82.8$^{\uparrow 0.2}$	64.3
+EAST+GALA (Ours)	49.4$^{\uparrow 0.3}$	24.4$^{\uparrow 0.4}$	**11.0**$^{\uparrow 0.5}$	**56.2**$^{\uparrow 0.4}$	**35.3**$^{\uparrow 0.4}$	84.6	**40.7**$^{\uparrow 0.4}$	**50.9**$^{\uparrow 1.2}$	**83.5**$^{\uparrow 0.9}$	**64.9**$^{\uparrow 0.6}$
DPO										
Full Data	49.0	20.1	10.6	53.5	33.3	84.4	41.0	49.7	81.4	64.1
+Random Sampling	48.8$^{\downarrow 0.2}$	18.4$^{\downarrow 1.7}$	11.2$^{\uparrow 0.6}$	54.3$^{\uparrow 0.8}$	33.2$^{\downarrow 0.1}$	83.2$^{\downarrow 1.2}$	40.1$^{\downarrow 0.9}$	48.5$^{\downarrow 1.2}$	80.8$^{\downarrow 0.6}$	63.2$^{\downarrow 0.9}$
+Uncertainty Sampling	48.7$^{\downarrow 0.3}$	18.1$^{\downarrow 2.0}$	10.5$^{\downarrow 0.1}$	53.1$^{\downarrow 0.4}$	32.6$^{\downarrow 0.7}$	84.5$^{\uparrow 0.1}$	39.9$^{\downarrow 1.1}$	48.2$^{\downarrow 1.5}$	80.8$^{\downarrow 0.6}$	63.4$^{\downarrow 0.7}$
+GALA (Ours)	**49.1**$^{\uparrow 0.1}$	20.2$^{\uparrow 0.1}$	11.3$^{\uparrow 0.7}$	54.3$^{\uparrow 0.8}$	33.7$^{\uparrow 0.4}$	**84.7**$^{\uparrow 0.3}$	40.9$^{\downarrow 0.1}$	49.8$^{\uparrow 0.1}$	82.0$^{\uparrow 0.6}$	64.4$^{\uparrow 0.3}$
+EAST	48.8$^{\downarrow 0.2}$	20.3$^{\uparrow 0.2}$	10.8$^{\uparrow 0.2}$	53.8$^{\uparrow 0.3}$	33.4$^{\uparrow 0.1}$	84.4	41.1$^{\uparrow 0.1}$	49.8$^{\uparrow 0.1}$	81.5$^{\uparrow 0.1}$	64.2$^{\uparrow 0.1}$
+EAST+GALA (Ours)	49.0	**20.5**$^{\uparrow 0.4}$	**11.4**$^{\uparrow 0.8}$	**54.4**$^{\uparrow 0.9}$	**33.8**$^{\uparrow 0.5}$	**84.7**$^{\uparrow 0.3}$	**41.2**$^{\uparrow 0.2}$	**49.9**$^{\uparrow 0.2}$	**82.2**$^{\uparrow 0.8}$	**64.5**$^{\uparrow 0.4}$

proceeds with the gradient computation and parameter update for a given mini-batch $\mathcal{B}_i$ if it passes the quality gate as:

$$\text{Update with } \mathcal{B}_i \quad \text{if} \quad \frac{1}{|\mathcal{B}_i|} \sum_{r \in \mathcal{B}_i} Q(r) \geq \tau. \quad (8)$$

This dynamic gating process is computationally efficient and effectively prevents harmful data from degrading the performance. The final parameter update for our GALA includes minimizing the loss objective only on the union of accepted mini-batches, i.e., $\mathcal{D}^*_{\text{train}}$. We also implement DPO (Rafailov et al., 2023) in our experiments.

4 Experiments

To rigorously evaluate the efficacy of our proposed GALA, we conduct a comprehensive set of experiments to evaluate assess its performance and data efficiency. We compare our proposed GALA against a range of competing baselines on standard mathematical reasoning benchmarks. Furthermore, we perform detailed ablation studies to measure the contribution of each component in our pipeline and analyze our data generated by the proposed GALA.

4.1 Experimental Setup

Models. We evaluate our proposed GALA on two representative open-source large language models from the Llama family (Dubey et al., 2024): *Llama-3.1-8B-Instruct* and *Llama-3.2-1B-Instruct*.

Datasets. We adopt several widely used mathematical reasoning benchmarks including *GSM8K* (Cobbe et al., 2021), *MATH* dataset (Hendrycks et al., 2021), *MMLU-Pro-MATH* (Wang et al., 2024) and *ASDiv* (Miao et al., 2020) for self-training and validation. Their details can be found in the Appendix.

Baselines. We compare our proposed approach GALA, against three series of baselines to demonstrate its effectiveness: (1) a *full-data reference baseline*, which involves fine-tuning on the entire set of verified generated reasoning paths ($\mathcal{D}_{\text{train}}$) as described in Section 2.1; (2) *data selection baselines*, including *Random Sampling*, a classic active learning strategy via *Uncertainty Sampling* (selecting data with the lowest model probability; Nguyen et al., 2022; Liu and Li, 2023), and a heuristic-based pipeline from *NeMo Curator* (Jennings et al., 2024); and (3) a state-of-the-art *data weighting baseline EAST* (Wang et al., 2025b).

Implementation Details. All models are trained using the *AdamW* optimizer with *bf16* precision for computational efficiency. For the larger *Llama-3.1-8B-Instruct* model, we employ the popular Low-Rank Adaptation (LoRA) (Hu et al., 2022) to make fine-tuning more memory-efficient. All experiments are conducted on 4x NVIDIA A800. The initial data generation template follows the pipeline of EAST (Wang et al., 2025b). To ensure statistical robustness, we report the mean accuracy over three separate runs, each using a different random seed. In practice, we determine τ by running a pilot evaluation on each dataset and setting it to the median quality score of the samples to provides a dataset-adaptive threshold.

4.2 Performance Comparison

The compared performance of different approaches on the benchmark datasets can be found in Table 1. From the results, we have the following observation. Firstly, our proposed GALA greatly improves

Table 2: Ablation study (SFT Results) on the GSM8K and MATH test set using Llama-3.1-8B.

Configuration	GSM8K (%)	MATH (%)
GALA (Full Pipeline)	84.7	40.6
GALA w/o Clustering	84.4$^{\downarrow 0.3}$	40.5$^{\downarrow 0.1}$
GALA w/o Augmentation	84.3$^{\downarrow 0.4}$	39.6$^{\downarrow 1.0}$
GALA w/o Validation	84.3$^{\downarrow 0.4}$	39.8$^{\downarrow 0.8}$

data efficiency while maintaining competitive accuracy. In particular, for Llama-3.1-8B on SFT, GALA essentially matches the accuracy of the Full Data baseline while using much less training data. Secondly, when reducing the sample size using random sampling and uncertainty sampling, the performance on the benchmark would generally drop, validating that data is the core of LLM post-training. Thirdly, we can observe that our proposed GALA outperforms a range of competing baselines such as random sampling and uncertainty sampling with the same sample size, which validates the superiority of our proposed GALA. The potential reason is that our strategy trajectory exploration module can make the best of the limited samples with high-quality reasoning paths.

Table 3: Ablation study (DPO Results) on the GSM8K and MATH test set using Llama-3.1-8B.

Configuration	GSM8K (%)	MATH (%)
GALA (Full Pipeline)	84.7	40.9
GALA w/o Clustering	84.5$^{\downarrow 0.2}$	40.5$^{\downarrow 0.4}$
GALA w/o Augmentation	84.3$^{\downarrow 0.4}$	40.1$^{\downarrow 0.8}$
GALA w/o Validation	84.5$^{\downarrow 0.2}$	40.2$^{\downarrow 0.7}$

4.3 Ablation Studies

To validate the contribution of different components within the proposed GALA, we then include extensive ablation studies. In particular, we systematically remove one component at a time and observe the impact on final performance. The model variants include (1) GALA *w/o Clustering*, which removes the clustering process and anchor point selection; (2) GALA *w/o Augmentation*, which removes the strategy brainstorming and reflection; and (3) GALA *w/o Validation*, which removes the final real-time quality gate.

The results can be found in Table 2 and Table 3. From the results, we have the following observations as follows. First, GALA *w/o Clustering* performs worse than the full model, which validates the superiority of our clustering-based selection. Second, our full model outperforms GALA *w/o Augmentation*, which demonstrates the effectiveness of our strategic trajectory exploration module. Thirdly, removing our real-time quality gate would result in a decrease in performance, which validates that our gating mechanism to mitigate the potential low-quality data for better performance.

Table 4: The compared clarity in different methods.

Methods	Teacher Clarity Score (1-5)
Llama-3.1-8B-Instruct	4.5
+ GALA (SFT)	4.7
+ GALA (DPO)	4.8

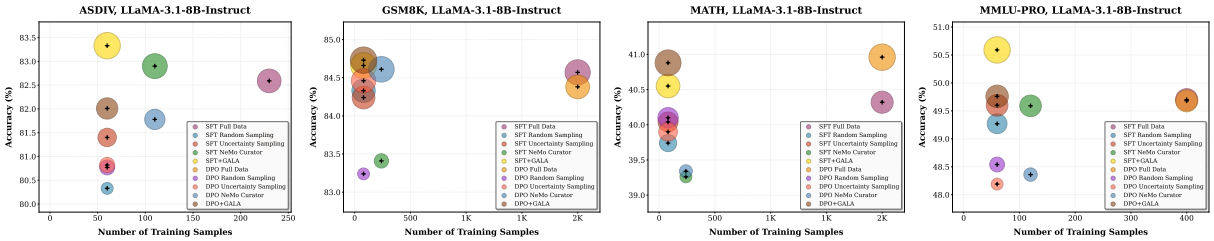
4.4 Further Analysis

Here, we provide further evidence ana analysis to support our claims regarding our module design.

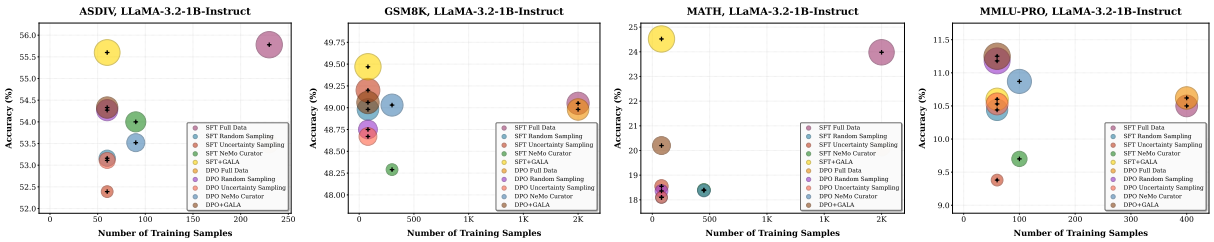
► **Efficiency Analysis.** In the context of self-training, each selected sample for self-training can be considered labeled data. Therefore, GALA demonstrates superior label efficiency with limited data. To validate this, we report the accuracy and number of training samples in Figure 3. From the results, we can observe that our proposed GALA reaches competing performance with a fraction of the data required by the full-data baseline. In particular, our proposed GALA achieves a target performance level with a dataset of a very limited number of samples, whereas the naive baseline requires fine-tuning on the full set of samples to approach similar performance, indicating a huge improvement in label efficiency.

► **Data Quality Analysis.** In this part, we study the data clarity of different methods. The compared results can be found in Table 4. From the results, we can observe that our proposed GALA produces better solutions with higher clarity scores compared with the baseline.

► **Data Diversity Analysis.** Our strategy brainstorming module uses Strategy Brainstorming to ensure each solution path is strategically unique. We aim to validate the effectiveness of this strategy. The evaluation is performed across different datasets. For each problem, the model generates 12 outputs, and the final score is the average of the per-problem Self-BLEU (Zhu et al., 2018) scores. We report the results in Table 5. From the results, we can observe that the synthetic data curated by GALA exhibits a significantly lower score in SELF-BLEU compared to Naive QA synthetic (Lewis et al., 2021) and Paraphrase (Allen-Zhu and Li, 2023). This confirms that our strategy successfully generates a diverse dataset to facilitate post-training.



(a) ASDIV, LLaMA-3.1-8B-Instruct. (b) GSM8K, LLaMA-3.1-8B-Instruct. (c) MATH, LLaMA-3.1-8B-Instruct. (d) MMLU-PRO, LLaMA-3.1-8B-Instruct.



(e) ASDIV, LLaMA-3.2-1B-Instruct. (f) GSM8K, LLaMA-3.2-1B-Instruct. (g) MATH, LLaMA-3.2-1B-Instruct. (h) MMLU-PRO, LLaMA-3.2-1B-Instruct.

Figure 3: Performance (Accuracy) vs. Training Data Size on the GSM8K, MATH, ASDIV and MMLU-PRO test set. Our GALA demonstrates competitive performance with fewer samples compared to both baselines.

Table 5: Evaluation of output diversity using the Self-BLEU score on synthetic data using Llama-3.1-8B. A lower Self-BLEU score indicates better diversity.

Method	Self-BLEU (Lower is better)		
	MATH	GSM8K	MMLU-PRO-MATH
Naive QA (Baseline)	0.72	0.77	0.69
Paraphrase	0.74	0.82	0.73
GALA (Ours)	**0.67**	**0.74**	**0.63**

5 Related Work

5.1 Self-Instruct Large Language Models

LLM self-training has become a popular direction for enhancing model capabilities. Recent work (Wang et al., 2022) demonstrates that a model could use its own generations filtered by certain criteria to enhance its instruction following abilities (Mondorf and Plank, 2024; Truhn et al., 2023; Mirzadeh et al., 2024). One line to solve the problem is to generate positive examples that match the ground truth for fine-tuning. (Zelikman et al., 2022; Singh et al., 2023; Luong et al., 2024). In addition, several works incorporate negative samples (Sun et al., 2024; Saeidi et al., 2024; Zhong et al., 2024; Ivison et al., 2024) and meta reinforcement learning (Qu et al., 2025; Sun et al., 2025b; Zuo et al., 2025) into the fine-tuning process. In this work, we propose a new approach named GALA for LLM self-training.

5.2 Data-Efficient LLM Training

Data-efficient LLM training has received extensive attention recently due to its ability to handle serious data challenges (Ye et al., 2025; Li et al., 2025). Several works have shown that the quality of the training data is critical to the performance of LLMs (Li et al., 2024; Sachdeva et al., 2024; Zhang et al., 2025b). Furthermore, a range of researchers have extended data selection approaches (Sener and Savarese, 2017; Nguyen et al., 2022; Xia et al., 2025) into solving supervised fine-tuning and reinforcement learning (Freitas and Curry, 2016; Lambert et al., 2024; Seedat et al., 2024; Zhang et al., 2026; Lin et al., 2025a; Ding et al., 2024; Wang et al., 2025b). On this basis, test-time reinforcement learning aims to make the best of unlabeled data for unsupervised learning, which saves the burdens of human annotation (Wang et al., 2025a; Zuo et al., 2025). In comparison to these works, our proposed GALA combines geometric data selection with strategic prospecting, which achieves competing performance with limited training samples.

6 Conclusion

In this work, we introduce GALA for LLM self-training. Our GALA aims to address the critical challenges of data redundancy and low strategic diversity in self-training methods for LLMs. Our experiments on various benchmarks demonstrate that GALA achieves superior performance while sig-

nificantly improving data efficiency compared to standard baselines. In future work, we aim to extend our proposed GALA to more real-world applications, such as medical and geographic foundation models.

Limitations

Despite the promising efficiency and performance gains, one limitation of our work is our data enhancement strategy. In particular, our data enhancement strategy still risks introducing noise through brainstorming. Concurrently, our mini-batch validation can be noisy. Future work should focus on guiding reasoning depth using a reward model to further improve LLM self-training.

Acknowledgments

We would like to thank the anonymous reviewers and meta-reviewers for their insightful comments.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. 2025. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

Chitta Baral, Michael Gelfond, and Nelson Rushton. 2009. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144.

Hung-Fu Chang and Tong Li. 2025. A framework for collaborating a large language model tool in brainstorming for triggering creative thoughts. *Thinking Skills and Creativity*, 56:101755.

Nuo Chen, Yicheng Tong, Jiaying Wu, Minh Duc Duong, Qian Wang, Qingyun Zou, Bryan Hooi, and Bingsheng He. 2025. Beyond brainstorming: What drives high-quality scientific ideas? lessons from multi-agent collaboration. *arXiv preprint arXiv:2508.04575*.

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*.

Jairo Diaz-Rodriguez. 2025. k-llmmeans: scalable, stable, and interpretable text clustering via llm-based centroids. *arXiv preprint arXiv:2502.09667*.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Animashree Anandkumar, et al. 2024. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *Advances in Neural Information Processing Systems*, 37:44323–44365.

Jiawei Du, Juncheng Hu, Wenxin Huang, Joey Tianyi Zhou, et al. 2024. Diversity-driven synthesis: Enhancing dataset distillation through directed weight adjustment. *Advances in neural information processing systems*, 37:119443–119465.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

André Freitas and Edward Curry. 2016. Big data curation. In *New horizons for a data-driven economy: A roadmap for usage and exploitation of big data in Europe*, pages 87–118. Springer International Publishing Cham.

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. *NeurIPS*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hanna Hajishirzi. 2024. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *Advances in neural information processing systems*, 37:36602–36633.

Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Shrimai Prabhumoye, Ayush Dattagupta, Vibhu Jawa, Jiwei Liu, Ryan Wolf, Sarah Yurick, and Varun Singh. 2024. NeMo-Curator: a toolkit for data curation.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*.

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

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Rui Li, Deji Fu, Chunyu Shi, Zhilan Huang, and Gang Lu. 2024. Efficient llms training and inference: An introduction. *IEEE Access*.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*.

Yisen Li, Lingfeng Yang, Wenxuan Shen, Pan Zhou, Yao Wan, Weiwei Lin, and Dongping Chen. 2026. Crowdselect: Syntheticinstruction data selection with multi-llm wisdom. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1542–1569.

Jessy Lin, Vincent-Pierre Berges, Xilun Chen, Wen-Tau Yih, Gargi Ghosh, and Barlas Oğuz. 2025a. Learning facts at scale with active reading. *arXiv preprint arXiv:2508.09494*.

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

Shang Liu and Xiaocheng Li. 2023. Understanding uncertainty sampling. *arXiv preprint arXiv:2307.02719*.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2023. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*.

Keer Lu, Xiaonan Nie, Zheng Liang, Da Pan, Shusen Zhang, Keshi Zhao, Weipeng Chen, Zenan Zhou, Guosheng Dong, Bin Cui, et al. 2024. Datasculpt: Crafting data landscapes for long-context llms through multi-objective partitioning. *arXiv preprint arXiv:2409.00997*.

Junyu Luo, Xiao Luo, Xiusi Chen, Zhiping Xiao, Wei Ju, and Ming Zhang. 2024. Semi-supervised fine-tuning for large language models. *arXiv preprint arXiv:2410.14745*.

Junyu Luo, Xiao Luo, Xiusi Chen, Zhiping Xiao, Wei Ju, and Ming Zhang. 2025. Semi-supervised fine-tuning for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2795–2808.

Xiao Luo, Wei Ju, Yiyang Gu, Yifang Qin, Siyu Yi, Daqing Wu, Luchen Liu, and Ming Zhang. 2023. Toward effective semi-supervised node classification with hybrid curriculum pseudo-labeling. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(3):1–19.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*.

Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Xinyu Pan, Di Wang, and Fugee Tsung. 2025. Empowering intelligent quality control with large models: A comprehensive survey of methods, challenges, and perspectives. *Authorea Preprints*.

Chong Peng, Zhao Kang, Huiqing Li, and Qiang Cheng. 2015. Subspace clustering using log-determinant rank approximation. In *Proceedings of the 21th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 925–934.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. 2024. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*.

Amir Saeidi, Shivanshu Verma, Md Nayem Uddin, and Chitta Baral. 2024. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *arXiv preprint arXiv:2404.14723*.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. 2024. Curated LLM: Synergy of LLMs and data curation for tabular augmentation in low-data regimes. In *Forty-first International Conference on Machine Learning*.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Junyou Su, He Zhu, Xiao Luo, Liyu Zhang, Hong-Yu Zhou, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. 2026. Instructdiff: Domain-adaptive data selection via differential entropy for efficient llm fine-tuning. *arXiv preprint arXiv:2601.23006*.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. 2025a. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*.

Zexu Sun, Yongcheng Zeng, Erxue Min, Heyang Gao, Bokai Ji, and Xu Chen. 2025b. Cog-rethinker: Hierarchical metacognitive reinforcement learning for llm reasoning. *arXiv preprint arXiv:2510.15979*.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. *Advances in Neural Information Processing Systems*, 37:51118–51168.

Daniel Truhn, Jorge S Reis-Filho, and Jakob Nikolas Kather. 2023. Large language models should be used as scientific reasoning engines, not knowledge databases. *Nature medicine*, 29(12):2983–2984.

Bo Wang, Mingda Chen, Ming Deng, Youfang Lin, Mark Harman, Mike Papadakis, and Jie M Zhang. 2026. A comprehensive study on large language models for mutation testing. *ACM Transactions on Software Engineering and Methodology*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Xiaoxuan Wang, Yihe Deng, Mingyu Derek Ma, and Wei Wang. 2025b. Entropy-based adaptive weighting for self-training. *arXiv preprint arXiv:2503.23913*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yu Xia, Subhojyoti Mukherjee, Zhouhang Xie, Junda Wu, Xintong Li, Ryan Aponte, Hanjia Lyu, Joe Barrow, Hongjie Chen, Franck Dernoncourt, et al. 2025. From selection to generation: A survey of llm-based active learning. *arXiv preprint arXiv:2502.11767*.

Zhiping Xiao, Yusheng Zhao, Qixin Zhang, Jiaye Xie, Wanjia Zhao, Weizhi Zhang, Xiao Luo, Philip S Yu, and Ming Zhang. 2026. Sample lottery: Unsupervised discovery of critical instances for llm reasoning. In *The Fourteenth International Conference on Learning Representations*.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E Gonzalez, Bin Cui, and Shuicheng Yan. 2024. Supercorrect: Advancing small llm reasoning with thought template distillation and self-correction. *arXiv preprint arXiv:2410.09008*.

Wang Yang, Zirui Liu, Hongye Jin, Qingyu Yin, Vipin Chaudhary, and Xiaotian Han. 2025. Longer context, deeper thinking: Uncovering the role of long-context ability in reasoning. *arXiv preprint arXiv:2505.17315*.

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

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Self-taught reasoner. In *Proceedings of the NIPS*, volume 22.

Jenny Zhang, Shengran Hu, Cong Lu, Robert T. Lange, and Jeff Clune. 2025a. Darwin godel machine: Open-ended evolution of self-improving agents. *CoRR*, abs/2505.22954.

Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. 2025b. Lori: Reducing cross-task interference in multi-task low-rank adaptation. *arXiv preprint arXiv:2504.07448*.

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. 2026. Beyond random sampling: Efficient language model pretraining via curriculum learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5776–5794.

Wanru Zhao, Hongxiang Fan, Shell Xu Hu, Wangchunshu Zhou, Bofan Chen, and Nicholas D Lane. 2025. Clues: Collaborative high-quality data selection for llms via training dynamics. *arXiv preprint arXiv:2507.03004*.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*.

Zhi Zhou, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Xiaoxing Ma, and Yu-Feng Li. 2025. Bridging internal probability and self-consistency for effective and efficient llm reasoning. *arXiv preprint arXiv:2502.00511*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Yuchang Zhu, Huizhe Zhang, Bingzhe Wu, Jintang Li, Zibin Zheng, Peilin Zhao, Liang Chen, and Yatao Bian. 2025. Measuring diversity in synthetic datasets. *arXiv preprint arXiv:2502.08512*.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. 2025. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.

A Training Dataset Details

We used widely used mathematical reasoning benchmarks such as for training and validation.

- **MATH** (Hendrycks et al., 2021): This dataset contains 12,500 competition-level problems from sources such as AMC and AIME, spanning seven mathematical subjects and five difficulty levels. We use the full training set and test set for training and evaluation.
- **GSM8K** (Cobbe et al., 2021): GSM8K is a dataset of 8,500 grade-school math word questions collected by human problem writers. We use the full training set and test set for training and evaluation.
- **MMLU-Pro-MATH** (Wang et al., 2024): MMLU-Pro-MATH is the math subset of MMLU-Pro dataset for data enhancement. MMLU-Pro dataset features 12,000 complex questions spanning various disciplines, including 1,356 mathematical problems. We randomly sample 400 problems for training and use the rest for evaluation.
- **ASDiv** (Miao et al., 2020): ASDiv dataset features 2,305 complex mathematical questions. We randomly sample 230 problems for training and use the rest for evaluation.

B Prompt Examples

This appendix provides concrete examples of the prompt templates used for the data augmentation strategies described in Section 3.3 and Teacher Clarity Score Prompt.

Prompt Example for Multi-Strategy Solution Generation

Part A: Strategy Brainstorming Prompt

Instruction:
You are a creative and experienced mathematics teacher. For the problem below, your task is to first brainstorm diverse reasonable strategies for solving it. Do NOT solve the problem yet. For each strategy, provide a name and a brief description of its core logic.
Problem:
`{{problem_text}}`

Part B: Guided Demonstration Prompt (using one of the brainstormed strategies)

Instruction:
You are a helpful AI assistant. Please solve the problem below by strictly following the specific strategy provided. Show your reasoning and calculations step-by-step.
Problem:
`{{problem_text}}`
Strategy to use:
`{{strategy_text}}`

Prompt Example for Reflective Enhancement

Part A: Ensemble Reflection Prompt

Instruction:
You are a logical analyst. Below are three different attempts to solve the same problem. Some may contain errors or be inefficient. Your task is to analyze all three, identify the most effective and correct logical steps, and produce a concise summary of the optimal solution strategy. Do not solve the problem yet.
Problem:
`{{problem_text}}`
Solution Attempt 1:
`{{solution_1_text}}`
Solution Attempt 2:
`{{solution_2_text}}`
Solution Attempt 3:
`{{solution_3_text}}`
Your Summary of the Core Logic:

Part B: Guided Regeneration Prompt

Instruction:
Now, using only the "Core Logic" you just summarized, provide the final, correct, and clear step-by-step solution to the problem.
Problem:
`{{problem_text}}`
Core Logic Summary:
`{{summary_from_part_a}}`
Final Enhanced Solution:

Evaluation Prompt: Teacher Clarity Score (1.0–5.0 with Decimals)

Role:
You are a Senior Pedagogical Consultant and Mathematics Educator. Your task is to perform a fine-grained, blind evaluation of the "Teacher Clarity" of a provided solution for a Grade School Math problem.
Instruction:
Evaluate the pedagogical effectiveness of the solution. You are required to provide a decimal score to capture subtle differences in clarity. Assess the solution based on three weighted dimensions: (1) **Logical Scaffolding (40%):** The flow and transition between steps. (2) **Explanatory Depth (40%):** The clarity of the "why" behind each calculation. (3) **Accessibility (20%):** Use of student-friendly language and intuitive framing.
Scoring Anchors (Reference Levels):

1.0 (Very Poor): Fragments of logic; mathematically misleading; no pedagogical value.

2.0 (Poor): A raw list of numbers/operations with minimal context; high cognitive load.

3.0 (Acceptable): Technically sound step-by-step solution; functional but mechanical.

4.0 (Good): Clear structure with purposeful explanations and transitional phrasing.

5.0 (Masterful): Flawless pedagogical flow; anticipates confusion; exceptionally intuitive.

Decimal Scoring Logic:
Start from an integer anchor and adjust by ± 0.1 to 0.9 based on minor strengths or weaknesses.
Required Output Format:
[Teacher Clarity Score]: <A decimal value between 1.0 and 5.0>

Problem:
`{{problem_text}}`