

Dynamic Sampling that Adapts: Self-Aware Iterative Data Persistent Optimization for Mathematical Reasoning

Jun Rao¹, Xuebo Liu^{1*}, Hexuan Deng¹, Zepeng Lin¹,
Zixiong Yu², Jiansheng Wei², Xiaojun Meng^{2*}, and Min Zhang¹

¹Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China

²Huawei Large Model Data Technology Lab

{rao7jun, zepenglin11, hxuandeng}@gmail.com,

{liuxuebo, zhangmin2021}@hit.edu.cn,

yuzx19@tsinghua.org.cn, {weijiansheng, xiaojun.meng}@huawei.com

Abstract

In mathematical reasoning, data selection strategies predominantly rely on static, externally defined metrics, which fail to adapt to the evolving capabilities of models during training. This misalignment limits the efficiency of Supervised Fine-Tuning and Reinforcement Learning. To bridge this gap, we introduce SAI-DPO (Self-Aware Iterative Data Persistent Optimization), a dynamic sampling framework that aligns training data with the model’s intrinsic competence. SAI-DPO operationalizes two novel metrics: Knowledge Semantic Alignment for targeting domain weaknesses, and Self-Aware Difficulty, derived from pass rates and reasoning path characteristics, to gauge instance complexity relative to the model’s current state. By iteratively recalibrating the data distribution based on real-time feedback, SAI-DPO dynamically aligns training samples with the model’s evolving competence, ensuring the data remains strictly relevant to the model’s current capability level. Extensive experiments on eight benchmarks (including AIME24 and AMC23) demonstrate that SAI-DPO outperforms static baselines at most nearly 6 points, achieving state-of-the-art efficiency with significantly less data.

1 Introduction

Recent advances in Large Language Models (LLMs), particularly in reasoning tasks (Jaech et al., 2024; Guo et al., 2025), highlight the critical role of high-quality data. However, current data selection paradigms remain largely static, relying on fixed datasets or external difficulty scorers. This creates a fundamental disconnection: as a model learns, what was once “hard” becomes “easy”, rendering static datasets progressively inefficient. Continued training on trivial samples yields diminishing returns, while overly complex samples may induce hallucinations.

* Corresponding Authors.

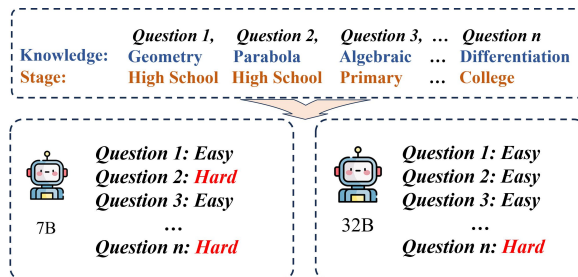


Figure 1: The difficulty levels and knowledge scopes of problems solvable by different models vary significantly. Basic knowledge points are manageable for small models, while complex or advanced content poses challenges. Even for a large model, such advanced problems (Differentiation) remain difficult, highlighting the importance of aligning difficulty with capabilities. Therefore, the difficulty definition should rely on the current capabilities of the model.

Current works mainly focus on Supervised Fine-Tuning (SFT) (Taori et al., 2023; Rao et al., 2024; Zhou et al., 2023; Liu et al., 2024b; Zeng et al., 2025c; Jiang et al., 2025) after data filtering (Muenighoff et al., 2025; Ye et al., 2025) or online reinforcement learning algorithms (Shao et al., 2024; Schulman et al., 2017; Deng et al., 2026). Most of these methods are static, failing to adaptively select suitable data for continuous training based on the model’s current capabilities, thereby limiting the sustainable improvement of its reasoning abilities.

As shown in Figure 1, different models have varying capabilities, thus leading to differences in their discrimination of the questions. Although some existing works (Team et al., 2025; Zhou et al., 2026) have addressed the impact of difficulty on models (Ke et al., 2025; Liu et al., 2024b), the related metrics remain unclear. To address the issue of the lack of dynamic adaptive training for reasoning data, we propose the SAI-DPO (Self-Aware Iterative Data Persistent Optimization) algorithm for mathematical reasoning. This algorithm dynamically selects training data that matches the

model’s current competence (*Self-Aware Difficulty*) and weaknesses (*Knowledge Semantic Alignment*), enhancing its reasoning abilities through iterations. Using the defined metrics, the algorithm dynamically selects data and filters low-quality inputs to enhance training efficiency.

We conducted extensive experiments to explore the defined metric, data acquisition strategy, and the gradual improvement through iterative training. The experiments were carried out on 8 existing public mathematical test sets and 4 public models (Qwen2.5-7B-Math-Base, Qwen2.5-Math-7B-SFT, Llama3.1-8B-Instruct and Qwen3-8B). Our approach not only achieves better performance compared to the original DPO but also accelerates the training process. And compared to some current common strategies, such as externally defined difficulty (Ye et al., 2025) and curriculum learning (Chen et al., 2025; Hong et al., 2025), our strategy has better results. Our results show that externally defined difficulty does not align with what is difficult for the model, and it is better to train with the model’s defined difficulty. Our main contributions are as follows:

- We propose a Dynamic Data Acquisition strategy that clusters knowledge tags to systematically target specific weakness domains.
- We formulate a Self-Aware Difficulty Metric that integrates statistical priors (pass rate) with cognitive load indicators (step count and length), providing a nuanced view of model competence.
- We demonstrate through extensive experimentation that aligning data difficulty with model capability yields superior performance, improving accuracy on competition-level benchmarks (AIME24 and AMC23) by nearly 4 points over strong baselines.

2 Related Work

2.1 Post-training Preference Optimization

In the post-training stage, many RL algorithms improve model performance by aligning the model’s output objectives with human preferences—specifically, by increasing the probability of generating high-quality responses and decreasing the probability of producing low-quality ones. A common algorithm is Proximal Policy Optimization (PPO) (Schulman et al., 2017), which has been applied in multiple current LLM systems (Ouyang

et al., 2022; Dubey et al., 2024) . Recently, more powerful reasoning models such as KIMI K1.5 (Team et al., 2025), Deepseek V3 (Liu et al., 2024a), and R1 (Guo et al., 2025) have made modifications to PPO, giving rise to algorithms like GRPO (Shao et al., 2024) and REINFORCE++ (Hu et al., 2025). Although these algorithms have shown good performance, their practical deployment is often complicated due to the time-consuming nature of the online exploration involved. In contrast, some offline methods (Rafailov et al., 2023; Ethayarajh et al., 2024; Zhang et al., 2025) are simpler to deploy. Direct Preference Optimization (DPO) efficiently trains large models for knowledge alignment using preference rankings instead of reward models. DPO optimizes classification loss from preference data, making implementing it simpler than RL from human feedback. Some papers (Guo et al., 2024; Pang et al., 2024; Rao et al., 2025b,a) collectively advance LLM alignment by shifting from static datasets to iterative self-improvement, demonstrating that dynamic, online feedback loops and repeated preference optimization significantly boost both general instruction following and complex reasoning capabilities. SPHERE (Singh et al., 2025), IDPO (Tu et al., 2025) employs a self-evolving, iterative data augmentation approach for mathematical reasoning, called Online DPO. Unlike existing work, we improve the effectiveness through the model’s self-judgment of the current data selection, rather than the algorithm.

2.2 Post-training Data Strategies

Data plays a crucial role in unlocking the capabilities of models (Rao et al., 2025c; Li et al., 2024b; Han et al., 2025; Yu et al., 2026; Zhang et al., 2026). In the early days, the LIMA (Zhou et al., 2023) found that a small amount of data could activate the relevant capabilities of the model and improve the test results of multiple tasks. Recently, some data selections in the field of mathematics have also demonstrated the importance of data quality and diversity. For instance, selections like S1 (Muennighoff et al., 2025) and LIMO (Ye et al., 2025), which used a small amount of data, managed to stimulate the mathematical reasoning capabilities of the models. KIMI K1.5 (Team et al., 2025) adopted curriculum learning and constructed a curriculum-based data training strategy. Pangu Ultra (Yin et al., 2025) assigned quality and difficulty labels to the data and also used a curriculum-

based sampling strategy throughout its three pre-training stages. In this work, we explored an approach to dynamic data training during the training process, aiming to enhance the final RL performance by selecting training data that is aligned with the model’s own competency.

3 Methods

3.1 Overview

In Figure 2, the system operates in cycles. At iteration t , the current model M_t acts as a probe to evaluate the training data pool \mathcal{D} . By leveraging two distinct metrics: Knowledge Semantic Alignment and Self-Aware Difficulty, we construct a dynamic curriculum $\mathcal{D}_{train}^{(t)}$ that targets the model’s specific weaknesses. The model is then updated via preference optimization to yield M_{t+1} , shifting the difficulty frontier for the subsequent cycle.

3.2 Metric Definition

To operationalize dynamic data selection, we introduce two complementary metrics: one for semantic coverage (what the problem is about) and one for intrinsic complexity (how hard it is for the current model), shown in Figure 2 (Phase 1).

3.2.1 Knowledge Semantic Alignment

Effective training requires diversity across mathematical concepts. We treat knowledge identification as a Latent Semantic Clustering problem. **Annotation:** We employ an expert model to generate explicit knowledge tags $T(x)$ for each instance x (e.g., Geometry, Sequence Summation). **Embedding and Clustering:** These tags are mapped into a vector space using Sentence-Transformers (all-MiniLM-L6-v2)¹. We then apply K-Means clustering to partition the dataset into n semantic domains $C = \{C_1, C_2, \dots, C_n\}$. This granular partitioning allows us to detect and up-sample specific domains where the model exhibits high error rates. We present some examples in Appendix A.2.

3.2.2 Self-Aware Difficulty Calibration

Unlike external difficulty scorers which are static, we define difficulty as a function of the model’s interaction with the data. We propose a **Hierarchical Difficulty Metric** composed of three dimensions:

1). Probabilistic Solvability (NoP, Primary):

We perform K explorations for each query. The Number of Passes (NoP), defined as the count of correct responses, serves as the primary proxy for difficulty. A lower NoP indicates higher aleatoric uncertainty and difficulty. We define the “solvable range” as problems where the model is neither consistently correct nor consistently incorrect. Specifically, this includes instances with a NoP where $0 < NoP < K$.

2). Reasoning Depth (Steps, Secondary):

For instances with identical NoP, we differentiate complexity by the number of reasoning steps. We posit that among problems with equal solvability, those necessitating longer logical chains (higher step counts) represent a higher tier of intrinsic complexity. We present examples in Appendix A.3.

3). Generation Length:

As a tertiary measure, total token length is used to resolve ties, reflecting the cognitive load of generation.

3.3 Dynamic Data Acquisition

Our goal is to shift the training distribution $P(x)$ towards regions of high model uncertainty. This is achieved through a three-step process: Error Distribution Estimation, Category-Level Re-weighting, and Instance-Level Selection.

1). Error Distribution Estimation:

Since evaluating the entire dataset \mathcal{D} at every iteration is computationally prohibitive, we employ a subset approximation strategy. **Subset Construction:** We first sample a small subset $\mathcal{D}_{sub} \subset \mathcal{D}$ that preserves the original cluster proportions. This subset acts as a practice exam to probe the current model capability. **Error Identification:** The model M_t performs K explorations on \mathcal{D}_{sub} . We filter out invalid samples (those consistently correct or consistently incorrect) and identify the Error Dataset E , defined as the top 50% most difficult instances within the solvable range. The distribution of E effectively highlights the semantic clusters where the model currently lacks competency.

2). Category-Level Re-weighting:

We use the error statistics from \mathcal{D}_{sub} to adjust the sampling probability for the full dataset. Let the dataset be partitioned into n clusters C_1, \dots, C_n , with total size $N = \sum |C_i|$.

Baseline Distribution: The initial sampling probability $P_{initial}(i)$ for category i reflects the natural

¹<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/tree/main/>

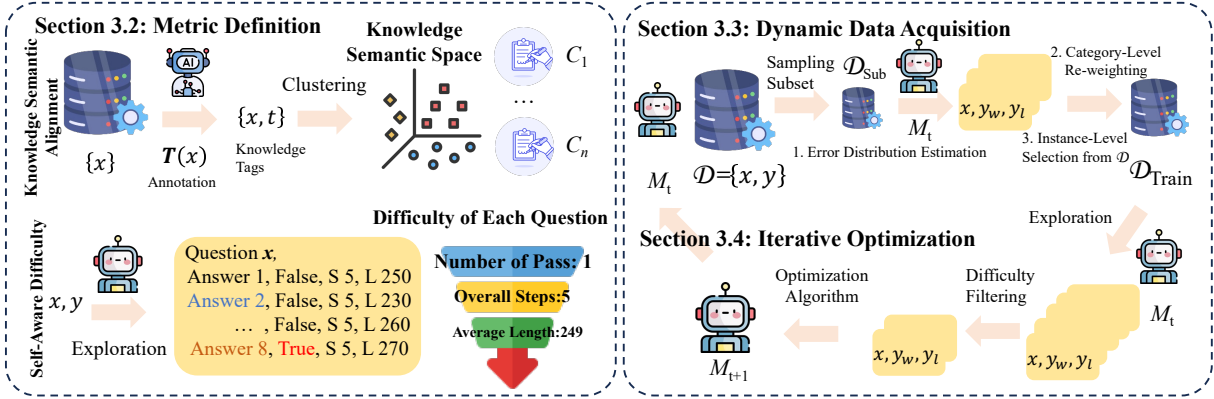


Figure 2: The SAI-DPO Framework. It operates as an iterative closed-loop consisting of three core phases: (1) Metric Calibration: We first map the dataset into a semantic space using knowledge clustering and simultaneously define instance complexity using the model’s self-aware metrics (Pass Rate, Steps, and Length). (2) Dynamic Acquisition: Based on a probing subset, we identify weakness domains and re-weight the sampling distribution $P_{adjusted}$ to prioritize high-error clusters and appropriate difficulty levels. (3) Iterative Optimization: The selected curriculum is filtered to remove trivial or intractable samples and used to update the model policy via DPO. The updated model M_{t+1} then re-evaluates the data pool, dynamically shifting the difficulty frontier for the next iteration.

data distribution:

$$P_{initial}(i) = \frac{|C_i|}{N}. \quad (1)$$

Error-Aware Adjustment: We calculate an adjusted weight $W(i)$ by amplifying categories that appear frequently in the Error Dataset E :

$$W(i) = P_{initial}(i) \times (|C_i \cap E| + 1), \quad (2)$$

where the term $(+1)$ ensures non-zero smoothing. The final sampling probability for category i becomes $P_{adjusted}(i) = W(i) / \sum W(k)$. This step systematically biases the sampling distribution towards high-error semantic domains. Equation 2 focuses on absolute error counts. This is an intentional design to ensure the model maintains performance on the most frequent topics in the distribution.

3). Instance-Level Selection: Finally, we convert category-level probabilities into instance-level selection criteria. Equation 3 mitigates bias by normalizing the final instance-level weight w_{ij} by the cluster size $|C_i|$, ensuring that individual hard problems in small clusters still receive significant attention. For each specific data point x_{ij} belonging to cluster C_i , its individual sampling weight w_{ij} is defined as:

$$w_{ij} = \frac{P_{adjusted}(i)}{|C_i|}. \quad (3)$$

To construct the final training set \mathcal{D}_{train} , we sort all available instances based on their weights w_{ij} and, secondarily, by their intrinsic difficulty scores. We select the top- S instances to form the training set for the current iteration. This mechanism ensures that the model focuses its limited training budget on difficult problems located within its weakest semantic domains.

3.4 Iterative Optimization

Difficulty Filtering We use rule-based rewards to annotate the model’s outputs to obtain correct and incorrect responses. For each problem, we randomly select one correct answer generated by the model as the positive sample and one incorrect answer as the negative sample, thereby constructing a triplet (question x , positive answer y_w , negative answer y_l) to form the training data. After obtaining the triplet, we can also filter the annotated training samples using the previously defined *Self-aware Difficulty Measurer* to derive the final training data. Typically, we first filter out data that is either all incorrect (too difficult) or all correct (too easy), then select the relatively more challenging problems (the top 70% of the filtered data sorted by difficulty) as the final training set.

Optimization Algorithm Here, we have obtained triplets of training data (question x , positive answer y_w , negative sample y_l) through the previous steps. We can apply different optimization algorithms, such as the commonly used DPO ([Rafailov](#)

et al., 2023), or directly use the SFT loss to achieve the RFT (Rejection Sampling Fine-tuning) (Dubey et al., 2024). As the model’s capabilities improve, ideally, the proportion of difficult training data will gradually decrease. Eventually, the trainable data will gradually decrease, so that the model’s performance reaches a stable value. At this point, the iterative process has reached an end.

3.5 Efficiency Discussion

Our training efficiency is higher (better results with less data), utilizing only a small amount of additional computation: 1) obtaining the distribution of the subset and 2) measuring the similarity of the overall data. For the subset, we only select 1% data in each iteration for exploratory annotation. We dynamically adjust the selection of the training set based on difficulty distribution. This part’s resource consumption is negligible compared to the total 20,000 training data samples, accounting for less than 1% of additional time. Regarding the measurement of the entire sample dataset, the model labeling we adopt only requires a single annotation process and can be reused. The additional time does not exceed 5% of the overall training process. Generation of 2,000 items takes approximately 10 minutes on our devices. Due to the adoption of DPO, the time consumption of our training method is also substantially lower than that of online RL.

4 Experiments

4.1 Setup

Baseline and Models Following previous work (Guo et al., 2024; Rafailov et al., 2023; Bai et al., 2023) of RL, we use Numina-Math (Li et al., 2024a), including 400,000 prompts, as the dataset pool for preference learning dataset construction. For fair comparison in the same environment, dataset, and parameters, we compare IDPO (Tu et al., 2025) (iterative training 8 times with random sampling of 20,000 samples per iteration) and PPO (400,000) (Schulman et al., 2017; Zeng et al., 2025a) as baselines. SAI-DPO is also conducted for 8 iterations in our main experiments. We use the Qwen2.5 series (7B-Math-Base, 7B-Math-SFT with 15K QwQ (Team, 2024) data), Qwen3-8B (Yang et al., 2025) and Llama3.1-8B-Instruct as the base models for RL for the main results. We report the average results and standard deviation of three random seeds. For other analytical experiments, we primarily report

the results related to 7B-Math-Base.

Training Details The model was trained using a learning rate of 5×10^{-7} , following a cosine decay strategy for the learning rate schedule. A per-device batch size of 1 was used during training, and to achieve an effective total batch size of 128, we employed gradient accumulation over 16 steps, which optimized memory usage during training. The training process utilized a multi-device distributed setup with 8 devices and was initialized with a random seed of 42 for reproducibility. To construct the final training set, We select the top-70% instances to form the training set for the current iteration. For optimization, we used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. Training proceeded for a total of 2 epochs, determined by the total number of training samples and the batch size. Regarding the hyperparameters of DPO, we set $\beta = 0.1$. For the rollout step in RL, we set the temperature to 1 and perform exploration 8 times, generating 8 responses for each question to obtain positive and negative samples for DPO. This value was chosen to provide a stable statistical estimate of the Pass Rate while maintaining manageable inference costs. We allow a maximum generated length of 3000.

Evaluations We follow Yang et al. (2024) using two common English math benchmarks GSM8K (Cobbe et al., 2021), Math (Hendrycks et al., 2021), and using different sets of math tests at different stages to test stronger math skills, such as Minerva Math (Lewkowycz et al., 2022), Gaokao 2023 En (Liao et al., 2024), Olympiad Bench (He et al., 2024), College Math (Tang et al., 2024), AIME 24, and AMC 23. We report greedy performance on all benchmarks in the zero-shot setting, except for the competition benchmarks (Amc23 and Aime24). Considering the limited size of Amc23 and Aime24, we sample 8 times for each question to mitigate randomness. During the answer generation process for these two datasets, we use a temperature of 0.1 and a top_p of 0.95.

4.2 Main Results

Performance across Different Models Table 1 establishes the universality of SAI-DPO across diverse model architectures (Llama-3.1, Qwen-2.5/3) and training paradigms (Base vs. SFT). Our method consistently outperforms the robust IDPO baseline across all evaluated settings, demonstrat-

Model	Method	Elementary		Middle			Competition		Avg.	
		GSM	MATH	Minerva	Gaokao	Olympiad	College	Aime24		Amc23
Llama3.1-8B-Instruct	None	69.1	29.0	16.5	31.9	7.3	21.3	0.0	15.3	23.8
	IDPO	76.9±0.6	35.8±0.3	19.4±2.0	34.5±1.0	8.5±0.9	23.8±0.5	1.4±1.7	15.1±0.2	26.9±0.4
	+SAI-DPO	77.4±0.9	36.3±0.2	19.6±0.6	35.7±0.1	10.7±0.5	25.0±0.1	4.0±0.7	20.6±1.6	28.7±0.1
Qwen2.5-7B-Math-Base	None	66.7	64.0	12.1	56.1	28.3	41.3	13.9	41.1	40.4
	IDPO	88.4±0.3	72.2±0.5	30.4±1.7	61.5±1.6	38.0±1.4	45.9±0.2	19.4±2.0	60.4±1.7	52.0±0.3
	+SAI-DPO	89.1±0.2	74.0±0.2	31.7±1.2	62.3±0.9	38.9±1.7	46.3±0.5	23.7±1.5	63.3±1.2	53.7±0.3
Qwen2.5-7B-Math-SFT	None	90.8	73.8	32.0	65.5	37.9	47.4	11.7	65.6	53.1
	IDPO	91.3±0.2	81.3±0.3	29.9±0.2	65.7±0.9	44.7±0.3	47.9±0.1	17.9±1.1	62.9±0.2	55.2±0.1
	+SAI-DPO	91.7±0.0	81.1±0.2	31.3±0.2	67.5±0.3	45.1±0.9	47.3±0.2	19.2±1.1	71.7±1.5	56.9±0.2
Qwen3-8B	None	92.1	79.0	34.9	68.1	41.3	44.9	13.8	57.2	53.9
	IDPO	93.0±0.3	83.4±2.1	38.8±0.6	71.0±0.8	44.8±1.3	46.1±0.3	23.9±1.3	66.4±1.2	58.4±0.4
	+SAI-DPO	92.9±0.3	83.9±0.5	37.6±0.5	72.7±0.9	46.0±1.9	46.1±0.3	24.9±1.6	69.2±0.8	59.2±0.1

Table 1: Main results on multiple test sets for multiple models. The results show that ours can deliver superior results compared to the existing method over multiple model series, notably on competition-level benchmarks.

Method	Type	Size	Avg.
LIMO (Ye et al., 2025)	SFT	0.8K	46.8
S1 (Muennighoff et al., 2025)	SFT	1K	45.9
IRFT (Yuan et al., 2023)	SFT	67K	47.1
PPO (Schulman et al., 2017)	Online RL	400K	54.6
IDPO (Zhang et al., 2025)	Offline RL	67K	52.0
Ours	Offline RL	48K	53.7

Table 2: Compare the training sample utilization efficiency. Qwen2.5-Math-7B serves as the base model.

ing that dynamic data alignment is a fundamental optimizer for reasoning.

Crucially, the performance gains are highly correlated with task difficulty. While improvements on simpler datasets (e.g., GSM8K) are marginal due to performance saturation, SAI-DPO unlocks substantial gains on competition-level benchmarks. For instance, on the challenging AIME24, our method boosts Qwen2.5-7B-Base from 19.6 (IDPO) to 23.7, and Llama-3.1-8B from 1.4 to 4.0. This trend validates that our error-driven curriculum effectively directs the model’s focus toward the frontier of capability, enabling it to master complex reasoning patterns that static baselines fail to capture.

Training Sample Utilization Efficiency To validate data efficiency, we benchmark SAI-DPO against three paradigms: Data Distillation (LIMO, S1), Iterative SFT (IRFT), and Online RL (PPO²). As shown in Table 2, while distillation methods achieve decent baselines with minimal data (1K samples), they suffer from a performance ceiling

²<https://huggingface.co/RLHFlow/Qwen2.5-7B-PP0-Zero>



Figure 3: Compare the results of the data scale. Our method achieved better results with fewer data.

due to the lack of exploration. Conversely, PPO achieves high performance (54.6) but is notoriously data-inefficient, requiring 400K on-policy samples.

Accelerate Convergence As shown in Figure 3, we show the results of the score at the data scale. Since the amount of training data obtained in each iteration is different, their horizontal coordinates are not the same. As the amount of training data gradually increases, the overall scores for both methods rise. However, the results of our method increased more rapidly and reached a peak earlier. This indicates that even with less training data but with difficulty levels matching the model’s capabilities, we achieved better results in fewer iterations, highlighting the importance of appropriately difficult samples in training. These results demonstrate that our method has a higher training efficiency.

4.3 Ablation Study

Dynamic Data Acquisition We dismantle the contribution of each component in Table 3. The full SAI-DPO framework achieves the highest av-

Component	Variant	Avg.	δ
Full Model	–	53.7	–
Knowledge Semantic Alignment	w/o Similarity	51.4	–2.3
	w/o Difficulty	52.5	–1.2
Self-Aware Difficulty	w/o Pass Rate	52.1	–1.6
	w/o Step	52.7	–1.0
	w/o Length	53.1	–0.6

Table 3: Ablation study on the defined measurer.

Task	Sampling Strategies			
	Hard	Curriculum	Clustering	Ours
GSM	82.8	84.4	88.5	89.1
MATH	74.0	76.5	71.8	74.0
Minerva	19.1	18.8	31.2	31.7
Gaokao	59.5	60.5	62.3	62.3
Olympiad	37.5	36.6	39.0	38.9
College	42.2	42.9	45.8	46.3
Aime24	31.7	24.6	23.3	23.7
Amc23	53.8	56.6	60.3	63.3
Avg.	50.1	50.1	52.5	53.7

Table 4: Dynamic sampling strategies comparison.

erage score of 53.7, confirming the synergy of its design. Impact of Semantic Alignment: Removing the “Knowledge Semantic Alignment” module causes the most significant performance degradation ($\delta = -2.3$). This suggests that targeted remedial training on specific weakness domains is the primary driver of our gains. Hierarchy of Difficulty Features: Within the difficulty modeling, Pass Rate emerges as the dominant factor ($\delta = -1.6$). While fine-grained metrics like Step Count ($\delta = -1.0$) and Length ($\delta = -0.6$) contribute marginally, they act as essential tie-breakers for constructing a smooth curriculum. This validates our hierarchical ranking strategy: statistical uncertainty (Pass Rate) defines the coarse difficulty, while cognitive load (Steps/Length) refines it.

Sampling Strategies We further compared the following three sampling configurations on Qwen2.5-Math-7B-Base: 1) Hard-mining without tagging (persistent training on the same instances with the most difficult) named “Hard” (Ye et al., 2025), 2) Curriculum learning (progressing sampling from simple to complex) named “Curriculum” (Chen et al., 2025), 3) Clustering-only sampling named “Clustering” alongside our approach. We only change the sampling strategies. The total amount of data remains the same. Table 4 demonstrates that none of these alternatives outperformed our sampling method, which combines difficulty and similarity.

Task	Filtering Strategies			
	Easy	Hard	Ours	All
Train Samples	34K	34K	48K	67K
GSM	78.4	86.7	89.1	87.3
MATH	68.4	72.2	74.0	73.0
Minerva	21.0	29.4	31.7	29.8
Gaokao	54.5	60.8	62.3	60.3
Olympiad	25.5	37.0	38.9	37.6
College	30.4	44.9	46.3	44.7
Aime24	13.7	27.9	23.7	34.3
Amc23	54.4	58.1	63.3	63.1
Avg.	43.3	52.1	53.7	53.8

Table 5: Compare the results of difficulty filtering. “Train Samples” refers to the total number of samples.

Difficulty Filtering Table 5 demonstrates the training results of the data that have been processed using different filtering strategies and then used for training. Hard refers to the top 50% of the most difficult data after directly ranking the acquired data. Our refers to the top 70% of the most difficult data after removing the results that are all correct or all wrong and then ranking the remaining data. All refers to all the data after removing the results that are all correct or all wrong. It shows that training with overly difficult data or an excessive amount of easy data can lead to a decline. A proper mix of data difficulty (including both hard and simple examples) enhances final performance. The comparison between our and all demonstrates that using 70% of the data achieves performance comparable to that of the original data. The simplest portion of the data contributes little to performance.

5 Analysis

Application to Other Post-training Method Rejection Sampling Fine-tuning (RFT) is a popular and simple baseline for performing preference fine-tuning, which is performed by many instruction-tuned models (Dubey et al., 2024; Bai et al., 2023). Since this step typically only enhances the base model’s capabilities and not those after instruction fine-tuning, we did further comparative experiments on the Qwen-Math-Base model. From Table 6, we can see that combining our data strategy can further improve the effect of RFT with less data. This further demonstrates the generalizability.

The Misalignment of External Difficulty A critical finding of this work is the orthogonality between external difficulty and model-intrinsic difficulty. In Table 7, we compare our method

Task	Base	RFT	RFT+Ours	δ
GSM	66.7	85.7	87.9	+2.2
MATH	64.0	68.4	71.2	+2.8
Minerva	12.1	26.1	32.4	+6.3
Gaokao	56.1	59.0	59.7	+0.7
Olympiad	28.3	34.1	35.6	+1.5
College	41.3	41.5	43.2	+1.7
Aime24	13.9	13.8	18.4	+4.6
Amc23	41.1	48.1	52.2	+4.1
Avg.	40.4	47.1	50.1	+3.0

Table 6: Orthogonality among post-training algorithms.

Task	Original	External	P@K-S	Internal
GSM	88.4	84.4	87.9	89.1
MATH	72.2	76.4	70.4	74.0
Minerva	30.4	18.8	29.4	31.7
Gaokao	61.5	60.5	60.0	62.3
Olympiad	38.0	36.6	38.4	38.9
College	45.9	42.9	45.5	46.3
Aime24	19.4	25.6	17.9	23.7
Amc23	60.4	53.8	63.4	63.3
Avg.	52.0	49.9	51.6	53.7

Table 7: Compare the results of difficulty measurement criteria. External refers to acquiring data from easy to difficult using external difficulty metrics (scoring model) with the human definition. P@K-S is a comparative metric proposed by Zeng et al. (2025b). Internal denotes combining all metrics we defined.

against “External” sampling (using a teacher model’s score, DeepSeek-R1-Distill-Qwen-14B)³ and P@K-S (Zeng et al., 2025b; Chen et al., 2025). Surprisingly, using external difficulty metrics often leads to performance regression (49.9 vs. 52.0 baseline). This implies that a problem deemed hard by a human or a teacher model might be trivial (or hallucinatory) for the student model due to different knowledge boundaries. Only Internal (Self-Aware) metrics consistently drive positive transfer, proving that effective curriculum learning must be personalized to the learner’s specific latent state.

Numbers of Clusters Table 8 investigates how the number of knowledge clusters (N) affects performance. The model achieves the best results with $N = 150$ (Avg. 53.7), while deviating from this number leads to degradation: Too Few Clusters ($N = 50$): Performance drops to 52.1. With overly broad categories, distinct knowledge points are merged, preventing the system from accurately targeting specific weaknesses. Too Many Clusters

³<https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B>

Task	Number of Categories			
	50	150	200	250
GSM	87.1	89.1	88.6	88.2
MATH	70.4	74.0	72.8	70.0
Minerva	27.6	31.7	26.5	30.9
Gaokao	62.6	62.3	61.6	60.5
Olympiad	39.1	38.9	39.1	37.9
College	43.9	46.3	45.8	46.0
Aime24	26.7	23.7	26.7	27.1
Amc23	59.4	63.3	65.0	58.1
Avg.	52.1	53.7	53.3	52.3

Table 8: Compare the impacts of the number of clusters.

Task	Expert Model Scale		
	7B	14B	32B
GSM	88.6	89.1	88.2
MATH	73.0	74.0	73.4
Minerva	31.2	31.7	31.6
Gaokao	62.1	62.3	61.2
Olympiad	39.1	38.9	37.9
College	45.9	46.3	45.6
Aime24	21.2	23.7	30.0
Amc23	62.8	63.3	67.2
Avg.	52.4	53.7	54.4

Table 9: Compare the impacts of the Expert Model Scale. Superior model capabilities yield better results.

($N = 200, 250$): Increasing granularity beyond the optimal point hurts performance. The score decreases to 53.3 at $N = 200$ and drops further at $N = 250$. Excessive fragmentation results in sparse data per category, causing unstable difficulty estimation and reducing the effectiveness of the sampling strategy

Dependence on Knowledge Tags Annotation Model We further investigate the sensitivity of SAI-DPO to the quality of semantic tags by scaling the expert annotator (DeepSeek-R1-Distill-Qwen from 7B to 32B) (Table 9). It demonstrates a robust positive correlation between annotator capability and downstream performance, with the average score rising from 52.4 to 54.4. Notably, this benefit is disproportionately concentrated on complex reasoning benchmarks; for instance, performance on AIME24 surges from 21.2 to 30.0. This suggests that highly capable annotators are essential for accurately disentangling deep semantic structures in challenging problems, thereby ensuring the curriculum precisely targets the model’s cognitive blind spots. We employ the 14B expert in other experiments with minimal additional compute.

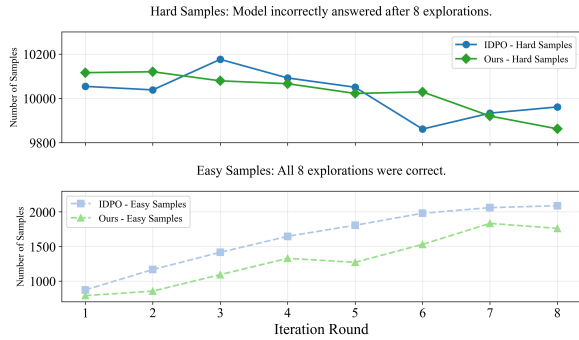


Figure 4: Comparison of the ability to solve sample variability. Both our method and IDPO reduce unsolvable hard samples and enhance simple samples.

Benefits of Hard Samples How does the curriculum evolve? Figure 4 visualizes the dynamic composition of the training set over iterations. Initially, the sampler identifies a large volume of Hard samples (high uncertainty). As training progresses, the model digests these instances, converting them from Hard (low NoP) to Easy (high NoP), causing the volume of remaining Hard samples to decrease. Crucially, SAI-DPO maintains a higher solving rate for hard problems compared to IDPO. This demonstrates our method’s superior ability to learn effectively from sampled data.

6 Conclusion

This work challenges the prevailing paradigm of static data selection in mathematical reasoning. By introducing SAI-DPO, we demonstrate that the definition of high-quality data is inherently dynamic and model-dependent. Our framework, which couples semantic clustering with self-aware difficulty metrics, enables models to autonomously navigate their learning trajectory. The significant gains observed on AIME24 and AMC23 suggest that future research should pivot from simply scaling dataset size to optimizing the alignment between data complexity and real-time model capability.

Limitations

There are several limitations to our work. First, due to training resource constraints, our algorithm primarily focuses on offline RL methods such as iterative DPO and does not conduct sampling-related experiments on mainstream online approaches like PPO. Another limitation is that because of the capacity ceiling of offline algorithms, our final performance does not surpass mainstream online RL algorithms like PPO. Finally, we have only conducted

experiments on test sets of varying difficulty levels within the field of mathematical reasoning, and other domains such as code and general domains remain unexplored.

Ethics Statement

Our work follows the ACL Ethics Policy. Our findings are based on publicly available datasets for reproducibility purposes. All procedures performed in this study are in accordance with the ethical standards. In addition, it is hard to control the generation of LLMs. We should be aware of the potential problems caused by hallucinations.

Acknowledgments

This work was supported in part by Guangdong S&T Program (Grant No. 2024B0101050003), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), and Shenzhen Science and Technology Program (Grant Nos. ZDSYS20230626091203008, KJZD20231023094700001, KQTD20240729102154066). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. [Qwen technical report](#). *arXiv preprint arXiv:2309.16609*.
- Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamaloo. 2025. [Self-evolving curriculum for llm reasoning](#). *arXiv preprint arXiv:2505.14970*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Jun Rao, and Min Zhang. 2026. [REA-RL: Reflection-aware online reinforcement learning for efficient reasoning](#). In *The Fourteenth International Conference on Learning Representations*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. [The llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.

- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. [Model alignment as prospect theoretic optimization](#). In *Proceedings of the 41st International Conference on Machine Learning*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024. [Direct language model alignment from online ai feedback](#). *arXiv preprint arXiv:2402.04792*.
- Guangzeng Han, Weisi Liu, and Xiaolei Huang. 2025. [Attributes as textual genes: Leveraging LLMs as genetic algorithm simulators for conditional synthetic data generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19367–19389, Suzhou, China. Association for Computational Linguistics.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. 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](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. 2025. [Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning](#). *arXiv preprint arXiv:2507.01006*.
- Jian Hu, Jason Klein Liu, and Wei Shen. 2025. [Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models](#). *Preprint*, arXiv:2501.03262.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helvar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- Yuxuan Jiang, Dawei Li, and Frank Ferraro. 2025. [Drp: Distilled reasoning pruning with skill-aware step decomposition for efficient large reasoning models](#). *Preprint*, arXiv:2505.13975.
- Xiaopeng Ke, Hexuan Deng, Xuebo Liu, Jun Rao, Zhenxi Song, Jun Yu, and Min Zhang. 2025. [Aquila: Weaving logic and self-inspection into low-cost, high-relevance data synthesis for specialist llms](#). In *EMNLP*.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- 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.
- Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. 2024a. [Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions](#). *Hugging Face repository*, 13:9.
- Xinjin Li, Yu Ma, Yangchen Huang, Xingqi Wang, Yuzhen Lin, and Chenxi Zhang. 2024b. [Synergized data efficiency and compression \(sec\) optimization for large language models](#). In *2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 586–591.
- Minpeng Liao, Chengxi Li, Wei Luo, Wu Jing, and Kai Fan. 2024. [MARIO: MATH reasoning with code interpreter output - a reproducible pipeline](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 905–924, Bangkok, Thailand. Association for Computational Linguistics.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. [Deepseek-v3 technical report](#). *arXiv preprint arXiv:2412.19437*.
- Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024b. [SelectIT: Selective instruction tuning for LLMs via uncertainty-aware self-reflection](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#). *Preprint*, arXiv:2501.19393.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. [Training language models to follow instructions with human feedback](#). *NeurIPS*, 35:27730–27744.

- Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. [Iterative reasoning preference optimization](#). *Preprint*, arXiv:2404.19733.
- 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](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Jun Rao, Yunjie Liao, Xuebo Liu, Zepeng Lin, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025a. [Seapo: Strategic error amplification for robust preference optimization of large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics.
- Jun Rao, Zepeng Lin, Xuebo Liu, Xiaopeng Ke, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025b. [APT: Improving specialist LLM performance with weakness case acquisition and iterative preference training](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20958–20980, Vienna, Austria. Association for Computational Linguistics.
- Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. [CommonIT: Commonality-aware instruction tuning for large language models via data partitions](#). In *EMNLP*, pages 10064–10083, Miami, Florida, USA. Association for Computational Linguistics.
- Jun Rao, Xuebo Liu, Haotian Yan, Junjie Shen, Haosi Mo, Yanghaopeng Dong, Zihao Yan, Ziyi Wang, Zepeng Lin, Xiaojun Meng, Zixiong Yu, Liqun Deng, Jiansheng Wei, Yunhe Wang, and Min Zhang. 2025c. [A data-centric perspective on the lifecycle of large language models](#). *TechRxiv*, 2025(1220).
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Joykirat Singh, Tanmoy Chakraborty, and Akshay Nambi. 2025. [Self-evolved preference optimization for enhancing mathematical reasoning in small language models](#). *Preprint*, arXiv:2503.04813.
- Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. [Mathscale: Scaling instruction tuning for mathematical reasoning](#). In *ICML*, pages 47885–47900.
- 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.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. [Kimi k1. 5: Scaling reinforcement learning with llms](#). *arXiv preprint arXiv:2501.12599*.
- Qwen Team. 2024. [Qwq: Reflect deeply on the boundaries of the unknown](#).
- Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, and Dongbin Zhao. 2025. [Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation](#). *Preprint*, arXiv:2503.12854.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. [Qwen3 technical report](#). *arXiv 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. 2024. [Qwen2.5-math technical report: Toward mathematical expert model via self-improvement](#). *Preprint*, arXiv:2409.12122.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. [Limo: Less is more for reasoning](#). *Preprint*, arXiv:2502.03387.
- Yichun Yin, Wenyong Huang, Kaikai Song, Yehui Tang, Xueyu Wu, Wei Guo, Peng Guo, Yaoyuan Wang, Xiaojun Meng, Yasheng Wang, Dong Li, Can Chen, Dandan Tu, Yin Li, Fisher Yu, Ruiming Tang, Yunhe Wang, Baojun Wang, Bin Wang, Bo Wang, Boxiao Liu, Changzheng Zhang, Duyu Tang, Fei Mi, Hui Jin, Jiansheng Wei, Jiarui Qin, Jinpeng Li, Jun Zhao, Liqun Deng, Lin Li, Minghui Xu, Naifu Zhang, Nianzu Zheng, Qiang Li, Rongju Ruan, Shengjun Cheng, Tianyu Guo, Wei He, Wei Li, Weiwen Liu, Wulong Liu, Xinyi Dai, Yonghan Dong, Yu Pan, Yue Li, Yufei Wang, Yujun Li, Yunsheng Ni, Zhe Liu, Zhenhe Zhang, and Zhicheng Liu. 2025. [Pangu ultra: Pushing the limits of dense large language models on ascend npus](#). *Preprint*, arXiv:2504.07866.
- Zixiong Yu, Jun Rao, Guhan Chen, Songtao Tian, Bohan Li, Jiansheng Wei, Min Zhang, and Xiaojun Meng. 2026. [Mathagent: Adversarial evolution of constraint graphs for mathematical reasoning data synthesis](#). *Preprint*, arXiv:2604.11188.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. [Scaling relationship on learning mathematical reasoning with large language models](#). *arXiv preprint arXiv:2308.01825*.

- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025a. [Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild](#). In *COLM*.
- Weihao Zeng, Yuzhen Huang, Lulu Zhao, Yijun Wang, Zifei Shan, and Junxian He. 2025b. [B-star: Monitoring and balancing exploration and exploitation in self-taught reasoners](#). In *The Thirteenth International Conference on Learning Representations*.
- Yiming Zeng, Wanhao Yu, Zexin Li, Tao Ren, Yu Ma, Jinghan Cao, Xiyan Chen, and Tingting Yu. 2025c. [Bridging the editing gap in LLMs: FineEdit for precise and targeted text modifications](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2193–2206, Suzhou, China. Association for Computational Linguistics.
- Hanning Zhang, Jiarui Yao, Chenlu Ye, Wei Xiong, and Tong Zhang. 2025. [Online-dpo-r1: Unlocking effective reasoning without the ppo overhead](#). *Notion Blog*.
- Hengyuan Zhang, Shiping Yang, Xiao Liang, Chenming Shang, Yuxuan Jiang, Chaofan Tao, Jing Xiong, Hayden Kwok-Hay So, Ruobing Xie, Angel X. Chang, and Ngai Wong. 2026. [Find your optimal teacher: Personalized data synthesis via router-guided multi-teacher distillation](#). *Preprint*, arXiv:2510.10925.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. [LIMA: less is more for alignment](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. 2026. [Look inward to explore outward: Learning temperature policy from llm internal states via hierarchical rl](#). *arXiv preprint arXiv:*

A Appendix

A.1 Training Times Comparison

The following table summarizes the total wall-clock training time for 8 iterations, comparing our SAI-DPO framework with the IDPO baseline. While SAI-DPO introduces a dynamic sampling phase, its superior data efficiency leads to a significant reduction in total training time.

Table 10: Training Efficiency Comparison of Different Methods

Model Scale	Method	Total Data Volume	Avg. Time per Iteration
7B Model	IDPO (Baseline)	67K	7.0 hours
	SAI-DPO (Ours)	48K	~5.3 hours
8B Model	IDPO (Baseline)	67K	8.0 hours
	SAI-DPO (Ours)	48K	~6.0 hours

A.2 Examples of Tags

We present several examples for different types of math problems. We use a uniform prompt template to prompt the model to output the corresponding tag. The specific prompt template is as follows: “<|begin of sentence|><|User|>What knowledge points need to be involved in solving the following questions. Answer should be output in the following format, no need to output the answer, reply in English, ###Knowledge Points:{ }\nPlease output the results directly, reducing the thought process.{input}<|Assistant|>###Knowledge Points:” The {input} is a placeholder indicating the input question. As shown in the figure (Tag Case), the second and third examples both involve trigonometry, indicating repetitive knowledge points. By leveraging this tagging, we can better identify data with similar knowledge points, thereby enabling self-learning by locating example problems for knowledge points where the model currently has weaknesses.

A.3 Examples of Problem-Solving Steps

We present the output results of several models, which can automatically continue to generate replies in this format by adding the field “Step: 1”. We present response examples from the Qwen-7B-Distill model. As shown in the examples, the model can output logical problem-solving steps and include reflections (e.g., “wait”), where correct reflection processes solve problems using fewer logical steps. Additionally, the output results can be used to calculate total step counts, response length, and correctness, thereby determining the difficulty of the current data for the model.

Responses of Explorations

Question:

A girl has three different types of coins in her wallet: 3 nickels, 3 dimes, and 2 quarters. She also adds 2 half-dollars. She randomly picks two coins, calculates the total value, and puts them back. She repeats this process multiple times. What is the maximum number of different sums she can record?

gt:10

False Response 1 (No answer output):

Step 1: Identify all possible coin combinations. First, list all the coins available:- 3 nickels (each worth 0.05)- 3 dimes (each worth 0.10)- 2 quarters (each worth 0.25)- 2 half-dollars (each worth 0.50)...

Step 2: Determine all possible sums for picking two coins. ...

Wait, perhaps not directly.

Wait, she picks two coins, calculates their sum, puts them back, and repeats.

So, each pick results in a sum, which is recorded.

She does this multiple times, perhaps until she gets all possible sums.

Wait, but the problem seems to suggest that the sums are recorded based on how many times they occur, so perhaps it's about the frequency.

But, in any case, we need to find out how

False Response 2 (Wrong answer):

Step 1: Identify the possible values of each type of coin ...

Step 2: Determine all possible combinations of two coins being picked. - Nickels: There are 3 nickels, but since the girl is picking two coins at random, she could pick two nickels, a nickel and a dime, a nickel and a quarter, a nickel and a half-dollar, a dime and a quarter, a dime and a half-dollar, and a quarter and a half-dollar.

Step 3: Calculate the unique sums for each combination ...

Step 4: Ensure that all sums are unique. Looking at the sums: 0.10, 0.15, 0.30, 0.55, 0.35, 0.60, 0.75. These sums are all distinct.

Step 5: Determine the number of different sums. From the above, we have: - 0.10 from two nickels - 0.15 from nickel and dime - 0.30 from nickel and quarter - 0.55 from nickel and half-dollar - 0.35 from dime and quarter - 0.60 from dime and half-dollar - 0.75 from quarter and half-dollar

So, there are 7 different sums.

Therefore, the maximum number of different sums she can record is ...

****Final Answer**:**

True Response 1:

Step 1: Determine the total number of coins and their values ...

So, total coins: $3 + 3 + 2 + 2 = 10$ coins.

Step 2: Calculate all possible sums of two coins.

We need to consider all combinations of picking two coins and summing their values, then find out how many unique sums are possible.

Step 3: List all possible pairs and their sums ...

So, it seems like these are all the possible sums.

Therefore, the number of different sums is 10.

Wait, but is there any overlap or additional sums I might be missing?

Let me think differently.

...

Hence, the maximum number of different sums is 10.

Therefore, the final answer is:

I hope it's correct.

**Final Answer** I think that's the answer.

Tag Case 1

Question: Evaluate the expression $5^{\log_5(9+4)}$.

Tags: Logarithmic and Exponential Expressions and Equations

Tag Case 2

Question: Determine the domain of the function $f(x) = \tan(\arcsin(x^2))$.

Tags: Trigonometric functions, inverse trigonometric functions, domain and range

Tag Case 3

Question:

Given the function $f(x) = A \sin(\omega x + \phi)$, where $(A > 0, \omega > 0, |\phi| < \frac{\pi}{2})$, its graph intersects the y -axis at $(0, \frac{3}{2})$, and its first highest and lowest points on the right side of the y -axis are $(x_0, 3)$ and $(x_0 + 2\pi, -3)$ respectively.

1. Find the analytical expression of the function $y = f(x)$.
2. How can the graph of this function be obtained through translation and scaling transformations from the graph of $y = \sin x (x \in \mathbb{R})$?
3. Find the intervals where this function is monotonically increasing and its center of symmetry.

Tags: Trigonometric functions, function transformation, monotonicity, symmetry, maximum and minimum points.

Tag Case 4

Question: Find the largest prime divisor of the number 102111011_6 .

Tags: Number Bases, Prime Factorization.