Adaptive Curriculum Strategies: Stabilizing Reinforcement Learning for Large Language Models

Anonymous ACL submission

Abstract

Curriculum learning has shown promise for enhancing Large Language Models (LLMs) through progressive difficulty management, yet existing approaches suffer from severe instability when applied to reinforcement learning paradigms. A core unresolved challenge is how to handle samples that exceed the model's current capability: discarding them wastes valuable training signal, while forcing models to learn from them destabilizes training. Inspired by how human learners tackle difficult problems—by consulting partial solutions before completing the rest independently—we propose "Guided Prompting," a step-level sample transformation technique that prepends solution steps as hints to transform challenging samples into accessible learning opportunities, enabling models to safely learn from difficult content without policy collapse. To precisely identify which samples require such transformation, we further introduce model-specific difficulty calibration that assesses sample accessibility based on each model's actual performance rather than fixed external labels. Together, these two components form Adaptive Curriculum Strategies (ACS), a framework that promotes stable and effective curriculum-based RL training. Experiments demonstrate that ACS prevents performance collapse across five mathematical reasoning benchmarks while achieving substantial performance improvements.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in complex reasoning tasks, with reinforcement learning emerging as an effective paradigm for mathematical reasoning due to its clear reward signals and iterative optimization capabilities (Guo et al., 2024; Shao et al., 2024; Yang et al., 2025; Kavukcuoglu, 2025; OpenAI, 2024). Recent work has introduced curriculum learning strategies into RL frameworks, progressively exposing models to problems of increasing difficulty and showing promising improvements in convergence speed and performance (Wen et al., 2025b; Huang et al., 2025; Shi et al., 2025; Team et al., 2025). However, a fundamental question remains unresolved: *what should a curriculum do when a model encounters samples beyond its current capability?*

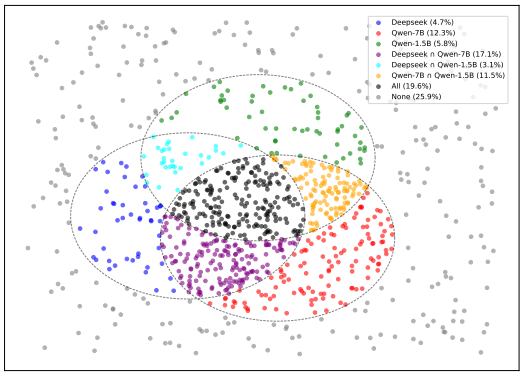

Figure 1: Solution correctness patterns for three mathematical reasoning models. Approximately 55% of questions easy for one model prove difficult for another.

The Core Challenge: Handling Difficult Samples in Curriculum RL. Consider how human students approach a difficult math problem: they do not simply skip it, nor do they struggle blindly—instead, they consult partial solutions or worked examples to bridge the gap, then complete the reasoning independently. Existing curriculum-based RL methods lack this nuanced handling. Current approaches either (1) *discard* challenging samples to avoid instability, wasting valuable training signal from hard problems, or (2) *retain* them unchanged, directly exposing models to samples far beyond their capabilities and triggering catastrophic performance collapse. As demonstrated in Figure 3, the latter leads to accuracy drops of approximately 30% during difficulty transitions, while the former sacrifices the learning value of challenging content.

Neither strategy resolves the fundamental tension between stability and sample utilization.

This instability is further compounded by a mismatch in difficulty perception across models. As illustrated in Figure 1, approximately 55% of questions that are easily solved by one model prove challenging for another. Existing methods rely on fixed external difficulty hierarchies (Yu et al., 2025; Wen et al., 2025a) that ignore these per-model differences, causing the same curriculum to destabilize one model while working well for another. This reveals a second limitation: without knowing *which* samples are truly challenging for a specific model, even a well-designed transformation strategy cannot be applied precisely.

Our Solution: Adaptive Curriculum Strategies. We propose Adaptive Curriculum Strategies (ACS), a framework that addresses both challenges through two tightly coupled innovations. First, inspired by the human learning strategy of consulting partial solutions, we introduce **Guided Prompting**—a step-level sample transformation technique that prepends solution steps as hints to challenging samples, transforming them into accessible learning opportunities. This allows models to learn from difficult content without triggering policy collapse, striking the balance between stability and full sample utilization that discard and retain strategies both fail to achieve. Second, to precisely identify which samples require such transformation for each specific model, we introduce **model-specific difficulty calibration** that assesses sample accessibility by directly measuring model performance, rather than relying on fixed external labels. Together, these components ensure that the curriculum always matches the model's evolving capabilities, enabling stable and effective progressive RL training.

The contributions of this work include:

- We identify the core challenge in curriculum-based RL as the handling of difficult samples: discarding wastes training signal while retaining causes catastrophic collapse, and we show that neither existing strategy resolves this tension.
- We propose **Guided Prompting**, a step-level sample transformation technique that resolves the discard-vs-retain dilemma by converting challenging samples into accessible learning opportunities via partial solution steps, enabling stable learning from difficult content without policy collapse.
- We demonstrate through comprehensive experiments on five mathematical reasoning benchmarks that ACS eliminates training instability while achieving consistent performance improvements across model scales and RL algorithms.

2 Related Work

Curriculum Learning and Adaptive Training Strategies. The challenge of effectively managing training sample difficulty has gained increasing attention in large language model optimization. Bengio et al. (2009) established foundational concepts for progressive difficulty management in machine learning, demonstrating that models learn more effectively when training examples are presented with appropriate difficulty sequencing. Recent work has explored various approaches to difficulty-aware training in language models. Xie et al. (2025) implements difficulty progression by adjusting task complexity based on logical reasoning requirements, while Wen et al. (2025b) identifies challenging samples based on model prediction failures and defers them to later training stages. Team et al. (2025) focuses on filtering strategies that remove problematic samples early in training, concentrating computational resources on high-quality examples. Huang et al. (2025) applies difficulty-aware training to retrieval-augmented generation, ordering tasks by the complexity of retrieved information. Shi et al. (2025) proposes dynamic sample selection based on predefined difficulty scores that align with model capabilities during training. Tong et al. (2024); Xue et al. (2025) implement empirical approaches through multi-round sampling, defining difficulty through response accuracy patterns and allocating training emphasis accordingly. Ma et al. (2024) extends this concept by implementing inverse accuracy weighting, where samples with lower success rates receive proportionally greater training attention. However, these approaches share two fundamental limitations: they rely on fixed difficulty hierarchies that fail to account for individual model capabilities, and they handle challenging samples crudely—either discarding them entirely or retaining them unchanged—without addressing the core tension between training stability and sample utilization.

Reinforcement Learning for Mathematical Reasoning. Reinforcement learning has emerged as a particularly effective paradigm for mathemat-

ical reasoning tasks, where reward signals can be clearly defined through solution correctness. Luo et al. (2023); Luong et al. (2024); Yue et al. (2025) demonstrate effectiveness of Proximal Policy Optimization (PPO) for mathematical reasoning enhancement. Shao et al. (2024); DeepSeek-AI et al. (2025); Yu et al. (2025) advance the field through Group Relative Policy Optimization (GRPO), showing substantial improvements in reasoning performance. However, when integrated with curriculum learning strategies, these methods lack stability-preserving mechanisms and often suffer from catastrophic performance collapse during difficulty transitions. The effectiveness of reinforcement learning in mathematical domains makes it an ideal testbed for adaptive difficulty management strategies, as reward signals provide clear feedback on model capability progression.

3 Method

The fundamental challenge in curriculum-based RL training is not simply ordering samples by difficulty—it is deciding what to do when a model encounters samples beyond its current capability. Existing approaches either discard such samples, forfeiting their training value, or retain them unchanged, triggering catastrophic policy collapse. We propose Adaptive Curriculum Strategies (ACS), a framework that resolves this tension through two tightly coupled components: **Guided Prompting** for sample transformation, and **model-specific difficulty calibration** to identify which samples require transformation. The full algorithm is detailed in Algorithm 1.

3.1 Guided Prompting for Sample Transformation

As shown in Figure 3, retaining difficult samples unchanged causes catastrophic collapse (e.g., 30% accuracy drop for 1.5B model in Stage III), while discarding them forfeits valuable training signal. We introduce **Guided Prompting**, a step-level sample transformation technique inspired by the educational scaffolding principle, that escapes this dilemma by prepending partial solution steps as hints to transform challenging samples into accessible learning opportunities.

For a challenging problem Q_i with reference solution $S_i = \{s_{i1}, s_{i2}, ..., s_{ik}\}$, we extract a strategic guidance prefix $P_i = \{s_{i1}, s_{i2}, ..., s_{ip}\}$ where $p < k$. We incrementally provide solution steps until either the hint ratio $\frac{|p|}{|k|}$ reaches a predefined maximum α, or the model's performance improves to meet an accuracy threshold τ. This transformation creates a stability-preserving training example:

$$y_i \sim \pi_\theta(Y|[Q_i; P_i]) \tag{1}$$

By providing just enough scaffolding to bring the sample within the model's learning capacity, Guided Prompting ensures stable reward signals and policy gradients while preserving the educational value of difficult content. As shown in Figure 3, this enables consistent performance improvement throughout all curriculum stages, eliminating the catastrophic collapses that plague the retain strategy and recovering the sample utilization lost by the discard strategy.

3.2 Model-Specific Difficulty Calibration

Guided Prompting requires knowing *which* samples are genuinely challenging for a given model—the prerequisite for deciding where to apply the transformation. This identification cannot rely on fixed external difficulty labels, because difficulty perception varies dramatically across model architectures (as shown in Figure 1): a sample that is easy for one model may be hard for another, making universal difficulty hierarchies unreliable.

We address this through model-specific difficulty calibration that directly measures each model's performance on training samples. For each training sample i, we generate n responses and evaluate their correctness against the reference answer:

$$ACC_i = \frac{\sum_{j=1}^{n} \mathbb{I}\{A_{ij} = A_i^*\}}{n} \tag{2}$$

where A_{ij} represents the j-th generated answer for sample i, A_i^* denotes the reference answer, and $\mathbb{I}\{\}$ is the indicator function. ACC_i provides a direct, model-specific measure of whether the sample is within the model's current learning capacity. Samples with low ACC_i are identified as challenging and routed to Guided Prompting; samples with high ACC_i are treated as accessible and used directly for training.

Computational Efficiency. We employ the VLLM framework for efficient inference acceleration. The additional computational overhead for difficulty calibration represents less than 5% of total GPU hours, making it negligible relative to the training process.

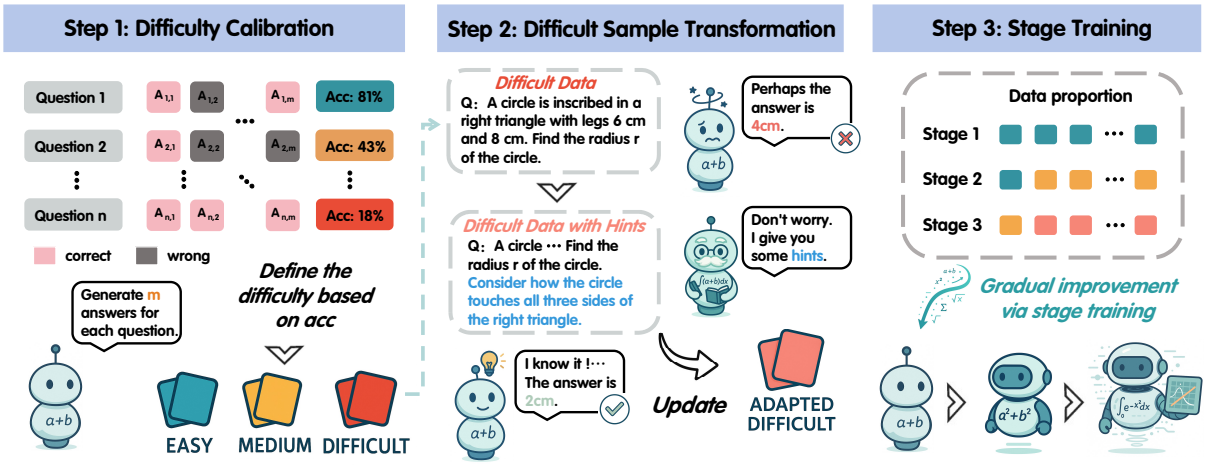

Figure 2: **Adaptive Curriculum Strategies Pipeline for Stable RL Training. Step 1:** Model-specific difficulty calibration adapts to individual model capabilities, ensuring curriculum construction aligns with each model's evolving learning capacity. **Step 2:** Guided prompting strategically transforms challenging samples to prevent training instability while preserving learning value, maintaining stable training dynamics. **Step 3:** Progressive curriculum training with data mixing strategy enables gradual improvement through staged learning and curriculum review, building capabilities from basic to advanced reasoning tasks.

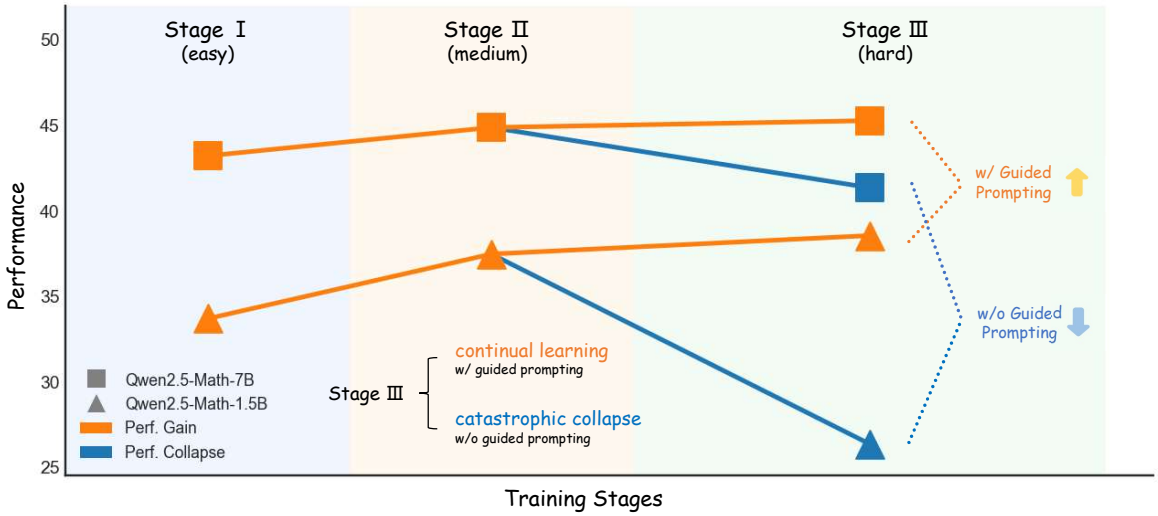

Figure 3: **Training Stability Analysis: ACS vs. Traditional Curriculum Approaches.** Our ACS framework (orange line) shows improved training progression throughout all curriculum stages, while traditional approaches (blue line) suffer significant performance degradation in Stage III. The results demonstrate the benefits of adaptive curriculum strategies for enhancing RL training stability across different model sizes: Qwen2.5-Math-1.5B (triangles) and Qwen2.5-Math-7B (squares).

Calibration Robustness. To validate the robustness of our difficulty calibration approach, we examined how variations in sampling parameters affect data partitioning consistency. Using our default configuration ($n = 16, T = 0.7$) as reference, we measure the overlap ratio between partitions generated with different parameters. As shown in Table 7, our results demonstrate high consistency across different parameter choices, with average overlap ratios exceeding 93% in all configurations. This indicates that our difficulty calibration produces stable and reliable sample partitioning that is robust to reasonable variations in sampling parameters.

Algorithm 1 Adaptive Curriculum Strategies for Stable RL Training

Input: training dataset $D = \{(Q_i, A_i)\}$, pre-trained model π_0, stability threshold τ, adaptation ratio α
Adaptive Curriculum Construction:
for each question $Q_i \in D$ **do**
  Generate n responses $\{A_{i1}, \ldots, A_{in}\}$ using π_θ
  $ACC_i = \frac{\sum_{j=1}^{n} \mathbb{I}\{A_{ij}=A_i^*\}}{n}$
end for
Partition D into $\{D_1, D_2, ..., D_p\}$ by accessibility
Stability-Preserving Transformation:
for each $(Q_i, A_i) \in D_p$ (challenging) **do**
  Decompose S_i into steps $\{s_{i1}, \ldots, s_{ik}\}$
  Provide hints $S_{i,1:l}$ until $ACC \geq \tau$ or ratio $\geq \alpha$
  if stability achieved **then**
    Transform: $Q'_i \leftarrow [Q_i; S_{i,1:l}]$, $A'_i \leftarrow S_{i,l+1:k}$
  else
    Discard sample from training set
  end if
end for
Progressive RL Training:
for each stage $s \in \{1, \ldots, p\}$ **do**
  $\pi_s = \arg\min_{(Q,A)\in D_s} \mathcal{L}_{RL}(\pi_{s-1})$
end for
Output: Stably trained model π_m

3.3 Progressive Reinforcement Learning

Using the difficulty calibration to partition training data into stability-ordered subsets $D = \{D_1, D_2, ..., D_p\}$ and Guided Prompting to transform challenging samples in D_p, we implement staged reinforcement learning that maintains training stability throughout curriculum progression. Each stage trains on the corresponding subset, progressively building from accessible to challenging content.

We employ Group Relative Policy Optimization (GRPO) (DeepSeek-AI et al., 2025), which generates G candidate responses per question and optimizes via relative reward advantages. The reward combines solution correctness and format compliance ($r = r_{accuracy} + r_{format}$), providing stable optimization signals that enable consistent policy improvement. By ensuring all training samples are within the model's learning capacity—either through natural accessibility or Guided Prompting transformation—our staged training eliminates the catastrophic collapses that plague traditional curriculum-based RL approaches. Full GRPO formulation and training hyperparameters are provided in the Appendix.

4 Experiments

4.1 Experimental Setup

We evaluate our ACS framework on mathematical reasoning tasks across two model scales: Qwen2.5-Math-1.5B and Qwen2.5-Math-7B, using GRPO as the RL backbone. Training data consists of 9,255 samples from levels 3–5 of the MATH dataset (Hendrycks et al., 2021), partitioned into three difficulty stages via model-specific difficulty calibration. We evaluate on five benchmarks spanning competition to olympiad difficulty: MATH 500, Minerva Math, OlympiadBench, AMC 23, and AIME 24. Complete experimental details are provided in the Appendix.

4.2 Comparative Methods and Experimental Design

To systematically evaluate our ACS framework, we design comprehensive comparisons across two key dimensions of curriculum learning.

4.2.1 Difficulty Calibration Strategies

We compare our model-specific difficulty calibration against three established approaches:

Light-R1. Following Wen et al. (2025b), this method employs a fixed external model to assess sample difficulty. Difficulty rankings are determined based on the external model's success rates, representing a model-agnostic assessment strategy that does not account for target model capabilities.

Length-based Ranking. This heuristic approach ranks samples based on solution length, assuming longer solutions correspond to more complex problems. Samples are organized in ascending order of reference solution lengths.

Original Dataset Labels. This approach utilizes the predefined difficulty levels (1-5) provided by the MATH dataset, organizing training samples according to expert-defined difficulty hierarchies without considering individual model performance.

4.2.2 Challenging Sample Processing Strategies

For samples exceeding model capabilities, we evaluate two alternative processing approaches:

Table 1: Main experimental results comparing ACS with baseline methods and ablation studies across two model scales. Results show performance on five mathematical reasoning benchmarks using GRPO training. ACS demonstrates consistent improvements through model-specific difficulty calibration and guided prompting for challenging samples.

MODEL	METHOD	MATH 500	MINERVA MATH	OLYMPIAD BENCH	AIME24	AMC23	AVERAGE
QWEN2.5 MATH 1.5B	BASE MODEL	42.8	7.7	25.2	3.3	22.5	20.3
	GRPO	51.8	18.4	21.0	10.0	22.5	24.7
	Difficulty Calibration Methods						
	LIGHT-R1	64.2	31.2	30.5	6.7	40.0	34.5
	LENGTH	67.4	**33.8**	28.7	6.7	32.5	33.8
	ORIGINAL	57.8	23.9	23.7	6.7	27.5	27.9
	Challenging Sample Processing						
	RETAIN	62.2	17.3	26.5	3.3	22.5	26.4
	DISCARD	72.0	30.9	30.1	10.0	40.0	36.6
	ACS (OURS)	**72.6**	31.6	**32.7**	**13.3**	**42.5**	**38.5**
QWEN2.5 MATH 7B	BASE MODEL	63.6	12.5	25.8	10.0	42.5	30.9
	GRPO	74.2	33.5	33.9	10.0	**62.5**	42.8
	Difficulty Calibration Methods						
	LIGHT-R1	73.2	**43.0**	37.8	**13.3**	55.0	44.5
	LENGTH	75.2	40.8	34.7	**13.3**	57.5	44.3
	ORIGINAL	75.6	35.7	35.7	**13.3**	45.0	41.1
	Challenging Sample Processing						
	RETAIN	71.8	41.5	35.9	10.0	47.5	41.3
	DISCARD	**77.4**	37.5	37.3	**13.3**	55.0	44.1
	ACS (OURS)	76.6	38.2	**38.2**	**13.3**	60.0	**45.3**

Table 2: Cross-model evaluation results on DeepSeek-Math-7B-Instruct model, comparing ACS with the best-performing baseline methods from each category

MODEL	METHOD	MATH 500	MINERVA	OLYMPIAD	AIME24	AMC23	AVG
DEEPSEEK MATH 7B	GRPO	39.6	18.0	13.6	**3.3**	15.0	17.9
	LIGHT-R1	40.2	20.6	14.0	0.0	**17.5**	18.5
	DISCARD	40.8	21.4	13.2	**3.3**	15.0	18.7
	ACS	**41.4**	**22.8**	**14.8**	**3.3**	15.0	**19.5**

Retain All Samples. This approach includes all difficult samples without modification, potentially exposing models to training instability from inaccessible examples.

Discard Difficult Samples. This conservative strategy removes samples below a predefined accuracy threshold, avoiding negative impacts but discarding valuable training data.

4.3 Main Results

As shown in Table 1, the results reveal a clear pattern that directly validates the motivation for Guided Prompting. Among challenging sample processing strategies, **Retain** (keeping all difficult samples unchanged) leads to severe instability and underperforms even the plain GRPO baseline in several settings (e.g., 26.4 vs. 24.7 average for 1.5B), confirming that directly exposing models to overly difficult content is harmful. **Discard** (removing difficult samples) avoids the collapse and achieves competitive performance (36.6 average for 1.5B), but leaves performance gains on the table by discarding valuable training signal. **ACS with Guided Prompting** resolves this tension: by transforming difficult samples into accessible learning opportunities, it achieves the best average performance on both model scales (38.5 for 1.5B, 45.3 for 7B), outperforming both Retain and Discard while maintaining stable training throughout all curriculum stages.

Compared to the GRPO baseline, the overall improvements are pronounced at 55.9% (from 24.7 to 38.5) and 5.8% (from 42.8 to 45.3) for Qwen2.5-Math-1.5B and Qwen2.5-Math-7B respectively. The gains are particularly notable on harder benchmarks: ACS achieves 13.3 on AIME24 for both model scales, compared to 3.3 (Retain) and 10.0 (Discard) for the 1.5B model, demonstrating that

Table 3: Generalizability evaluation using PPO algorithm on Qwen2.5-Math-1.5B model

METHOD	MATH 500	MINERVA	OLYMPIAD	AIME24	AMC23	AVG
PPO	65.6	16.8	19.5	6.7	32.5	28.2
PPO W/ ACS	**69.4**	**28.7**	**28.9**	**10.0**	**40.0**	**35.4**

Table 4: Impact of Guided Prompting on model performance. All test sets contain NO hints, providing direct assessment of independent problem-solving ability. "w/o Guided Prompting" corresponds to the "Retain" baseline.

MODEL	STRATEGY	MATH	MINERVA	OLYMPIAD	AIME	AMC	AVG
QWEN2.5	W/O GP	62.2	17.3	26.5	3.3	22.5	26.4
1.5B	W/ GP	**72.6**	**31.6**	**32.7**	**13.3**	**42.5**	**38.5**
QWEN2.5	W/O GP	71.8	41.5	35.9	10.0	47.5	41.3
7B	W/ GP	**76.6**	**38.2**	**38.2**	**13.3**	**60.0**	**45.3**

Guided Prompting is especially effective at enabling learning from the most challenging content.

ACS also consistently outperforms all difficulty calibration baselines (Light-R1, Length-based, Original). Among these, Light-R1 performs best (34.5 for 1.5B, 44.5 for 7B) but still falls short of ACS, confirming that model-specific difficulty calibration provides a more precise foundation for curriculum construction than fixed external labels. The gap is more pronounced for the smaller 1.5B model, suggesting that model-specific adaptation is especially critical when model capabilities are more limited.

4.4 Ablation Studies

4.4.1 Impact of Guided Prompting

A critical component of our ACS framework is the Guided Prompting mechanism, which transforms challenging samples to prevent training instability while preserving learning value. To validate the effectiveness of this design choice, we conducted direct comparison experiments between models trained with and without guided prompting. An important consideration is whether hint-augmented training might weaken the model's independent problem-solving ability. If the model truly developed a "dependency" or weakened capability, its performance on hint-free test sets should decline or stagnate. However, our results demonstrate the opposite.

As shown in Table 4, all test sets contain NO hints, providing a direct assessment of whether the model has internalized the reasoning patterns or merely become dependent on external guidance. The results clearly demonstrate that guided prompting significantly improves model performance across all benchmarks. These results provide strong evidence that guided prompting does not create dependency but rather helps models internalize effective problem-solving strategies. The consistent improvements across diverse mathematical reasoning benchmarks, evaluated without any hints, validate that our stability-preserving sample transformation successfully enables models to learn from challenging content while maintaining stable training dynamics.

4.4.2 Data Mixing Strategy

Drawing inspiration from human learning processes where students periodically review previously mastered knowledge, we investigate whether models undergoing staged curriculum learning require similar reinforcement of previously acquired content. This analysis is particularly crucial for maintaining training stability throughout the curriculum progression. We design two distinct data mixing strategies for comparative analysis within our ACS framework:

Naive Curriculum: Models receive samples corresponding only to the current difficulty level at each training stage, focusing exclusively on new, challenging content without revisiting previously learned material.

Curriculum Review: A strategic data mixing approach that incorporates a small proportion of easier samples from previous stages during later training phases, allowing the model to revisit and reinforce previously acquired capabilities while learning new content.

Experimental results in Table 5 demonstrate that the Curriculum Review strategy consistently

Table 5: Data mixing strategy comparison: Naive Curriculum vs. Curriculum Review across different model scales. Curriculum Review incorporates 10% of samples from the previous difficulty level at each training stage.

MODEL	METHOD	MATH	MINERVA	OLYMPIAD	AIME	AMC	AVG
QWEN2.5 1.5B	NAIVE	69.8	**33.8**	30.8	6.7	22.5	32.7
	REVIEW	**72.6**	31.6	**32.7**	**13.3**	**42.5**	**38.5**
QWEN2.5 7B	NAIVE	75.2	35.7	36.4	**13.3**	52.5	42.6
	REVIEW	**76.6**	**38.2**	**38.2**	**13.3**	**60.0**	**45.2**

outperforms the Naive Curriculum approach and achieves the best performance across both model scales. For the 1.5B model, Curriculum Review achieves a significant 17.7% average performance improvement compared to Naive Curriculum (from 32.7 to 38.5), while the 7B model shows a 6.1% improvement (from 42.6 to 45.2). These results confirm that incorporating previously learned content during later training stages prevents catastrophic forgetting and maintains training stability, aligning with our ACS stability principles.

4.4.3 Cross-Model Generalizability

To address concerns about the generalizability of our approach across different model architectures, we conducted additional experiments using DeepSeek-Math-7B-Instruct as the base model. This evaluation is particularly important given that prior work has shown some models may exhibit varying sensitivity to training signals in reinforcement learning paradigms. Additionally, to provide comprehensive comparison baselines, we include results from Light-R1 (the best performing difficulty calibration method from our main experiments) and Discard (the best performing challenging sample processing strategy from our comparative analysis).

As demonstrated in Table 2, our ACS framework maintains its effectiveness when applied to the DeepSeek-Math model architecture and achieves the highest average performance across all methods. Compared to the GRPO baseline, ACS achieves a significant improvement of 8.9% (from 17.9 to 19.5). ACS also outperforms other competitive methods, with improvements of 5.4% over Light-R1 (from 18.5 to 19.5) and 4.3% over the Discard strategy (from 18.7 to 19.5).

These results validate that our proposed ACS framework exhibits robust cross-model generalizability, addressing the limitation of model-specific optimization strategies. The consistent performance improvements across different architectural foundations demonstrate that the core principles of Guided Prompting and model-adaptive difficulty calibration are broadly applicable to various mathematical reasoning models, rather than being artifacts of specific model characteristics.

4.4.4 Algorithm Generalizability

To validate the generalizability of ACS beyond GRPO, we conducted additional experiments using Proximal Policy Optimization (PPO) on the Qwen2.5-Math-1.5B model. PPO represents the most classic and widely-adopted RL baseline in the field. As shown in Table 3, ACS demonstrates consistent stability-preserving benefits across different RL algorithms, with PPO w/ ACS achieving 25.5% improvement (from 28.2 to 35.4) over vanilla PPO.

5 Conclusions

We presented Adaptive Curriculum Strategies (ACS), a framework centered on a simple but effective insight: difficult samples in curriculum RL should neither be discarded nor forced upon models unchanged, but transformed. Our core contribution, Guided Prompting, addresses this by prepending partial solution steps as hints to convert challenging samples into accessible learning opportunities, resolving the stability-utilization tension that existing approaches fail to escape. Model-specific difficulty calibration complements this by precisely identifying which samples require transformation for each individual model. Together, these components enable stable and sample-efficient progressive RL training, achieving consistent improvements across five mathematical reasoning benchmarks, multiple model scales, and different RL algorithms.

This work reframes the key challenge in curriculum-based RL from "how to order samples" to "how to handle samples that exceed current capability," opening new directions for designing stable and effective progressive training methodologies.

Limitations

Our current implementation evaluates difficulty calibration once before training begins, using a static snapshot of model capabilities. As the model evolves during training, some samples may shift between difficulty levels, potentially making the initial calibration less accurate in later stages—though our analysis in the Appendix shows overlap ratios remain above 80% across stages, suggesting this effect is limited in practice. Additionally, Guided Prompting relies on the availability of step-level reference solutions for decomposition; tasks where reference solutions are not easily decomposable into discrete steps may require alternative scaffolding strategies. Finally, our experiments focus on mathematical reasoning, and the effectiveness of ACS in other domains with different reward structures warrants future investigation.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

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

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. Deepseek-coder: When the large language model meets programming - the rise of code intelligence. *ArXiv*, abs/2401.14196.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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 *Annual Meeting of the Association for Computational Linguistics*.

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

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, J. Hockenmaier, and Tong Zhang. 2025. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *ArXiv*, abs/2503.12759.

Koray Kavukcuoglu. 2025. Gemini 2.5: Our most intelligent ai model. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/. Accessed: 2025-05-17.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh 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. *ArXiv*, abs/2206.14858.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *ArXiv*, abs/2308.09583.

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

Jingyuan Ma, Rui Li, Zheng Li, Lei Sha, and Zhifang Sui. 2024. Plug-and-play training framework for preference optimization. *ArXiv*, abs/2412.20996.

OpenAI. 2024. Openai o1 system card. https://arxiv.org/abs/2412.16720. Accessed: 2025-05-17.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Feng Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *ArXiv*, abs/2501.12599.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Rui Min Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *ArXiv*, abs/2407.13690.

Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. 2025a. Sari: Structured audio reasoning via curriculum-guided reinforcement learning. *ArXiv*, abs/2504.15900.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025b. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *ArXiv*, abs/2503.10460.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *ArXiv*, abs/2502.14768.

Boyang Xue, Qi Zhu, Hongru Wang, Rui Wang, Sheng Wang, Hongling Xu, Fei Mi, Yasheng Wang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. 2025. Dast: Difficulty-aware self-training on large language models. *ArXiv*, abs/2503.09029.

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

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. *ArXiv*, abs/2409.12122.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, Tiantian Fan, Zhengyin Du, Xiang Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.

A Experimental Details

A.1 Dataset Construction

Following the experimental setting of Zeng et al. (2025), we selected the MATH dataset (Hendrycks et al., 2021) and extracted samples from level 3 to level 5 as training data, comprising a total of 9,255 instances. To implement our proposed ACS framework, we first perform model-specific difficulty calibration to identify which samples require Guided Prompting transformation—feeding all training set samples into the pre-trained model for inference and evaluating the model's accuracy on each sample.

To ensure that the evaluation results are as reliable as possible while not causing excessive computational overhead, for each question in the dataset, we use the VLLM framework to generate 16 responses from the model, extract predictions from these responses using appropriate scripts, and compare them with golden answers to determine the correctness of the generations. To fully harness the model's potential, we did not adopt a greedy decoding strategy to generate responses, but instead set the temperature to 0.7, generating responses through sampling.

After calculating the model's accuracy on the samples through the above steps, we sort the samples and divide them into 3 equal parts according to quantity. The top 1/3 with the highest accuracy are classified as accessible samples, used for the first stage of model training. The bottom 1/3 with the lowest accuracy are classified as challenging samples, used for the final stage of model training.

In addition, for particularly challenging samples, we employed our "Guided Prompting" approach to reduce the difficulty for the model while preserving learning value. Specifically, we first collected reference answers for these challenging samples, then segmented these reference answers into step-by-step reasoning processes, as illustrated in Figure 4.

Finally, we selected a small portion of the prefix combined with the original question as input to assist the model in solving problems more effectively while maintaining training stability.

All data was processed into a conversational format.

A.2 Model Selection and Hardware Setup

To effectively validate the efficacy of our ACS method across foundation models of varying capabilities, we selected three different models for our experiments: Qwen2.5-MATH-1.5B (Yang et al., 2024), Qwen2.5-MATH-7B, and DeepSeek-Math-7B-Instruct for cross-model generalizability validation. We conducted our experiments using 8 NVIDIA A100 GPUs for the GRPO experiments within Hugging Face's Open R1 framework (Face, 2025).

A.3 Computational Overhead Analysis

Our ACS framework requires evaluating model performance on training samples through multiple sampling, which introduces additional computational overhead. To minimize this cost while maintaining evaluation reliability, we employed the VLLM framework (Kwon et al., 2023) for efficient model inference acceleration.

Thanks to VLLM's optimized memory management and dynamic batching capabilities, the additional time overhead introduced by our curriculum construction is minimal. Taking the Qwen2.5-Math-7B model as an example, the sample evaluation phase for difficulty calibration requires less than 5% of the total GPU hours compared to the entire training process on NVIDIA A100 GPUs, making the overhead negligible compared to the overall computational cost.

Specifically, the time cost breakdown is as follows:

- **Sample evaluation phase**: Using VLLM, we generate 16 responses per sample for the 9,255 training instances, totaling approximately 148,080 inference calls.
- **Training phase**: Standard fine-tuning process using GRPO on the curriculum-organized data.

This efficient implementation ensures that the benefits of model-specific difficulty calibration can be achieved without significant computational burden, making our approach practical for real-world applications.

A.4 GRPO Formulation

We employ Group Relative Policy Optimization (GRPO) with a reward combining accuracy and format correctness: $r = r_{accuracy} + r_{format}$, where each component is binary (1.0 if correct, 0.0 otherwise). GRPO generates G candidate responses per question and computes relative advantages as:

$$A_i = \frac{r_i - \text{mean}(\{r_1, ..., r_G\})}{\text{std}(\{r_1, ..., r_G\})} \tag{3}$$

The optimization objective with clipping and KL regularization is:

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{(x)\sim D}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\big(\min(r_{i,t}(\theta)A_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_i) - \beta D_{KL}(\pi_\theta||\pi_{ref})\big)\Bigg], \tag{4}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance sampling ratio.

A.5 Training Hyperparameters

GRPO Training Details. We conducted our experiments using bf16 precision under the DeepSpeed framework with zero-2 configuration. We set per_device_train_batch_size to 16 and gradient_accumulation_steps to 8, employing a cosine lr_scheduler with warmup set to 0.1 and beta to 0.04, num_generations to 7, max_prompt_length to 512 and max_completion_length 1024. For the Qwen2.5-Math-1.5B model, we used a learning rate of 3e-6 and trained for 6 epochs. For the Qwen2.5-Math-7B model, we used a learning rate of 3e-6 and trained for 4 epochs. For the DeepSeek-Math-7B-Instruct model, we used the same configuration as the Qwen2.5-Math-7B model (learning rate of 3e-6, 4 epochs).

A.6 Evaluation

We evaluated our models using the evaluation script from Zeng et al. (2025). For evaluation, we use five benchmark datasets to assess the model's performance across different levels of difficulty and mathematical reasoning:

- **MATH 500** (Lightman et al., 2023): A subset of the MATH dataset, containing 500 representative problems designed to test a model's general mathematical capability.

Case Study

Problem
Let n be the smallest positive integer that satisfies the following conditions:
n divided by 2 is a perfect square
n divided by 3 is a perfect cube
n divided by 5 is a perfect fifth power
How many divisors of n are NOT multiples of 10?

Solution
The first condition implies that the power of each prime factor of n must be an even power (excluding 2, which must be an odd power). The second condition implies that the power of each prime factor of n must be divisible by 3 (excluding 3, which must leave a residue of 1 upon division by 3). The third condition implies that the power of each prime factor of n must be divisible by 5 (excluding 5, which must leave a residue of 1 upon division by 5).\nClearly, to minimize n, we want to just use the prime factors $2,3,5$. The power of 2 must be divisible by $3,5$, and 2^{15} works. Similarly, the powers of 3 and 5 must be 10 and 6, respectively, both of which leave a residue of 1 upon division. Thus, we need the number of factors of $2^{15} \\cdot 3^{10} \\cdot 5^{6}$ which are not multiples of 10.\nApplying the complement principle, there are a total of $(15+1)(10+1)(6+1) = 1232$ factors. We can draw a bijection between the number of divisors of $2^{15} \\cdot 3^{10} \\cdot 5^{6}$ that are divisible by 10 and the number of divisors of $2^{14} \\cdot 3^{10} \\cdot 5^{5}$ (as each of these divisors, when multiplied by 10, will provide a factor of the original number that is divisible by 10). There are $(14+1)(10+1)(5+1) = 990$. The answer is $1232-990 = \\boxed{242}$.

Step 1
The first condition implies that the power of each prime factor of n must be an even power (excluding 2, which must be an odd power).
Step 2
The second condition implies that the power of each prime factor of n must be divisible by 3 (excluding 3, which must leave a residue of 1 upon division by 3).
Step 3
The third condition implies that the power of each prime factor of n must be divisible by 5 (excluding 5, which must leave a residue of 1 upon division by 5).
Step 4
Clearly, to minimize n, we want to just use the prime factors $2,3,5$.
Step 5
The power of 2 must be divisible by $3,5$, and 2^{15} works.
Step 6
Similarly, the powers of 3 and 5 must be 10 and 6, respectively, both of which leave a residue of 1 upon division.
Step 7
Thus, we need the number of factors of $2^{15} \cdot 3^{10} \cdot 5^{6}$ which are not multiples of 10.
Step 8
Applying the complement principle, there are a total of $(15+1)(10+1)(6+1) = 1232$ factors.
Step 9
We can draw a bijection between the number of divisors of $2^{15} \cdot 3^{10} \cdot 5^{6}$ that are divisible by 10 and the number of divisors of $2^{14} \cdot 3^{10} \cdot 5^{5}$ (as each of these divisors, when multiplied by 10, will provide a factor of the original number that is divisible by 10).
Step 10
There are $(14+1)(10+1)(5+1) = 990$.
Step 11
The answer is $1232-990 = \boxed{242}$.

Figure 4: Decomposition of reference answers into step-by-step solution.

GRPO Prompt

You are a helpful AI Assistant that provides well-reasoned and detailed responses.
You first think about the reasoning process as an internal monologue and then provide the user with the answer.
Respond in the following format:
<think>
reasoning process here
</think>
<answer>
answer here
</answer>
User:
{Problem}
Assistant:
Let me solve this step by step.
<think>

Figure 5: Prompt template used in GRPO training for ACS implementation.

- **OlympiadBench** (He et al., 2024): Includes a collection of problems from Olympiad-level mathematics and physics competitions.
- **Minerva Math** (Lewkowycz et al., 2022): A curated set of undergraduate-level math problems that assess complex mathematical reasoning and symbolic manipulation.
- **AMC 23** and **AIME 24**: Include problems from the 2023 American Mathematics Competitions and the 2024 American Invitational Mathematics Examination, respectively.

A.7 Prompt Details

During both training and testing processes, the data was processed into a conversational format. Figure 5 demonstrate the prompts we used during the GRPO processes respectively. After training the models using their respective methods, we employed the corresponding prompts during testing as well. Additionally, during the GRPO training process, besides adding the User's description, we also appended part of the Assistant's content prefixed with the special token `<think>`. This approach helps the model quickly learn format compliance during the reinforcement learning process, greatly enhancing the stability of the model's reinforcement learning training within our ACS framework.

B Additional Results

B.1 Limitations of Predefined Difficulty Metrics

Predefined difficulty metrics exhibit fundamental flaws that undermine their effectiveness in curriculum learning applications. First, these metrics lack precision in capturing actual problem complexity as experienced by language models. As demonstrated in Figure 6, Our systematic evaluation on the MATH dataset reveals that model performance does not correlate with predefined difficulty rankings. Most notably, models consistently achieve higher accuracy on supposedly more difficult Level 5 problems compared to Level 4 problems, directly contradicting the assumed difficulty hierarchy. This counterintuitive pattern indicates that expert-defined difficulty levels may not align with the computational challenges actually faced by neural models.

Second, the assumption of universal difficulty standards proves fundamentally flawed in practice. Our analysis reveals significant variation in how different models perceive problem complexity, with difficulty assessments that effectively characterize challenge levels for one architecture often failing to generalize to other models. This model-specific variation in difficulty perception explains why curriculum strategies based on fixed difficulty rankings produce inconsistent training outcomes across different architectures, highlighting the critical need for adaptive, model-aware difficulty calibration approaches.

B.2 Training Progression

In this section, we demonstrate the overall performance changes on the test set when applying our ACS framework to Qwen2.5-Math-1.5B and Qwen2.5-Math-7B using reinforcement learning methods for multi-stage training. As shown in Fig-

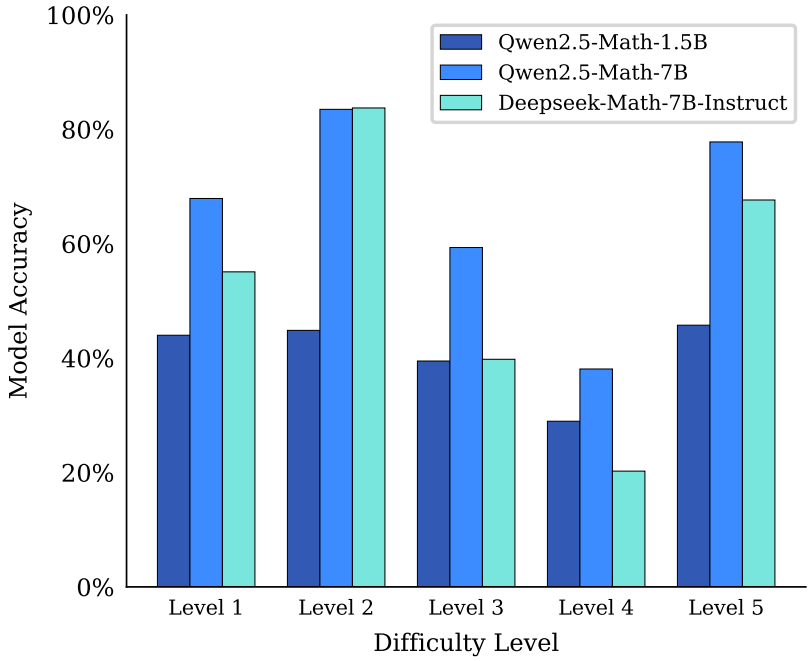

Figure 6: Performance of multiple models on MATH dataset subsets with predefined difficulty levels. As predefined difficulty increases from Level 1 to Level 5, model accuracy does not consistently decline but instead exhibits significant fluctuations, demonstrating that predefined difficulty standards may not correctly adapt to all models.

Table 6: Difficulty level stability across training stages. The table shows the percentage of samples that remain in the same difficulty level after each training stage, demonstrating the robustness of our difficulty calibration.

TRAINING STAGE	LEVEL 1 OVERLAP	LEVEL 2 OVERLAP	LEVEL 3 OVERLAP	AVERAGE
INITIAL	100%	100%	100%	100%
AFTER STAGE 1	87.4%	75.7%	91.6%	84.9%
AFTER STAGE 2	82.7%	80.5%	88.4%	83.9%
AFTER STAGE 3	81.2%	79.6%	84.1%	81.6%

ure 7, our ACS method maintains stable performance progression as training iterations advance, demonstrating the effectiveness of our stability-preserving curriculum design.

B.3 Difficulty Stability Across Training Stages

To address concerns about the dynamic nature of difficulty assessment, we conducted an additional experiment to track how sample difficulty changes across training stages. Specifically, we measured the overlap of samples within each difficulty level before and after each training stage. As shown in Table 6, the results demonstrate that after multiple training stages, the majority of samples remain within their original difficulty levels, with average overlap ratios exceeding 80% across all stages.

Further analysis reveals that difficulty transitions occur predominantly between adjacent levels. From initial to Stage 3, only 1.3% of Level 1 samples migrated to Level 3. Notably, difficult samples (Level 3) exhibit the highest stability, with 84.1% remaining in Level 3 after all training stages. This observation reinforces the importance of proper adaptation strategies for difficult samples through our Guided Prompting mechanism.

B.4 Calibration Robustness Analysis

We conducted comprehensive robustness analysis to examine how variations in sampling parameters affect the consistency of our difficulty calibration. Using our default configuration ($n = 16, T = 0.7$) as reference, we measure the overlap ratio between partitions generated with different parameters.

As demonstrated in Table 7, our results show high consistency across different parameter choices, with average overlap ratios exceeding 93% in all configurations. This indicates that our difficulty calibration approach produces stable and reliable sample partitioning that is robust to reasonable variations in sampling parameters. The con-

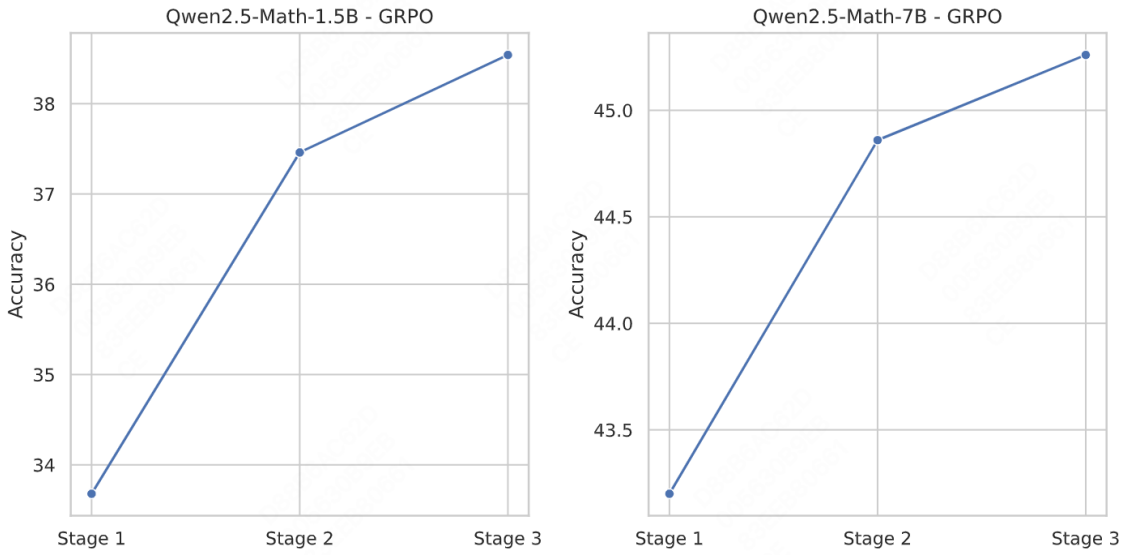

Figure 7: Performance progression across training stages using ACS framework, demonstrating stable improvement without catastrophic collapse.

Table 7: Calibration robustness to sampling parameters. Overlap ratios show the percentage of samples that remain in the same difficulty level when using different sampling configurations.

CONFIGURATION	STAGE I	STAGE II	STAGE III	AVERAGE
Robustness to Number of Samples (n)				
$n = 8$	94.2%	91.5%	93.8%	93.2%
$n = 16$ (OURS)	100%	100%	100%	100%
$n = 32$	97.5%	95.1%	96.9%	96.5%
Robustness to Sampling Temperature (T)				
$T = 0.5$	95.3%	92.8%	94.6%	94.2%
$T = 0.7$ (OURS)	100%	100%	100%	100%
$T = 0.9$	96.1%	93.5%	95.2%	94.9%

figuration ($n = 16$, $T = 0.7$) achieves an optimal balance between reliable assessment and computational efficiency while maintaining stability.

B.5 Number of Curriculum Stages

We conducted ablation experiments on the number of curriculum stages to examine the impact of curriculum granularity. As shown in Table 8, the 3-stage configuration achieves optimal performance (38.5 average), while both 2-stage (33.9) and 4-stage (36.8) configurations show degraded results. Too few stages (2) provide insufficient progressive adaptation, while too many stages (4) may introduce unnecessary complexity and transition overhead.

B.6 Dynamic Curriculum Learning Comparison

To provide comprehensive comparison with state-of-the-art adaptive curriculum approaches, we evaluated AdaRFT (Shi et al., 2025), which implements dynamic curriculum learning that continuously adapts training data to the model's evolving capabilities. As shown in Table 9, while AdaRFT demonstrates effectiveness through dynamic scheduling, our ACS framework achieves superior performance through Guided Prompting for stability-preserving sample transformation, complemented by model-specific difficulty calibration for precise identification of samples requiring transformation.

Compared with AdaRFT, ACS achieves improvements of 3.8% (from 37.1 to 38.5) for Qwen2.5-Math-1.5B and 6.8% (from 42.4 to 45.3) for Qwen2.5-Math-7B. These results demonstrate that Guided Prompting's sample transformation mecha-

Table 8: Ablation study on the number of curriculum stages using Qwen2.5-Math-1.5B model

STAGES	MATH 500	MINERVA	OLYMPIAD	AIME24	AMC23	AVG
2	67.6	27.6	30.1	6.7	37.5	33.9
3	**72.6**	**31.6**	**32.7**	**13.3**	**42.5**	**38.5**
4	71.8	30.8	31.5	10.0	40.0	36.8

Table 9: Comparison with AdaRFT dynamic curriculum baseline across two model scales

MODEL	METHOD	MATH 500	MINERVA	OLYMPIAD	AIME24	AMC23	AVG
QWEN2.5	ADARFT	68.8	29.4	32.1	10.0	**45.0**	37.1
1.5B	ACS	**72.6**	**31.6**	**32.7**	**13.3**	42.5	**38.5**

nism provides complementary benefits beyond pure dynamic scheduling strategies, validating the effectiveness of combining stability-preserving sample transformation with adaptive difficulty assessment.

B.7 Guided Prompting Parameter Analysis

We conducted comprehensive ablations over hint ratio (α) and accuracy threshold (τ) to analyze the trade-offs in our Guided Prompting mechanism. In all our main experiments, we set $\alpha = 25\%$ and $\tau = 20\%$.

As shown in Table 10, the results reveal a clear trade-off in hint ratio selection: (1) Too few hints ($\alpha = 10\%$) under-utilize the stabilization benefit, limiting the model's ability to learn from challenging problems; (2) Too many hints ($\alpha = 40\%$) lead to performance degradation, suggesting distribution shift where models become overly dependent on hint structure; (3) The optimal balance at $\alpha = 25\%$ provides sufficient scaffolding while maintaining adequate challenge. During inference, our models receive only original problem statements without any hints, and the strong generalization results validate that models do not develop inference-time dependency on hint structures.

Table 10: Ablation study on Guided Prompting parameters using Qwen2.5-Math-1.5B model. The results demonstrate the optimal balance achieved by our default configuration ($\alpha = 25\%$, $\tau = 20\%$).

CONFIGURATION	MATH 500	MINERVA	OLYMPIAD	AIME24	AMC23	AVG
$\alpha = 10\%$, $\tau = 20\%$	71.2	30.2	31.8	10.0	40.0	36.6
$\alpha = 25\%$, $\tau = 20\%$ (OURS)	**72.6**	**31.6**	**32.7**	**13.3**	**42.5**	**38.5**
$\alpha = 40\%$, $\tau = 20\%$	69.4	29.8	29.4	10.0	35.0	34.7
$\alpha = 25\%$, $\tau = 30\%$	70.8	30.5	30.9	13.3	40.0	37.1