AdaCuRL: Adaptive Curriculum Reinforcement Learning with Invalid Sample Mitigation and Historical Revisiting

Renda Li, Hailang Huang, Fei Wei*, Feng Xiong, Yong Wang*, Xiangxiang Chu

AMAP, Alibaba Group

{lirenda.lrd, huanghailang.hhl, xixia.wf, jingxun.xf, wangyong.lz, chuxiangxiang.cxx}@alibaba-inc.com

Abstract

Reinforcement learning (RL) has demonstrated considerable potential for enhancing reasoning in large language models (LLMs). However, existing methods suffer from Gradient Starvation and Policy Degradation when training directly on samples with mixed difficulty. To mitigate this, prior approaches leverage Chain-of-Thought (CoT) data, but the construction of high-quality CoT annotations remains labor-intensive. Alternatively, curriculum learning strategies have been explored but frequently encounter challenges, such as difficulty mismatch, reliance on manual curriculum design, and catastrophic forgetting. To address these issues, we propose AdaCuRL, a **Ada**ptive **Cu**rriculum **R**einforcement **L**earning framework that integrates coarse-to-fine difficulty estimation with adaptive curriculum scheduling. This approach dynamically aligns data difficulty with model capability and incorporates a data revisitation mechanism to mitigate catastrophic forgetting. Furthermore, AdaCuRL employs adaptive reference and sparse KL strategies to prevent Policy Degradation. Extensive experiments across diverse reasoning benchmarks demonstrate that AdaCuRL consistently achieves significant performance improvements on both LLMs and MLLMs.

1 Introduction

Post-training methods designed to enhance complex reasoning capabilities have emerged as a prominent area of research. Supervised fine-tuning (SFT) typically distills expert models to obtain high-quality reasoning trajectories for achieving satisfactory performance. In contrast, RL-based approaches, exemplified by GRPO (Guo et al. 2025), demonstrate that models can self-improve reasoning through RL without relying on high-quality distillation data. This has inspired numerous subsequent efforts on both LLMs (Dang and Ngo 2025; Chu et al. 2025) and MLLMs (Chen et al. 2025; Meng et al. 2025).

Despite the notable success of RL-based methods, their performance is critically dependent on the training data curriculum. A critical bottleneck emerges when models are trained on mixed-difficulty data, leading to severe **Data Inefficiency**. This inefficiency stems from an intrinsic coupling between sample difficulty and the relative advantages of rollouts within GRPO groups. Specifically, when training samples exhibit extreme difficulty levels relative to the current policy, the reward signal often collapses into a binary state, where simple samples uniformly receive rewards of 1, while difficult ones invariably yield rewards of 0 (Chu et al. 2025). These invalid samples culminate in two core dilemmas: **Gradient Starvation** and **Policy Degradation**. (i) Gradient Starvation occurs when all exploration rollouts produce rewards of 1 or 0, causing the advantage function to collapse to zero. Consequently, the policy gradient is nullified, depriving the model of any meaningful learning signal. (ii) Policy Degradation arises when the KL divergence penalty imposed on invalid samples dominates the optimization signals. This forces the policy to revert to a conservative reference model, which impairs the reasoning capability obtained through RL. As illustrated in Figure 1, these invalid samples are common in standard datasets, highlighting the urgent need for an improved training paradigm.

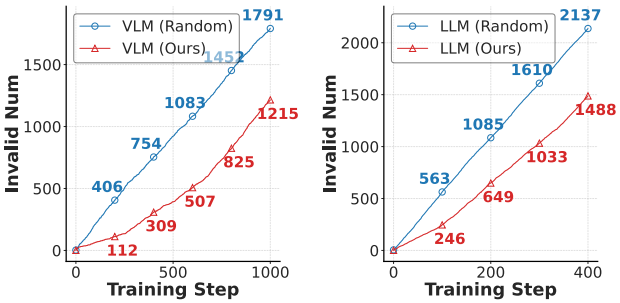

Figure 1: Cumulative invalid samples during GRPO training: shuffled data (Baseline) vs curriculum learning (Ours) on standard open-source datasets (VLM: Qwen2.5-VL-3B; LLM: Qwen2.5-Math-7B).

To alleviate Gradient Starvation, it is necessary to avoid rollouts producing all-zero or all-one rewards. Hint-GRPO (Huang et al. 2025a) incorporates expert reasoning trajectories to avoid difficult samples receiving all-zero rewards. However, this approach does not recognize the importance of aligning model capability and sample difficulty during the RL process. The methods based on curriculum learning aim to train on samples matched to the model's

*Project leads and corresponding authors.

capacity to reduce invalid samples. However, existing approaches face three primary limitations: (i) difficulty mismatch, (ii) manual curricula, and (iii) forgetting of past samples. Some works define difficulty using human prior knowledge (Deng et al. 2025; Song et al. 2025) or expert models (Shi et al. 2025), while failing to capture how the model itself perceives difficulty. Additionally, other works (Team et al. 2025; Deng et al. 2025) employ handcrafted curricula as training schedulers, without considering model feedback. Moreover, all the above methods typically train within a specific difficulty range, neglecting earlier data, and as training shifts toward harder samples, performance on easier data may deteriorate. Although the above methods reduce the frequency of invalid samples, they lack effective mechanisms for addressing Policy Degradation caused by the occurrence of invalid samples.

To address the above issues, we propose AdaCuRL, a novel curriculum reinforcement learning approach. Specifically, AdaCuRL introduces a coarse-to-fine difficulty estimation strategy that can sample data with desired difficulty distributions from large-scale datasets and accurately estimate sample difficulty. During fine-tuning, AdaCuRL partitions the data into buckets from easy to hard and updates the specific bucket based on model feedback to avoid invalid samples. Besides, the bucket update mechanism allows revisiting historical data to mitigate catastrophic forgetting. To prevent degradation from invalid samples, AdaCuRL incorporates a sparse KL mechanism. Furthermore, we introduce self-pacing mechanism into AdaCuRL, called Re-AdaCuRL, which enhances data utilization and continuously improves reasoning.

Our main contributions are as follows:

- We propose AdaCuRL, which integrates a coarse-to-fine difficulty estimation strategy and a novel curriculum RL algorithm to enhance data efficiency in GRPO, enabling (M)LLMs to progressively improve their reasoning capabilities.
- We design sparse KL to effectively prevent Policy Degradation and propose an adaptive reference strategy to avoid excessive alignment with the reference model.
- We further introduce Re-AdaCuRL, which iteratively re-estimates sample difficulty and conducts curriculum RL to mine data and strengthen reasoning. Extensive experiments across diverse benchmarks demonstrate the effectiveness of our approach on both MLLMs and LLMs.

2 Preliminary

Curriculum Learning (CL) is a training strategy inspired by human incremental learning, where models are trained sequentially on data ordered by difficulty to achieve better performance and faster convergence. CL comprises two main components, difficulty estimation and a training scheduler, and both can be categorized as either predefined or automatic. Formally, given a dataset:

$$\mathcal{D} = \{(x_i, y_i, d_i)\}_{i=1}^{N}, \tag{1}$$

where d_i is the difficulty of sample (x_i, y_i) and N is the dataset size, the curriculum ensures: $d_1 \leq d_2 \leq \cdots \leq d_N$.

Self-Paced Learning (SPL) automates difficulty estimation by selecting samples according to current loss. Formally, SPL constructs training sets for each epoch as:

$$\mathcal{D} = \{(x_i, y_i) \mid \ell(f_{\mathbf{w}}(x_i), y_i) \leq \tau\}, \tag{2}$$

where τ is an adaptive loss threshold.

Predefined schedulers select samples according to fixed rules (Dong et al. 2025), whereas automatic ones (Graves et al. 2017) make scheduling decisions based on the model's feedback.

Group Relative Policy Optimization (GRPO) eliminates the need for a value model by normalising outcome rewards within a group of G samples and applying a policy-gradient objective regularised by a KL term.

For a prompt q, the policy π_θ generates G responses $\{o_i\}$ with scalar rewards $\{r_i\}$. Let μ_r and σ_r denote the group mean and standard deviation. GRPO defines the group-relative advantage: $\hat{A}_i = \frac{r_i - \mu_r}{\sigma_r + \varepsilon}$, where $\varepsilon > 0$ prevents division by zero. We define $\rho_i = \pi_\theta(o_i \mid q) / \pi_{\text{old}}(o_i \mid q)$ as the importance ratio between the learned policy π_θ and a fixed reference policy π_{ref}, and $\text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)$ as the CLIP operation. The objective of GRPO is then expressed as:

$$\begin{aligned}\mathcal{L}_{\text{GRPO}}(\theta) = &-\mathbb{E}_i\big[\min(\rho_i \cdot \hat{A}_i, \text{CLIP} \cdot \hat{A}_i)\big] \\ &+ \beta\, \mathbb{E}_i\Big[\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big],\end{aligned} \tag{3}$$

where β controls the KL regularization strength.

3 Method

AdaCuRL consists of three key components. First, we introduce a coarse-to-fine difficulty estimation strategy to effectively extract subsets with a target difficulty distribution from large-scale datasets. Then, we present the core training scheduling algorithm, which serves as the central framework of AdaCuRL. Finally, we extend this framework with Re-AdaCuRL, an enhanced variant designed to further optimize data utilization for improved reasoning capabilities.

3.1 Coarse-to-Fine Difficulty Estimation

Accurate difficulty estimation is essential for effective curriculum learning. We adopt an unbiased approach to evaluate problem difficulty based on the frequency of correct solutions generated by the base model across multiple attempts (Snell et al. 2024; Shi et al. 2025).

Curriculum learning typically requires sampling from the training set to form a specified difficulty distribution (e.g., containing more hard problems than easy ones). However, given the large dataset size, precisely estimating each sample's difficulty by generating answers multiple times incurs substantial inference overhead, while random sampling fails to match the desired difficulty distribution. To address this, we propose a coarse-to-fine difficulty estimation strategy.

- **Coarse stage.** For each problem, the model produces five answers. Based on the number of correct answers, we assign each problem to one of three bins. We then sample from each bin according to a predefined ratio, while ensuring that the selected samples remain evenly

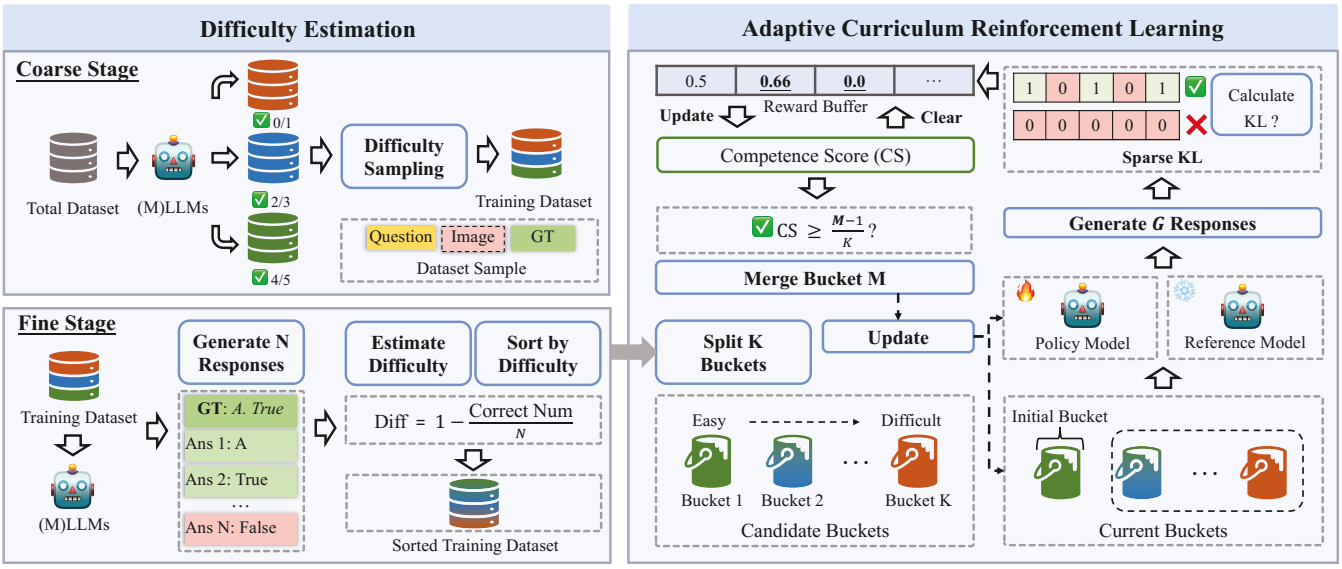

Figure 2: The overall framework of AdaCuRL. Difficulty Estimation (left) samples a training subset from a large-scale dataset to match a target difficulty distribution and sorts the data from easy to hard. Curriculum Reinforcement Learning (right) monitors the average accuracy reward during training to assess the model's mastery of the current difficulty level and progressively introduces more challenging samples. In addition, AdaCuRL incorporates sparse KL and adaptive reference mechanisms to prevent degradation of the model's reasoning capability.

distributed across datasets. Formally, let $c_i \in \{0, \ldots, 5\}$ be the number of correct answers for problem i. We define three bins: $\mathcal{G}_1 = \{i \mid c_i \in \{0, 1\}\}$, $\mathcal{G}_2 = \{i \mid c_i \in \{2, 3\}\}$, and $\mathcal{G}_3 = \{i \mid c_i \in \{4, 5\}\}$, and draw

$$\mathcal{S} = \bigcup_{k=1}^{3} \text{Sample}\Big(\mathcal{G}_k,\ n_k = \lfloor \rho_k \, |\mathcal{G}_k| \rfloor\Big), \tag{4}$$

where ρ_k is the predefined sampling ratio for k-th bin.

- **Fine stage.** For each problem in $\mathcal{S}$, we generate N ($N \gg 5$) answers for precise difficulty estimation. Let $c(q)$ denote the number of correct solutions out of these N attempts for problem q. We define its difficulty score as

$$\text{Difficulty}(q) \ = \ 1 - \frac{c(q)}{N}, \tag{5}$$

and then filter out problems with difficulty above 0.95 or below 0.05 to avoid overly hard or trivial cases. The remaining data are sorted by ascending difficulty to form the final training dataset $\mathcal{D}$.

3.2 Curriculum Reinforcement Learning

After sorting $\mathcal{D}$ by difficulty, we partition it into K consecutive buckets $\{\mathcal{B}1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}$ with equal size:

$$\mathcal{B}_k = \Big\{ q_{(k-1)\frac{|\mathcal{D}|}{K}+1}, \ldots, q_{k\frac{|\mathcal{D}|}{K}} \Big\}, \quad k = 1, \ldots, K, \tag{6}$$

where $q_1, \ldots, q_{|\mathcal{D}|}$ are ordered from easy to hard.

The current training subset $\mathcal{D}_c$ is initialized as the first bucket $\mathcal{B}_1$. During training, we merge the next bucket into $\mathcal{D}_c$ at each update stage t to mitigate catastrophic forgetting:

$$\mathcal{D}_c^{(t+1)} = \mathcal{D}_c^{(t)} \cup \mathcal{B}_{t+2}, \quad t = 0, \ldots, K-2. \tag{7}$$

The training processs continues until $\mathcal{D}_c = \mathcal{D}$. This incremental expansion retains knowledge from easier samples during initial stages, while gradually adding more challenging samples as the model's reasoning capabilities improve.

Reward Function. We use two binary reward signals: *format reward* and *accuracy reward*. We observe that the format reward converges rapidly, while the accuracy reward, especially from harder buckets, remains relatively low and progresses slowly. This imbalance affects the advantage function in Equation (3), dominated by the format reward and hindering π_θ from learning accurate reasoning paths effectively. To address this, we update the policy solely based on the accuracy reward after T_f training steps.

Bucket Update Strategy. A *naive* bucket update strategy trains each bucket sequentially. Such a schedule is often inefficient and sub-optimal because it ignores the *current state* of the model, leading to over-training of easy buckets while hard buckets may receive insufficient updates.

To adaptively assess progress, we use the *accuracy reward* during training to measure how well the model has mastered the current bucket and record the reward of each sample in the rewards buffer $\mathcal{R}_b$. Specifically, we maintain a *competence score* $cs \in [0, 1]$, which is initialized as $cs^{(0)} = 0$ and updated as:

$$cs^{(t+1)} \ \leftarrow \ cs^{(t)} + (\bar{r} - 0.5) \times \max\Big(1 - cs^{(t)},\ \gamma\Big) \tag{8}$$

where $\bar{r}$ is the average reward over the most recent M training samples and $\max(1 - cs, \gamma)$ acts as a decay factor on the update rate. As cs increases, the update step becomes smaller, mimicking human learning by spending more time

on harder buckets, while γ prevents the rate from becoming too small.

Once $\mathcal{R}_b$ contains M samples, we update cs and check whether the curriculum set $\mathcal{D}_{\text{c}}$ should be expanded. The curriculum expansion condition is defined as follows:

$$cs \;\geq\; \frac{k-1}{K} \tag{9}$$

When the condition in Equation (9) is satisfied for the next bucket index k, bucket $\mathcal{B}_k$ is merged into $\mathcal{D}_{\text{c}}$.

To keep the estimate of $\bar{r}$ faithful to the model's ability on *newly introduced* data, only samples drawn from the *latest merged bucket* contribute to $\bar{r}$. Upon merging $\mathcal{B}_k$, the competence score is re-initialized to $cs = \frac{k-1}{K}$ to ensure an accurate reflection of the policy model's mastery over the data in the newly added bucket.

KL Divergence Design. In Equation (3), GRPO calculates the KL divergence with the base model during each loss computation, leading to two issues: (i) When the advantage function is a full zero vector, the loss is dominated by the KL term, causing the policy model to unnecessarily align with the base model, (ii) as the model's reasoning ability improves, continuing to compute KL divergence with the base model undermines the already acquired reasoning capabilities. To address these limitations, we introduce two strategies into our proposed framework:

- **Conditional KL computation**. When all rewards within a rollout group are either 0 or 1, we exclude the KL divergence term from the loss computation for that specific group, enabling more effective enhancement of the model's reasoning abilities. The GRPO loss in AdaCuRL is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{GRPO}}(\theta) = &-\mathbb{E}_i\big[\min(\rho_i \cdot \hat{A}_i, \text{CLIP} \cdot \hat{A}_i)\big] \\ &+ \mathbb{I}\big[\hat{A}_i \neq 0\big]\, \beta\, \mathbb{E}_i\Big[\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big]\end{aligned} \tag{10}$$

- **Reference model resetting**. After each bucket update, the reference model π_{ref} is reset to the current policy model π_θ, thus avoiding excessive alignment with the initial reference model as the reasoning capability of π_θ improves.

3.3 Self-pacing Mechanism

After the first round of training with coarse-to-fine difficulty estimation and curriculum RL, the model develops stronger reasoning capabilities. To further improve performance, we introduce a *self-pacing* mechanism, called Re-AdaCuRL.

Specifically, we refine the coarse-to-fine difficulty estimation using the updated policy model π_θ and filter out previously trained data during sampling. Let $\text{Difficulty}^{(1)}(q)$ denote the re-estimated difficulty score. To preserve acquired reasoning capabilities, we discard samples with difficulty scores below a threshold (e.g., 0.2) in the second iteration:

$$\mathcal{D}' = \{\, q \in \mathcal{D} \mid \text{Difficulty}^{(1)}(q) \geq 0.2 \,\}. \tag{11}$$

The remaining data $\mathcal{D}'$ is then sorted and repartitioned into K buckets in ascending order of updated difficulty:

$$\mathcal{B}_k^{(1)} = \Big\{ q^{(1)}_{(k-1)\frac{|\mathcal{D}'|}{K}+1}, \ldots, q^{(1)}_{k\frac{|\mathcal{D}'|}{K}} \Big\}, \quad k = 1, \ldots, K. \tag{12}$$

We then repeat the training process described in Sec. 3.2 on these updated buckets. This self-pacing mechanism allows data that was previously excluded due to excessive difficulty to be revisited, while simultaneously filtering out samples already solved with high confidence. As a result, the current policy π_θ continues to train on increasingly informative data, further enhancing its reasoning capabilities.

4 Experiments

4.1 Datasets

For training MLLMs, we curate a training dataset from a broad range of mathematical reasoning sources, including CLEVR (Johnson et al. 2017), CLEVR-Math (Lindström and Abraham 2022), Geo3K (Lu et al. 2021a), GeoMverse (Kazemi et al. 2023), GeoQA+ (Chen et al. 2021), IconQA (Lu et al. 2021b), Super-CLEVR (Li et al. 2023), TabMWP (Lu et al. 2022), UniGeo (Chen et al. 2022), GEOS(Seo et al. 2015), WeMath (Qiao et al. 2024), SceMQA (Liang et al. 2024), and PolyMath (Gupta et al. 2024). Together, they comprise about 100K problems spanning various types (e.g., geometry, algebra, counting) and difficulty levels. To align with GRPO, we filter out samples whose answers cannot be reliably validated. For multiple-choice questions, we standardize the format to explicitly include both the option label and content (e.g., "A. 1.8"), preventing the model from exploiting superficial answer patterns. As detailed in Sec. 3.1, we partition the data into three coarse difficulty groups ($\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$). Following standard curriculum learning, we increase the proportion of harder samples by sampling 2K, 3K, and 5K examples from these groups, yielding a 10K training dataset.

For training LLMs, we utilize the Open-RS dataset (Dang and Ngo 2025), which contains 7K samples. Given its moderate size, we directly perform fine-grained difficulty estimation and sorting.

4.2 Benchmarks

For MLLMs, we build two complementary benchmarks: mathematical reasoning and general multimodal reasoning. For LLMs, we adopt standard mathematical reasoning benchmarks.

The multimodal mathematical reasoning benchmark comprises DynaMath (Zou et al. 2024), MathVista_MINI (Lu et al. 2023), Math-V (Wang et al. 2024a), MathVerse_MINI (Zhang et al. 2024a), and LogicVista (Xiao et al. 2024), and the multimodal general reasoning benchmark includes MMStar (Chen et al. 2024), MMMU (Yue et al. 2024), HallusionBench (Guan et al. 2024), AI2D (Kembhavi et al. 2016), and MMVET (Yu et al. 2023). For unimodal reasoning, we adopt standard datasets such as AIME24, AMC23, MATH500 (Lightman et al. 2023), Minerva (Lewkowycz et al. 2022), and Olympiadbench (He et al. 2024). Together, these benchmarks offer a comprehensive, multi-dimensional assessment of the models' reasoning capabilities.

Model	Mathematical Reasoning						General Reasoning					
	DynaMath	MathVista	Math-V	MathVerse	LogicVista	Avg.	MMStar	MMMU	Hallu.	AI2D	MMVET	Avg.
Qwen2.5-VL-3B Models												
Qwen2.5-VL-3B	40.90	62.00	22.62	33.75	38.70	39.59	56.00	50.88	45.66	80.40	60.20	58.63
+ SFT	38.74	60.60	22.27	34.37	41.61	39.52	58.00	51.11	49.88	79.60	63.71	60.46
+ GRPO	41.16	65.00	23.02	35.31	38.70	40.64	55.53	52.11	47.14	77.95	61.37	58.82
+ AdaCuRL (Easy)	45.44	64.10	22.10	37.00	39.37	41.60	57.60	52.00	**50.58**	81.60	61.78	60.71
+ AdaCuRL (Hard)	42.43	66.20	22.56	35.96	38.92	41.21	58.66	52.33	46.76	78.17	60.36	59.26
+ AdaCuRL	48.10	66.50	23.70	40.67	40.09	43.81	59.95	52.66	49.03	81.34	62.76	61.15
+ Re-AdaCuRL	**49.22**	**67.40**	**24.54**	**42.24**	**42.51**	**45.18**	**60.07**	**53.11**	48.27	**81.74**	**63.64**	**61.37**
Qwen2.5-VL-7B Models												
Qwen2.5-VL-7B	51.99	68.50	25.42	44.53	46.97	47.48	65.00	58.22	52.35	84.71	67.38	65.53
+ SFT	44.59	64.20	39.69	25.59	43.62	43.54	62.93	56.00	52.72	83.45	64.86	63.99
+ GRPO	48.12	70.90	26.94	47.22	45.41	47.72	63.06	57.44	54.42	83.29	69.03	65.45
+ AdaCuRL	55.10	70.40	27.07	**48.75**	48.10	49.88	**65.36**	**58.66**	**57.27**	**85.85**	69.31	**67.29**
+ Re-AdaCuRL	**56.67**	**71.60**	**28.92**	48.38	**48.99**	**50.91**	65.27	58.00	56.53	85.56	**69.91**	67.05

Table 1: Comparison of methods on mathematical (Left) and general (Right) reasoning benchmarks for MLLMs.

Model	AIME	AMC	MATH	Minerva	Oly.	Avg.
Qwen2.5-Math-1.5B Models						
Base Model	6.45	36.40	46.33	12.62	24.74	25.31
+ GRPO	7.50	40.62	56.00	12.99	27.25	28.87
+ AdaCuRL	**9.58**	45.63	**62.46**	**14.58**	29.33	**32.32**
- SparseKL	9.29	**45.71**	61.46	14.46	**29.53**	32.09
- Reset Ref	9.37	45.00	59.13	14.34	28.74	31.32
- Revisiting	8.13	44.22	60.46	13.60	29.18	31.12
Qwen2.5-Math-7B Models						
Base Model	15.83	51.87	64.66	17.40	29.18	35.79
+ GRPO	18.95	56.56	68.80	17.28	31.55	38.63
+ AdaCuRL	**22.22**	**59.22**	**74.53**	**27.33**	**37.48**	**44.16**

Table 2: Comparison of methods on mathematical reasoning benchmarks for LLMs. Results averaged over AIME24@16, AMC23@16, others@3. Oly. denotes Olympiad-bench.

4.3 Training Settings

We employ Qwen2.5-VL-3B-Instruct (Bai et al. 2025) and Qwen2.5-VL-7B-Instruct for multimodal experiments. For fine-grained difficulty estimation, we set $N = 100$ generations, format reward cutoff $T_f = 64$, decay $\gamma = 0.5$, and competence score interval $M = 512$. Unless specified, $K = 4$ buckets are used. For unimodal experiments, we use Qwen2.5-Math (1.5B and 7B) (Yang et al. 2024) as base models. Unlike Open-RS's cosine reward, we employ accuracy reward, maintaining consistency in other hyperparameters. We set $K = 3$ buckets, with other curriculum learning hyperparameters following the multimodal settings.

4.4 Main Results

Tables 1 and 2 present a comprehensive comparison of different methods across reasoning benchmarks on both multimodal and language models. The results are as follows.

Neither the original GRPO nor SFT significantly enhances reasoning capabilities. As shown in Table 1, the original GRPO improves mathematical and general reasoning by only 0.85% and 0.19%, respectively, on Qwen2.5-VL-3B, with similar results on the 7B model. The SFT baseline even leads to degraded performance, particularly on the larger 7B model. We hypothesize this degradation stems from fine-tuning on lower-quality open-source data, which may harm an already strong baseline. For language models, the original GRPO yields noticeable gains, improving by 3.56% on Qwen2.5-Math-1.5B and 2.84% on Qwen2.5-Math-7B (Table 2). We attribute this to the additional information fusion in multimodal models, which increases the difficulty of reinforcement fine-tuning.

AdaCuRL achieves outstanding performance. On both multimodal and language models, AdaCuRL outperforms baselines across all benchmarks and model sizes. For example, on mathematical reasoning, AdaCuRL improves by 3.17% and 2.16% on Qwen2.5-VL-3B and 7B, respectively, and achieves gains of 3.45% and 5.53% on Qwen2.5-Math-1.5B and 7B. These results highlight the importance of progressively increasing training difficulty to enhance reasoning and demonstrate the consistent applicability of AdaCuRL to both unimodal and multimodal tasks.

Group	Stage	clever math	geo 3k	geom verse	geoqa plus	icon qa
$\mathcal{G}_1$	Before	1142	1324	558	25678	3318
	After	849	1017	451	19221	2203
$\mathcal{G}_2$	Before	246	926	554	19502	6824
	After	135	985	408	19871	4050
$\mathcal{G}_3$	Before	1560	151	574	3880	10423
	After	1964	399	827	9968	14312

Table 3: Coarse-grained data distribution before and after one round of training with AdaCuRL.

Re-AdaCuRL further improves reasoning. As shown in Table 1, Re-AdaCuRL achieves additional improvements of 1.37% and 1.03% on mathematical reasoning for the Qwen2.5-VL-3B and 7B models, respectively. We further elaborate on the motivation for this approach. Table 3 shows a shift toward the easier end: samples in $\mathcal{G}_1$ decrease while those in $\mathcal{G}_3$ increase, indicating improved mathematical reasoning capabilities. To further leverage the dataset, we re-sample after re-estimating difficulty using the updated policy and continue training on the resampled data.

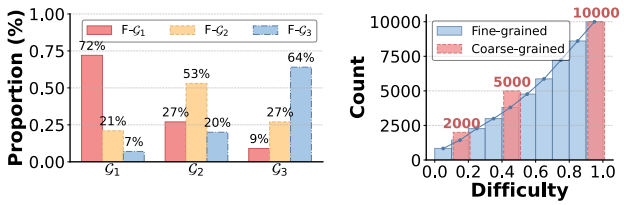

Figure 3: (Left) The proportion of samples from each of the three coarse-grained groups ($\mathcal{G}1/\mathcal{G}_2/\mathcal{G}_3$) that fall into each of the three fine-grained groups (F-$\mathcal{G}_1/\mathcal{G}_2/\mathcal{G}_3$) after fine-grained estimation. (Right) The difficulty distribution of coarse-grained sampling compared to that after fine-grained difficulty estimation.

5 Analysis

5.1 Evaluation of Difficulty Estimation

In this section, we evaluate whether the final difficulty distribution from fine-grained difficulty estimation aligns with the desired coarse-grained sampling distribution. As shown in the right part of Figure 3, the red regions represent the desired coarse-grained sampling distribution (i.e., $2\text{K}/3\text{K}/5\text{K}$), while the blue regions indicate the actual distribution obtained through fine-grained estimation. It is evident that the distributions are generally consistent. This demonstrates that the proposed coarse-to-fine difficulty estimation method achieves the desired difficulty distribution without large-scale inference. To further analyze, we use the fine-grained results as ground truth and evaluate the accuracy of the coarse-grained estimation by examining the proportion of fine-grained results within each coarse-grained group, as shown in the left part of Figure 3. The accuracy rates for the three groups $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ are 72%, 53%, and 64%, respectively. This indicates that coarse-grained estimation cannot provide precise difficulty assessments but effectively serves to obtain the desired difficulty distribution.

5.2 Different Difficulty Estimation Strategies

In addition to the fine-grained difficulty estimation based on the model itself, we explored two alternatives: (i) coarse-grained estimation only, and (ii) fine-grained estimation from a stronger external model. As shown in Table 4, both yield suboptimal results, highlighting that AdaCuRL relies on the model's own fine-grained estimation to provide accurate difficulty assessments for curriculum learning.

Model	Dyna Math	Math Vista	Math-V	Math Verse	Logic Vista	Avg.
3B + Fine	**48.10**	**66.50**	23.70	**40.67**	40.09	**43.81**
3B + Coarse	45.81	65.6	22.43	38.40	41.83	42.81
7B + Fine	47.84	63.9	**24.83**	38.82	**42.73**	43.62

Table 4: Results of training with different difficulty estimation strategies using Qwen2.5-VL-3B.

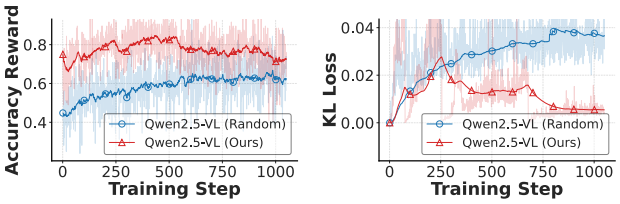

Figure 4: Training dynamics under AdaCuRL curriculum scheduling and randomly shuffled data. (Left) Accuracy reward. (Right) KL loss.

5.3 Training Comparison with Shuffled Data

Figure 4 compares training dynamics between AdaCuRL and training with randomly shuffled data. We observe that the curriculum scheduling in AdaCuRL enables the model to achieve higher average accuracy rewards through better alignment between model capability and sample difficulty. Furthermore, the adaptive reference strategy reduces average KL loss, preventing over-alignment with the base model and improving reasoning capability. These benefits ultimately result in superior performance on the test set.

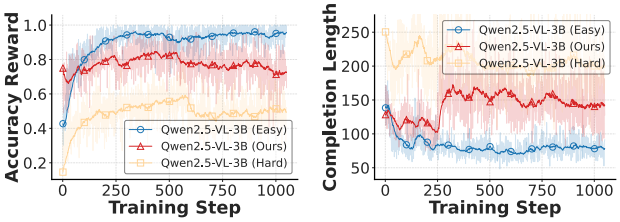

Figure 5: Reward and completion length during training with different difficulty distributions using Qwen2.5-VL-3B.

5.4 Difficulty Distribution

We further compare two alternative settings that exclusively use easy or hard samples. Specifically, we sample 10K training instances from $\mathcal{G}_3$ and $\mathcal{G}_1$, and train the model accordingly. We denote these as AdaCuRL (easy) and AdaCuRL (hard) in Table 1. Both variants underperform compared to the default difficulty setting, demonstrating the necessity of progressive difficulty data.

Figure 5 provides further analysis. Training on easy data yields high rewards but fails to develop deeper reasoning capabilities, evidenced by shorter reasoning lengths that decrease during training. In contrast, training solely on hard

data produces longer reasoning but often fails to reach correct answers, resulting in persistently low average rewards. With the default distribution, the model trains on appropriately challenging samples, maintaining high data utilization while steadily increasing reasoning length as harder data is gradually introduced.

Model	Dyna Math	Math Vista	Math-V	Math Verse	Logic Vista	Avg.
AdaCuRL	**48.10**	**66.50**	**23.70**	**40.67**	**40.09**	**43.81**
- SparseKL	47.26	65.60	22.43	38.68	38.15	42.42
- Reset Ref	44.65	63.90	22.46	37.95	38.93	41.58
- Revisiting	46.26	65.60	22.63	36.18	38.03	41.74
- KL	45.63	64.10	21.21	38.36	36.02	41.06

Table 5: Ablation results on mathematical reasoning benchmarks using Qwen2.5-VL-3B.

Model	# Revisit	# Degradation
Qwen2.5-VL-3B	5528	1048
Qwen2.5-VL-7B	5854	790

Table 6: Counts of revisits and reward degradations.

Base Model	Method	Avg.
Qwen2.5-VL-3B	naive CL	41.24
	AdaCuRL	**43.81**
Qwen2.5-Math-7B	naive CL	40.46
	AdaCuRL	**44.16**

Table 7: Ablation results on different training scheduler.

5.5 Ablation Study

Design of KL Divergence. We evaluate two KL-related mechanisms, including SparseKL and Adaptive Ref. Results in Tables 5 and 2 show that disabling either component degrades performance. We further observe that completely removing the KL divergence term from the loss results in a substantial performance drop. This is likely because revisiting earlier data amplifies overfitting to simpler samples, highlighting the necessity of the KL term in AdaCuRL.

Revisiting Historical Data. AdaCuRL revisits historical samples by merging the next bucket and resetting the training data, which helps mitigate forgetting. In this section, we analyze the forgetting issue and investigate an alternative strategy that keeps only the latest bucket without revisiting historical data. Table 6 shows statistics on the frequency of average group-reward decreases when previously seen samples were revisited, suggesting that training on harder samples can degrade performance on easier ones. Quantitative results in Tables 2 and 5 show that appropriately revisiting past data further boosts performance.

Dynamic Training Scheduler. AdaCuRL updates buckets dynamically based on average rewards during training. We also evaluate a naive curriculum strategy that processes samples from easy to hard using predefined buckets, without considering model feedback. As shown in Table 7, this approach consistently underperforms AdaCuRL across all models, highlighting the limitations of fixed schedules that overlook the model's evolving capabilities.

6 Related Work

6.1 Reasoning-oriented Reinforcement Learning

Reasoning for LLMs remains a central focus (Wang et al. 2024b; Saparov and He 2022; Xiong et al. 2025; Wang et al. 2025). CoT Prompting (Zhang et al. 2024b; Yao et al. 2023) guides models to reason step-by-step, while CoT Finetuning (Dong et al. 2025; Xu et al. 2024) fine-tunes models on large-scale CoT datasets. DeepSeek-R1 (Guo et al. 2025) demonstrates that RL can spontaneously induce strong reasoning abilities, reducing the need for extensive CoT data. However, since MLLMs typically possess limited initial reasoning skills, applying RL directly yields minimal improvements. This motivates studies (Yang et al. 2025; Huang et al. 2025b) to distill CoT data from DeepSeek-R1 or other reasoning-oriented models for SFT before RL, while Huang et al. (2025a) provides expert reasoning chains during RL to solve hard problems. However, these methods overlook the alignment between model capability and sample difficulty.

6.2 Curriculum Learning for RL

Curriculum learning (CL) (Bengio et al. 2009) trains models from easy to hard and is now broadly used in RL (Zhou et al. 2020; Wang et al. 2023). Deng et al. (2025) defines difficulty based on answer types, which fails to capture the model's intrinsic perception of difficulty. Other works (Team et al. 2025; Deng et al. 2025) employ fixed curricula without incorporating feedback from the model. Shi et al. (2025) estimate problem difficulty using expert models and propose an adaptive scheduler, however their method lacks historical data revisiting and does not address the degradation problem. In contrast, AdaCuRL dynamically schedules samples based on model feedback and incorporates historical data revisiting to prevent performance degradation on early data. Finally, through a designed KL loss computation, the model avoids Policy Degradation when learning signals are absent.

7 Conclusion

This work tackles the challenges of Gradient Starvation and Policy Degradation in GRPO training caused by random data sampling. We propose AdaCuRL, a curriculum RL approach that dynamically adjusts training difficulty based on the model's mastery of current samples. It also incorporates historical data replay and a meticulously designed KL divergence term to prevent reasoning deterioration. Without relying on external models or CoT datasets, AdaCuRL achieves significantly higher accuracy than random sampling on both multimodal and unimodal tasks using the same data. These results underscore the potential of curriculum learning in reasoning-oriented reinforcement learning.

References

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Chen, J.; Li, T.; Qin, J.; Lu, P.; Lin, L.; Chen, C.; and Liang, X. 2022. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. *arXiv preprint arXiv:2212.02746*.

Chen, J.; Tang, J.; Qin, J.; Liang, X.; Liu, L.; Xing, E. P.; and Lin, L. 2021. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*.

Chen, L.; Li, J.; Dong, X.; Zhang, P.; Zang, Y.; Chen, Z.; Duan, H.; Wang, J.; Qiao, Y.; Lin, D.; et al. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Chen, L.; Li, L.; Zhao, H.; Song, Y.; and Vinci. 2025. R1-V: Reinforcing Super Generalization Ability in Vision-Language Models with Less Than $3. https://github.com/Deep-Agent/R1-V. Accessed: 2025-02-02.

Chu, X.; Huang, H.; Zhang, X.; Wei, F.; and Wang, Y. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Dang, Q.-A.; and Ngo, C. 2025. Reinforcement Learning for Reasoning in Small LLMs: What Works and What Doesn't. arXiv:2503.16219.

Deng, H.; Zou, D.; Ma, R.; Luo, H.; Cao, Y.; and Kang, Y. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*.

Dong, Y.; Liu, Z.; Sun, H.-L.; Yang, J.; Hu, W.; Rao, Y.; and Liu, Z. 2025. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 9062–9072.

Graves, A.; Bellemare, M. G.; Menick, J.; Munos, R.; and Kavukcuoglu, K. 2017. Automated curriculum learning for neural networks. In *international conference on machine learning*, 1311–1320. Pmlr.

Guan, T.; Liu, F.; Wu, X.; Xian, R.; Li, Z.; Liu, X.; Wang, X.; Chen, L.; Huang, F.; Yacoob, Y.; et al. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14375–14385.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Gupta, H.; Verma, S.; Anantheswaran, U.; Scaria, K.; Parmar, M.; Mishra, S.; and Baral, C. 2024. Polymath: A challenging multi-modal mathematical reasoning benchmark. *arXiv preprint arXiv:2410.14702*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Huang, Q.; Chan, L.; Liu, J.; He, W.; Jiang, H.; Song, M.; Chen, J.; Yao, C.; and Song, J. 2025a. Boosting mllm reasoning with text-debiased hint-grpo. *arXiv preprint arXiv:2503.23905*.

Huang, W.; Jia, B.; Zhai, Z.; Cao, S.; Ye, Z.; Zhao, F.; Xu, Z.; Hu, Y.; and Lin, S. 2025b. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2901–2910.

Kazemi, M.; Alvari, H.; Anand, A.; Wu, J.; Chen, X.; and Soricut, R. 2023. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*.

Kembhavi, A.; Salvato, M.; Kolve, E.; Seo, M.; Hajishirzi, H.; and Farhadi, A. 2016. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 235–251. Springer.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35: 3843–3857.

Li, Z.; Wang, X.; Stengel-Eskin, E.; Kortylewski, A.; Ma, W.; Van Durme, B.; and Yuille, A. L. 2023. Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14963–14973.

Liang, Z.; Guo, K.; Liu, G.; Guo, T.; Zhou, Y.; Yang, T.; Jiao, J.; Pi, R.; Zhang, J.; and Zhang, X. 2024. Scemqa: A scientific college entrance level multimodal question answering benchmark. *arXiv preprint arXiv:2402.05138*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Lindström, A. D.; and Abraham, S. S. 2022. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Lu, P.; Gong, R.; Jiang, S.; Qiu, L.; Huang, S.; Liang, X.; and Zhu, S.-C. 2021a. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Lu, P.; Qiu, L.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; Rajpurohit, T.; Clark, P.; and Kalyan, A. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Lu, P.; Qiu, L.; Chen, J.; Xia, T.; Zhao, Y.; Zhang, W.; Yu, Z.; Liang, X.; and Zhu, S.-C. 2021b. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*.

Meng, F.; Du, L.; Liu, Z.; Zhou, Z.; Lu, Q.; Fu, D.; Shi, B.; Wang, W.; He, J.; Zhang, K.; et al. 2025. MM-Eureka: Exploring Visual Aha Moment with Rule-based Large-scale Reinforcement Learning. *arXiv preprint arXiv:2503.07365*.

Qiao, R.; Tan, Q.; Dong, G.; Wu, M.; Sun, C.; Song, X.; GongQue, Z.; Lei, S.; Wei, Z.; Zhang, M.; et al. 2024. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.

Saparov, A.; and He, H. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Seo, M.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1466–1476.

Shi, T.; Wu, Y.; Song, L.; Zhou, T.; and Zhao, J. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*.

Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Song, M.; Zheng, M.; Li, Z.; Yang, W.; Luo, X.; Pan, Y.; and Zhang, F. 2025. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models. *arXiv e-prints*, arXiv–2503.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Wang, K.; Pan, J.; Shi, W.; Lu, Z.; Ren, H.; Zhou, A.; Zhan, M.; and Li, H. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37: 95095–95169.

Wang, Y.; Chen, W.; Han, X.; Lin, X.; Zhao, H.; Liu, Y.; Zhai, B.; Yuan, J.; You, Q.; and Yang, H. 2024b. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*.

Wang, Y.; Xiong, F.; Wang, Y.; Li, L.; Chu, X.; and Zeng, D. D. 2025. Position bias mitigates position bias: mitigate position bias through inter-position knowledge distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 1495–1512.

Wang, Y.; Yue, Y.; Lu, R.; Liu, T.; Zhong, Z.; Song, S.; and Huang, G. 2023. Efficienttrain: Exploring generalized curriculum learning for training visual backbones. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5852–5864.

Xiao, Y.; Sun, E.; Liu, T.; and Wang, W. 2024. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*.

Xiong, F.; Xu, H.; Wang, Y.; Cheng, R.; Wang, Y.; and Chu, X. 2025. HS-STAR: Hierarchical Sampling for Self-Taught Reasoners via Difficulty Estimation and Budget Reallocation. *arXiv preprint arXiv:2505.19866*.

Xu, G.; Jin, P.; Hao, L.; Song, Y.; Sun, L.; and Yuan, L. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; et al. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Yang, Y.; He, X.; Pan, H.; Jiang, X.; Deng, Y.; Yang, X.; Lu, H.; Yin, D.; Rao, F.; Zhu, M.; et al. 2025. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Yu, W.; Yang, Z.; Li, L.; Wang, J.; Lin, K.; Liu, Z.; Wang, X.; and Wang, L. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9556–9567.

Zhang, R.; Jiang, D.; Zhang, Y.; Lin, H.; Guo, Z.; Qiu, P.; Zhou, A.; Lu, P.; Chang, K.-W.; Qiao, Y.; et al. 2024a. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, 169–186. Springer.

Zhang, R.; Zhang, B.; Li, Y.; Zhang, H.; Sun, Z.; Gan, Z.; Yang, Y.; Pang, R.; and Yang, Y. 2024b. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*.

Zhou, Y.; Yang, B.; Wong, D. F.; Wan, Y.; and Chao, L. S. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the association for computational linguistics*, 6934–6944.

Zou, C.; Guo, X.; Yang, R.; Zhang, J.; Hu, B.; and Zhang, H. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*.