Learning Like Humans: Advancing LLM Reasoning Capabilities via Adaptive Difficulty Curriculum Learning and Expert-Guided Self-Reformulation

Enci Zhang[1,2,3], Xingang Yan[1,2], Wei Lin[1,2*], Tianxiang Zhang[1,2], Qianchun Lu[1,2]

[1]State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen, China
[2]ZTE Corporation, Shenzhen, China
[3]Peking University, Shenzhen, China

E-mail: eczhang@stu.pku.edu.cn, weilincs@pku.org.cn

Abstract

Despite impressive progress in areas like mathematical reasoning, large language models still face challenges in consistently solving complex problems. Drawing inspiration from key human learning strategies, we propose two novel strategies to enhance the capability of large language models to solve these complex problems. First, Adaptive Difficulty Curriculum Learning (ADCL) is a novel curriculum learning strategy that tackles the *Difficulty Shift* phenomenon (i.e., a model's perception of problem difficulty dynamically changes during training) by periodically re-estimating difficulty within upcoming data batches to maintain alignment with the model's evolving capabilities. Second, Expert-Guided Self-Reformulation (EGSR) is a novel reinforcement learning strategy that bridges the gap between imitation learning and pure exploration by guiding models to reformulate expert solutions within their own conceptual framework, rather than relying on direct imitation, fostering deeper understanding and knowledge assimilation. Extensive experiments on challenging mathematical reasoning benchmarks, using Qwen2.5-7B as the base model, demonstrate that these human-inspired strategies synergistically and significantly enhance performance. Notably, their combined application improves performance over the standard Zero-RL baseline by 10% on the AIME24 benchmark and 16.6% on AIME25.

1 Introduction

The landscape of complex reasoning in large language models (LLMs) has been dramatically reshaped by recent breakthroughs, exemplified by models such as OpenAI-o1(OpenAI et al., 2024) and DeepSeek-R1(DeepSeek-AI et al., 2025). These models generate extensive Chains-of-Thought (CoT) (Wei et al., 2022) and exhibit self-reflection, particularly in mathematical problem solving. A pivotal training paradigm underpinning such advancements is Zero-RL (DeepSeek-AI et al., 2025; Liu et al., 2025; Zeng et al., 2025), which directly applies reinforcement learning (RL) to the base model. This paradigm leverages on-policy rollouts and rule-based rewards to elicit and enhance innate reasoning capabilities, often outperforming supervised fine-tuning (SFT) for complex tasks. Zero-RL's success underscores its potential to unlock deeper reasoning in LLMs.

Although the Zero-RL paradigm enhances complex reasoning, we focus on two main refinements: improving learning through strategic data arrangement, and extending model capabilities beyond the confines of on-policy exploration. **First**, although curriculum learning (CL) (Deng et al., 2025; Wen et al., 2025; Bengio et al., 2009) has been widely adopted to structure learning in an easier-to-harder progression using predefined difficulty metrics, it faces a critical limitation: the model's perception of difficulty is inherently dynamic and evolves during training. Applying static difficulty definitions directly results in curricula that misalign with the model's real-time learning requirements, ultimately leading to suboptimal training outcomes. **Second**, the challenge of expanding model capability boundaries presents another fundamental constraint in RL. Current approaches, including the Zero-RL paradigm, predominantly rely on on-policy methods that depend solely on self-generated rollouts. This creates an intrinsic limitation: the model's capacity for advancement becomes constrained by its pre-training knowledge base, as it lacks exposure to external reasoning patterns beyond its initial capabilities.

To address these challenges, we draw inspiration from well-established principles in cognitive science and educational psychology. First, human learning is most effective within the Zone of Proximal Development (ZPD) (Vygotsky and Cole, 1978), which describes the conceptual space

[*] Wei Lin is the corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6619–6633
November 4-9, 2025 ©2025 Association for Computational Linguistics

where tasks are challenging yet achievable with guidance. A static curriculum fails to remain within this zone as a model's competence evolves. This principle motivates our ADCL strategy, which dynamically adapts the training curriculum to consistently target the model's evolving ZPD, ensuring optimal learning conditions. Second, deep understanding of complex material is fostered not by passive absorption but through active processing, a phenomenon known as the Self-Explanation Effect (Chi et al., 1994). Learners who actively explain expert solutions to themselves—rephrasing logic and connecting it to their existing knowledge—assimilate information more effectively than those who simply memorize it. This insight underpins our EGSR strategy, which guides the model to self-reformulate expert solutions, promoting a genuine integration of new reasoning patterns rather than rote imitation.

Inspired by these human learning characteristics, we propose two novel training strategies to enhance the Zero-RL training paradigm: **(1) Adaptive Difficulty Curriculum Learning (ADCL)** addresses the challenge of dynamic difficulty perception by employing periodic difficulty re-estimation to adjust the curriculum based on the model's difficulty perception. **(2) Expert-Guided Self-Reformulation (EGSR)** tackles the limitations of on-policy exploration by enabling the model to learn from high-quality trajectories from expert policies. This is achieved not through mere imitation, but by guiding the model to actively reconstruct and internalize expert solutions, thereby fostering the development of new reasoning capabilities beyond its initial scope.

This paper makes the following main contributions:

- **Conceptual and Empirical Insights:** We identify the *Difficulty Shift* phenomenon, which undermines static curricula, and show that Expert-Guided Self-Reformulation enables more stable, on-policy knowledge integration than direct imitation.

- **Novel Strategies:** ADCL dynamically adjusts training curricula by periodically re-estimating difficulty within upcoming data batches to align with evolving model capabilities. EGSR enables knowledge assimilation by guiding models to reconstruct expert solutions within their own conceptual frameworks, rather than direct imitation.

- **Comprehensive Experiments:** Our experiments confirm the efficacy of our proposed ADCL and EGSR strategies, which synergistically enhance mathematical reasoning within the Zero-RL baseline. Notably, their combined application improves performance over the standard Zero-RL baseline by 10% on the AIME24 benchmark and 16.6% on AIME25.

2 Related Work

2.1 Curriculum Learning for LLMs

CL is a training strategy that mimics human learning progression by systematically increasing the complexity of training data, typically following an easier-to-harder trajectory. This principle has demonstrated significant efficacy across various machine learning domains from computer vision (CV) (Guo et al., 2018; Jiang et al., 2014) and natural language processing (NLP) (Platanios et al., 2019; Tay et al., 2019) to reinforcement learning (RL), and notably in both the pre-training (Zhang et al., 2025) and post-training (Wang et al., 2025; Shi et al., 2025) phases of LLMs.

Specifically for reinforcement fine-tuning (RFT) of LLMs, CL strategies are increasingly being adopted to optimize the training process and improve model performance (Naïr et al., 2024; Team et al., 2025; Deng et al., 2025). Many current CL applications in this context rely on static curricula, where task difficulty is predetermined offline, and data is curated accordingly(Lee et al., 2024; Team et al., 2025). Although such predefined curricula have demonstrated effectiveness, they may not dynamically adapt to the model's evolving learning state. To address this limitation of static curricula, we propose the ADCL strategy, which dynamically adjusts training curricula by periodically re-estimating difficulty within upcoming data batches to align with evolving model capabilities.

2.2 Reinforcement Learning for LLMs

Reinforcement learning methods are broadly categorized based on data utilization into on-policy and off-policy techniques. On-policy algorithms, such as PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015), learn from data generated by the current policy, ensuring stability but often requiring extensive data. In contrast, off-policy techniques, including DQN (Mnih et al., 2013) and SAC (Haarnoja et al., 2018), utilize data from various policies, enhancing data efficiency but poten-

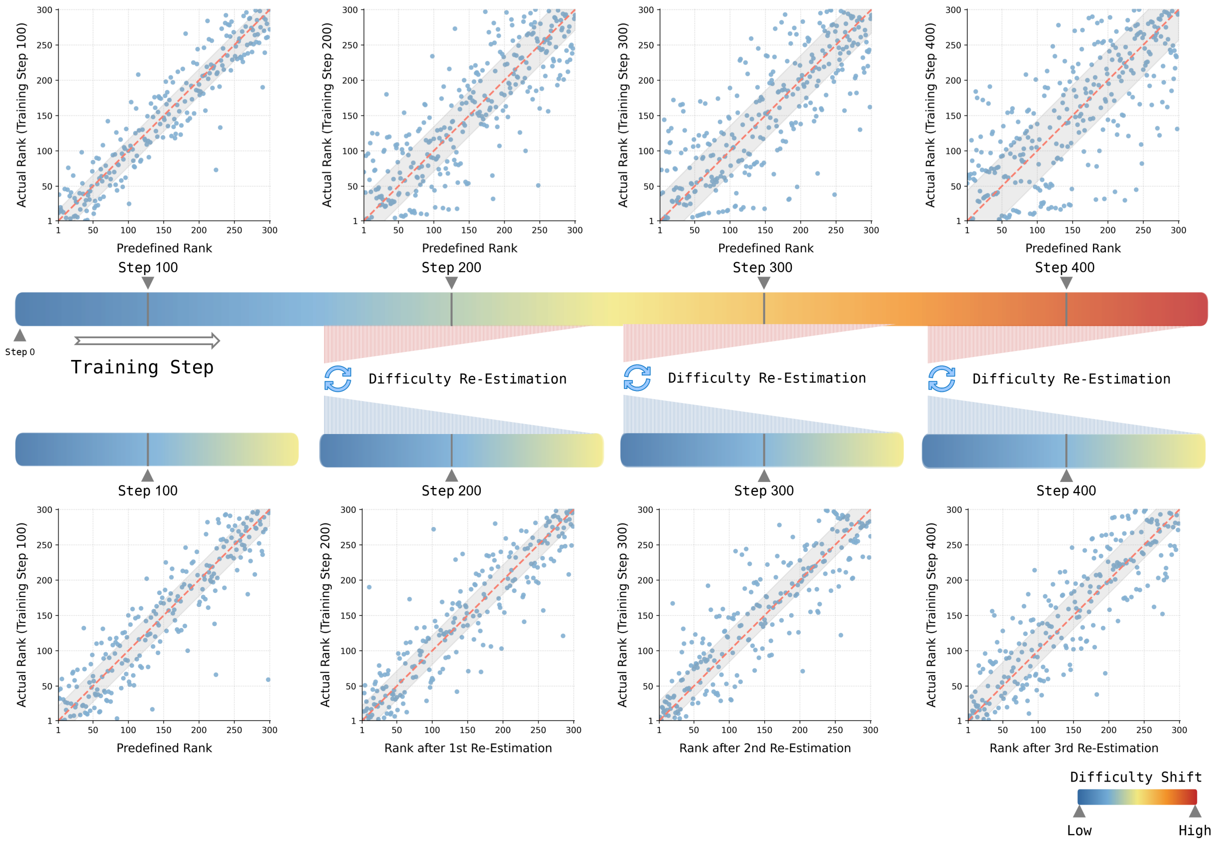

Figure 1: Illustration of the *Difficulty Shift* phenomenon and the ADCL counteraction. Top Row: Scatter plots show the increasing divergence between the initial Predefined Rank (x-axis) and the model's evolving Actual Rank (y-axis) when using a predefined curriculum, demonstrating the growing *Difficulty Shift* over training steps. Bottom Row: ADCL employs periodic difficulty re-estimation based on the current model state to re-sort the upcoming batch. This dynamic adjustment results in the model's Actual Ranks (y-axis) more closely aligning with the batch's re-estimated rank order (x-axis), correcting the *Difficulty Shift* observed with a predefined curriculum.

tially introducing instability and complexity.

In the context of post-training phases, on-policy methods such as PPO and GRPO (Team, 2024) have become the de facto standard. While off-policy approaches such as DPO (Rafailov et al., 2023) are also explored, a key challenge is integrating the benefits of rich off-policy data with on-policy stability. Direct integration often destabilizes training due to significant distributional differences between the learning policy and the expert policy. We propose the EGSR strategy to address this by using expert demonstrations to guide the current policy in generating improved, more on-policy trajectories.

3 Method

3.1 Adaptive Difficulty Curriculum Learning

CL is a training strategy inspired by human cognition, where models learn progressively from easier to harder examples. A critical challenge is the model's dynamic difficulty perception during training (a phenomenon we term *Difficulty Shift*). This dynamic shift can render the initial, fixed difficulty ranking(Deng et al., 2025; Wen et al., 2025) increasingly inaccurate, misaligning the presented curriculum with the model's real-time learning needs and potentially hindering training progression.

We provide empirical evidence of the *Difficulty Shift* phenomenon in Figure 1. The color-coded progress bar (left-to-right) illustrates the standard GRPO algorithm training on a dataset. The top row illustrates training using a predefined CL approach with a predefined difficulty ordering established by the base model. To analyze the shift, we extracted several model checkpoints during this predefined curriculum training. For each checkpoint, we assessed its current perception of difficulty by evaluating it on the immediately following 300 data points and ranking them based on the accuracy achieved by that specific checkpoint. The resulting actual difficulty ranks (y-axis) are plotted against their original predefined ranks (x-axis, monotonically incrementing from 1 to 300) in scatter plots corresponding to each checkpoint marker, with a gray shaded region highlighting the 25th-75th percentile deviation between the actual (y) and predefined (x) ranks. As observed in the top progression of Figure 1, training with the predefined curriculum leads to progressively increasing dispersion in the scatter plots and a widening of this quantile range, both indicating a larger *Difficulty Shift* over time. We attribute this increase in *Difficulty Shift* to the increased deviation of the model from the initial model during the training process.

To counteract this detrimental *Difficulty Shift* while maintaining computational tractability, we propose Adaptive Difficulty Curriculum Learning (ADCL), a dynamic difficulty-aware training strategy. Instead of relying on a fixed, predefined order or re-evaluating the entire dataset like standard Self-Paced Learning (SPL)(Kumar et al., 2010), ADCL leverages dynamic difficulty re-estimation based on the model's current state to re-sort upcoming data batches. This mirrors how humans dynamically adjust their learning paths based on perceived task difficulty, rather than rigidly following a fixed sequence.

The core mechanism of our ADCL algorithm is the dynamic re-estimation and re-sorting of upcoming data batches based on the model's evolving state. Specifically, the process begins by estimating difficulty scores $\delta_0(x_i)$ for all samples in dataset $\mathcal{D}$ using the initial model parameters θ_0. After sorting $\mathcal{D}$ according to these scores, it is partitioned into K sequential batches $B_1, B_2, ..., B_K$. These batches are then processed iteratively. At iteration k, the model parameters θ_{k-1} are updated to θ_k by training on batch B_k. Following this, ADCL re-evaluates the difficulty scores $\delta_k(x_i)$ for elements within the subsequent batch B_{k+1} using the updated model θ_k. This batch is then internally re-sorted according to the new difficulty estimation before proceeding to the next iteration. This localized, progressive re-sorting continues until all the training is completed. The detailed pseudocode for ADCL is provided in Appendix A.

As illustrated in the bottom row of Figure 1, ADCL dynamically re-estimates the difficulty of upcoming batches multiple times during training, significantly correcting the difficulty deviation compared to predefined CL.

3.2 Expert-Guided Self-Reformulation

RL for LLMs, particularly in reasoning tasks, often functions as a process to enhance sample efficiency. Although RL can improve accuracy on problems the model can occasionally solve, it fundamentally struggles to elicit solutions for problems entirely outside the model's current capabilities. This limitation implies that the base model's inherent knowledge restricts the upper bound of performance achievable through standard on-policy RL techniques(Yue et al., 2025).

The standard objective for certain on-policy RL algorithms, such as GRPO detailed in Eq.1, aims to optimize the policy π_θ using G trajectories $\{\tau_i\}_{i=1}^{G}$ sampled from a previous policy $\pi_{\theta_{\text{old}}}$ for a given question q. However, a frequent challenge in practical training scenarios with such algorithms is the "zero-reward" situation. This occurs when all G rollouts from $\pi_{\theta_{\text{old}}}$ yield a reward of zero, i.e., $R(\tau_i) = 0$ for all i. This outcome primarily stems from problem complexity exceeding the model's current capabilities. In such zero-reward scenarios, advantage estimates $\hat{A}_{i,t}$ across all actions effectively vanish, resulting in null gradient updates that fail to contribute to policy improvement. This learning impasse highlights the need to incorporate external guidance from an off-policy expert policy π_ϕ. Such guidance, often provided through demonstrations generated by this policy, can introduce meaningful learning signals and potentially infuse the model with knowledge beyond its current capabilities.

One straightforward way to leverage guidance from this expert policy is to replace a portion of on-policy trajectories with M expert demonstrations(Hester et al., 2018; Liu et al., 2022), creating a hybrid trajectory set $\mathcal{T}_{\text{mixed}} = \mathcal{T}_{\text{on}} \cup \mathcal{T}_{\text{off}}$, where $\mathcal{T}_{\text{on}} = \{\tau_i \mid \tau_i \sim \pi_{\theta_{\text{old}}}(\cdot|q), i = 1, ..., G - M\}$ and $\mathcal{T}_{\text{off}} = \{\tau_j \mid \tau_j \sim \pi_\phi(\cdot|q), j = 1, ...M\}$. When incorporating off-policy demonstrations, importance sampling must be applied to correct the distribution shift, modifying the probability ratio in the GRPO objective as:

$$r_{i,t}(\theta) = \begin{cases} \frac{\pi_\theta(\tau_{i,t}|q,\tau_{i,<t})}{\pi_{\theta_{\text{old}}}(\tau_{i,t}|q,\tau_{i,<t})}, & \text{if } \tau_i \in \mathcal{T}_{\text{on}} \\ \frac{\pi_\theta(\tau_{i,t}|q,\tau_{i,<t})}{\pi_\phi(\tau_{i,t}|q,\tau_{i,<t})}, & \text{if } \tau_i \in \mathcal{T}_{\text{off}} \end{cases} \quad (3)$$

And the advantage estimates are computed as:

$$A_i = \frac{R(\tau_i) - \text{mean}(R(\tau_{\text{mixed}}))}{\text{std}(R(\tau_{\text{mixed}}))} \quad (4)$$

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) &= \mathbb{E}[q \sim P(Q), \{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot|q)]\\ &\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\left\{\min\left[r_{i,t}(\theta)A_i, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)A_i\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta \| \pi_{ref}\right]\right\}\\ &\text{where}\quad r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,t}|q,\tau_{i,<t})}{\pi_{\theta_{\text{old}}}(\tau_{i,t}|q,\tau_{i,<t})} \quad \text{and} \quad A_i = \frac{R(\tau_i) - \text{mean}(\{R(\tau_k)\}_{k=1}^{G})}{\text{std}(\{R(\tau_k)\}_{k=1}^{G})}\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{J}_{\text{GRPO-EGSR}}(\theta) &= \mathbb{E}[q, g \sim P(Q), \{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot|q), \{\tau_i^{'}\}_{i=1}^{M} \sim \pi_{\theta_{old}}(\cdot|q,g)]\\ &\frac{1}{G-M}\sum_{i=1}^{G-M}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\left\{\min\left[r_{i,t}(\theta)A_i, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)A_i\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta \| \pi_{ref}\right]\right\}\\ &+\frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\tau_i^{'}|}\sum_{t=1}^{|\tau_i^{'}|}\left\{\min\left[r_{i,t}^{'}(\theta)A_i, \text{clip}\left(r_{i,t}^{'}(\theta), 1-\epsilon, 1+\epsilon\right)A_i\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta \| \pi_{ref}\right]\right\}\\ &\text{where}\quad r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,t}|q,\tau_{i,<t})}{\pi_{\theta_{\text{old}}}(\tau_{i,t}|q,\tau_{i,<t})} \quad , \quad r_{i,t}^{'}(\theta) = \frac{\pi_\theta(\tau_{i,t}^{'}|q,\tau_{i,<t}^{'})}{\pi_{\theta_{\text{old}}}(\tau_{i,t}^{'}|q,g,\tau_{i,<t}^{'})}\\ &\text{and}\quad A_i = \frac{R(\tau_i) - \text{mean}(\{R(\tau_k)\}_{k=1}^{G-M} + \{R(\tau_k^{'})\}_{k=1}^{M})}{\text{std}(\{R(\tau_k)\}_{k=1}^{G-M} + \{R(\tau_k^{'})\}_{k=1}^{M})}\end{aligned} \tag{2}$$

While this approach is theoretically sound, in practice the expert policy π_ϕ is typically inaccessible or uses incompatible tokenization schemes, making the probability ratio calculation infeasible. Even when π_ϕ is available, the substantial distributional differences between policies often produce extreme importance weights with destabilizing variance. Our experiments in Section 4.2 demonstrate that incorrectly treating expert demonstrations as on-policy samples may degrade performance on complex reasoning tasks.

We argue that the core issue lies in the distributional mismatch between the reference solution trajectories generated by an external expert policy π_ϕ and the trajectories naturally produced by the current policy $\pi_{\theta_{\text{old}}}$. Inspired by the human learning analogy of restating solutions, we propose a method that circumvents the need to estimate π_ϕ or handle large importance weights. Instead of directly incorporating the off-policy trajectory $\tau_j \sim \pi_\phi$, we use expert demonstrations to guide $\pi_{\theta_{\text{old}}}$ in generating more effective trajectories $\tau_i^{'} \sim \pi_{\theta_{\text{old}}}(\cdot|q,g)$, where g represents the guidance information. Similar to how students learn by rephrasing solutions in their own words, this approach produces trajectories that are naturally aligned with the model's current capabilities while incorporating expert knowledge. This guided generation creates samples that remain fundamentally "on-policy" while bridging the gap toward expert performance.

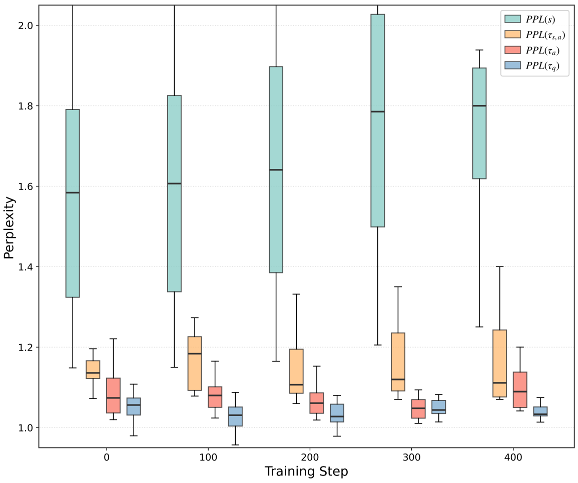

Figure 2: Perplexity (PPL) of four trajectory types under the current model policy π_θ across training checkpoints: unguided model generations $\boldsymbol{\tau_q}$ (blue), reference expert solutions $\boldsymbol{s}$ (green), model-generated trajectories guided by (s,a) $\boldsymbol{\tau_{s,a}}$ (orange), and those guided by a alone $\boldsymbol{\tau_a}$ (red). Lower PPL indicates better alignment with π_θ.

To verify our hypothesis, we experimentally measured the model's perplexity (PPL) on four distinct trajectory types across multiple training checkpoints, using 300 problems defined by (q, s, a) triples (question, expert solution, answer); these results are shown in Figure 2. We evaluated PPL for: (1) unguided model generations $\boldsymbol{\tau_q}$, (2) reference expert solutions $\boldsymbol{s}$, (3) model-generated trajectories $\boldsymbol{\tau_{s,a}}$ guided by both s and a, and (4) trajectories $\boldsymbol{\tau_a}$ guided by a alone. Throughout training, the PPL of reference expert solutions s consistently remained the highest, while the PPL of the model's unguided outputs τ_q was the lowest, serving as a baseline for its natural generative distribution. Crucially, the PPL of $\tau_{s,a}$ and τ_a was found to be substantially lower than $PPL(s)$ and markedly closer to $PPL(\tau_q)$. These PPL results indicate that guiding the model to generate solutions yields trajectories significantly more aligned with its current policy (i.e., more 'on-policy') than directly using off-policy expert solutions.

Building on our PPL analysis (Figure 2), we propose Expert-Guided Self-Reformulation (EGSR), a novel reinforcement learning training strategy designed to enhance on-policy reinforcement learning algorithms, like GRPO. The EGSR training objective, presented in Equation.2, integrates M expert-guided trajectories $\tau_i' \sim \pi_{\theta_{old}}(\cdot|q, g)$ (where g is guidance derived from expert policy π_ϕ) with $G - M$ standard rollouts, particularly when initial exploration yields zero rewards. For these guided trajectories, the ratio $r'_{i,t}(\theta)$ is calculated with respect to their specific generation policy $\pi_{\theta_{old}}(\cdot|q, g)$. This is justified as our PPL findings indicate that trajectories from $\pi_{\theta_{old}}(\cdot|q, g)$ are distributionally closer to the model's unguided outputs than trajectories from the original expert policy π_ϕ, thereby promoting more stable, near-on-policy learning while still leveraging expert knowledge. The detailed pseudocode for EGSR is provided in Appendix B.

4 Experiments

4.1 Setup

(1) Datasets: Our datasets are curated from high-quality reasoning corpora including S1 (Muennighoff et al., 2025) and DeepScaleR (Luo et al., 2025). We applied filtering criteria to ensure solutions are verifiable by Math-Verify[1] and problems require computational rather than proof-based solutions. We estimated problem difficulty using Qwen2.5-7B(Yang et al., 2024) as our base model, performing 32 rollouts per problem to calculate accuracy rates. From this evaluation, we created a collection of 6,894 problems with accuracy rates between 10%-90%, termed **BaseSet-7K**. To provide a more extensive set of instances for our EGSR method to leverage for guidance, we supplemented BaseSet-7K by incorporating an additional 3,000 problems where the base model consistently achieved 0% accuracy. This formed the augmented collection termed **AugSet-10K**.

(2) Benchmarks: To evaluate our models' mathematical reasoning capabilities, we use five established benchmarks: MATH500(Lightman et al.), AIME24[2], AIME25[3], AMC23[4], and Minerva-math(Lewkowycz et al., 2022). We report pass@8 for AIME24 and AIME25 to mitigate the high variance that would result from pass@1 on these smaller benchmark sets, and use standard pass@1 for all other benchmarks.

(3) Settings: We implemented our training framework using TRL(von Werra et al., 2020) with Qwen2.5-7B as the base model. For GRPO, we used 8 rollouts per problem, a global batch size of 1024, and a fixed learning rate of 1×10^{-6}. Our reward function, drawing inspiration from (Team, 2024), is a composite reward: $R(\tau) = \lambda_1 \cdot R_{\text{format}}(\tau) + \lambda_2 \cdot R_{\text{accuracy}}(\tau)$, where

$$R_{\text{format}}(\tau) = \begin{cases} 1 & \text{if } \tau \text{ follows the output format} \\ 0 & \text{otherwise} \end{cases}$$

and

$$R_{\text{accuracy}}(\tau) = \begin{cases} 1 & \text{if } \tau \text{ contains correct answer} \\ 0 & \text{otherwise} \end{cases}$$

, we set $\lambda_1 = 1.0$ and $\lambda_2 = 2.0$.

For the ADCL strategy, we utilized the BaseSet-7K dataset, and our implementation divided the curriculum into four difficulty-based batches and performed three difficulty re-estimations.

For the EGSR strategy, we utilized the AugSet-10K dataset, and explored two guidance strategies: using only the expert's final answer a (EGSR(a)) or using both the answer a and the solution process s (EGSR(s, a)). Considering that GRPO's learning effectiveness is maximized when the success rate is around 50% (Bae et al., 2025), we thus generated 4 guided trajectories in such instances. Complete hyperparameters and training results are provided in Appendix C.

[1] https://github.com/huggingface/Math-Verify

[2] https://huggingface.co/datasets/hendrydong/aime24

[3] https://huggingface.co/datasets/TIGER-Lab/AIME25

[4] https://huggingface.co/datasets/zwhe99/amc23

Model	MATH	AIME24	AIME25	AMC23	Minervamath
Qwen2.5-7B	69.40	16.67	16.67	32.50	15.44
+ SFT	70.80	13.33	13.33	35.00	16.91
+ RL	72.40	26.67	16.67	45.00	18.01
+ PCL	75.40	30.00	23.33	50.00	22.42
+ ADCL	76.20	$\underline{33.33}$	$\underline{30.00}$	$\underline{55.00}$	22.78
+ off-policy	66.00	23.33	16.67	30.00	18.65
+ EGSR(a)	72.60	30.00	20.00	50.00	20.96
+ EGSR(s, a)	$\underline{79.40}$	$\underline{33.33}$	$\underline{30.00}$	**57.50**	$\underline{24.63}$
+ ADCL & EGSR	**81.80**	**36.67**	**33.33**	$\underline{55.00}$	**25.74**

Table 1: Performance of Qwen2.5-7B with various training strategies on mathematical reasoning benchmarks, `+SFT`: Supervised Fine-Tuning, `+RL`: Standard GRPO algorithm, `+PCL`: GRPO with Predefined Curriculum Learning, `+ADCL`: GRPO with our Adaptive Difficulty Curriculum Learning, `+off-policy`: GRPO with direct replacement of rollouts by expert solutions, `+EGSR(`a`)`: Our ESGR method using only expert answer (a) for guidance, `+EGSR(`s, a`)`: Our ESGR method using expert solution (s) and answer (a) for guidance, `+ADCL&EGSR`: Combination of our ADCL and EGSR(s, a) methods. Bold and underline represent the 1st and 2nd in performance.

4.2 Main Results

Our main results are presented in Table 1. Initially, we observed that while SFT (+SFT) showed slight improvements on some benchmarks, its performance on more complex benchmarks such as AIME24 and AIME25 actually decreased compared to the base model. In contrast, the standard RL (+RL) method brought more significant and stable performance improvements on most benchmarks. These findings indicate that RL adapts to complex tasks better than SFT.

Comparing CL strategies, our proposed Adaptive Difficulty Curriculum Learning (+ADCL) consistently outperforms Predefined Curriculum Learning (+PCL). This advantage stems from ADCL's curriculum design, which ensures more effective alignment with the model's evolving state.

Turning to expert guidance, the naive incorporation of expert solutions via direct off-policy guidance (+off-policy) led to a degradation in performance compared to the +RL baseline. In contrast, our EGSR strategies, both EGSR(a) and EGSR(s,a), surpassed this naive approach, with EGSR(s,a) achieving the most favorable results among the guidance methods. While EGSR(a), which uses only the final expert answer for guidance, we observed instances where it could lead the model to generate trajectories that reach the correct answer via a flawed or incomplete reasoning process, even though such trajectories might exhibit lower perplexity. An example illustrating such a scenario, where the model correctly predicts the answer but with an erroneous solution when guided by the answer alone, is provided in Appendix F. EGSR(s,a), by incorporating guidance from both the expert's solution process and the final answer, encourages the model to reformulate a more complete and coherent reasoning path within its own conceptual framework, leading to more effective improvements in reasoning capabilities.

Furthermore, our two proposed training strategies, ADCL and EGSR, can be synergistically combined for even better performance. When used together (+ADCL & EGSR), these strategies show substantial gains over the standard RL approach. Specifically, the combined approach achieved a 10% improvement on AIME24, a 16.66% improvement on AIME25, and a 7.73% improvement on Minervamath compared to the +RL baseline.

5 Analysis

5.1 Generalization to Other Architectures

To assess the generalizability of our proposed strategies, we conducted additional experiments on models with different architectures and scales: Llama3.1-8B-Instruct(Grattafiori et al., 2024) and Qwen2.5-1.5B(Yang et al., 2024). As shown in Table 2, the combination of ADCL and EGSR consistently and significantly outperforms the RL baseline across both models. These results confirm that the benefits of our methods are not specific to a single model family but are broadly applicable, highlighting their robustness.

Model	MATH	AIME24	AIME25	AMC23
Qwen2.5-1.5B	32.60	10.00	3.33	22.50
+SFT	35.80	10.00	0.00	27.50
+RL	59.00	16.67	6.67	35.00
+PCL	61.20	16.67	10.00	35.00
+ADCL	67.00	16.67	10.00	40.00
+off-policy	30.80	6.67	0.00	7.50
+EGSR	64.60	**26.67**	16.67	45.00
+ADCL & EGSR	**69.80**	23.33	**20.00**	**47.50**

Model	MATH	AIME24	AIME25	AMC23
Llama3.1-8B	46.20	13.33	3.33	27.50
+SFT	44.00	6.67	6.67	22.50
+RL	53.20	13.33	6.67	32.50
+PCL	54.60	20.00	10.00	35.00
+ADCL	56.40	**23.33**	10.00	40.00
+off-policy	32.00	13.33	6.67	15.00
+EGSR	60.40	20.00	**13.33**	40.00
+ADCL & EGSR	**60.60**	**23.33**	**13.33**	**42.50**

Table 2: Performance on Qwen2.5-1.5B and Llama3.1-8B-Instruct, using the same experimental setup as in Table 1. Bold and underline represent the 1st and 2nd in performance.

5.2 ADCL Training Dynamic

Figure 3 illustrates the ADCL training dynamics under different strategies. Standard RL without curriculum (NoCL) shows a steady increase in accuracy reward. In contrast, curriculum-based strategies (ADCL and PCL), despite showing a decrease in accuracy reward later in training due to increased problem difficulty, consistently maintain higher rewards compared to the baseline raw accuracy at corresponding difficulty levels. This confirms that learning is occurring and the reward decrease is primarily driven by the curriculum's progression to harder tasks. The key advantage of our proposed ADCL over PCL lies in its adaptive nature. PCL's reward curve declines relatively smoothly because its static difficulty ranking suffers from *Difficulty Shift*; training batches inevitably mix samples of varying actual difficulty relative to the model's current state, averaging out the perceived challenge. ADCL, however, employs periodic difficulty re-estimation and batch re-sorting based on the current model's perception. Appendix D quantifies this re-sorting against the *Difficulty Shift*. This leads to a steeper reward curve, particularly after re-estimation points, because ADCL immediately presents the model with a batch ordered according to its updated sense of "easy-to-hard".

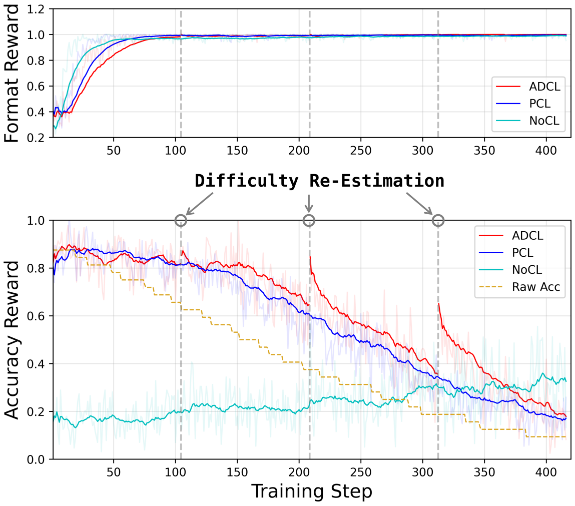

Figure 3: Comparison of reward dynamics under different training strategies. The curves represent Adaptive Difficulty Curriculum Learning (ADCL, red), Predefined Curriculum Learning (PCL, blue), and No Curriculum learning (NoCL, cyan). The dashed yellow line (Raw Acc) indicates the basemodel's performance on curriculum-ordered data.

Model	MATH	AIME25	AMC23	Minervamath
Qwen2.5-7B	89.40	30.00	87.50	49.63
+ SFT	90.20	23.33	90.00	50.36
+ RL	88.60	26.67	85.00	49.26
+ off-policy	93.20	40.00	**92.50**	51.47
+ EGSR(s, a)	**93.40**	**46.67**	**92.50**	**51.84**

Table 3: Model performance comparison using pass@32 metric, which better reflects capability boundaries.

5.3 Impact of Expert Guidance on Capability Boundaries

To further investigate whether expert guidance genuinely expands a model's intrinsic problem-solving capabilities, we evaluated various methods using a pass@32 metric. Compared to pass@1 or pass@8, pass@32 allows the model substantially more opportunities to explore diverse solution paths, offering a clearer view of its underlying capability boundaries. The results presented in Table 3 reveal that standard RL, while effective at optimizing for common success, did not expand and even slightly contracted the model's capability boundary compared to the base model. SFT provided a marginal improvement. In contrast, methods incorporating expert guidance demonstrated more significant expansions of these boundaries. Notably, our EGSR(s,a) strategy achieved the largest improvement, increasing the pass@32 accuracy on AIME25 by a substantial 16.67% over the base model, suggesting that it fosters deeper knowledge assimilation rather than merely refining existing skills.

6 Conclusion

Drawing inspiration from human learning strategies, this work addresses key challenges in enhancing the capabilities of large language models to solve complex tasks. We first observed the *Difficulty Shift* phenomenon, where a model's perception of problem difficulty changes dynamically during training, hindering static curriculum learning. To counteract this, we propose Adaptive Difficulty Curriculum Learning (ADCL), which periodically re-estimates difficulty to maintain an aligned learning path. Secondly, recognizing that existing methods often fail to help models assimilate knowledge beyond their initial capabilities, we introduce Expert-Guided Self-Reformulation (EGSR). EGSR guides models to actively reformulate expert solutions within their own conceptual framework, rather than relying on direct imitation, fostering deeper knowledge assimilation. Experiments validate our proposed ADCL and EGSR, showing they significantly outperform baselines, especially in combination. These findings highlight the value of incorporating adaptive human-like learning mechanisms into LLM training.

Limitations

While we identify and address the *Difficulty Shift* phenomenon within our RL framework, a broader investigation into its prevalence and characteristics across different training paradigms, such as Supervised Fine-Tuning (SFT), and various domains would offer a more comprehensive understanding. The current implementation of ADCL employs a fixed number of difficulty re-estimations and re-sortings; an exploration of varying these frequencies could further illuminate its impact on correcting the *Difficulty Shift* and optimizing performance. The efficacy of EGSR is closely tied to the quality of expert demonstrations, and while our study assumes access to high-quality demonstrations, the precise impact of imperfections, errors, or biases within these demonstrations on the model's learning process warrants further investigation. Our findings are primarily demonstrated using a specific model architecture, size, and datasets focused on mathematical reasoning; further studies would be beneficial to ascertain the generalizability of our proposed ADCL and EGSR strategies across a wider range of model types, scales, and diverse application domains.

References

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *Preprint*, arXiv:2504.03380.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Michelene TH Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive science*, 18(3):439–477.

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

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *Preprint*, arXiv:2503.07065.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John

Quan, Andrew Sendonaris, Ian Osband, and 1 others. 2018. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 547–556.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1281–1309, Mexico City, Mexico. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Haochen Liu, Zhiyu Huang, Jingda Wu, and Chen Lv. 2022. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. In *2022 IEEE intelligent vehicles symposium (IV)*, pages 921–928. IEEE.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset`. Notion Blog.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *Preprint*, arXiv:1312.5602.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Marwa Naïr, Kamel Yamani, Lynda Lhadj, and Riyadh Baghdadi. 2024. Curriculum learning for small code language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 390–401, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL-HLT (1)*, pages 1162–1172. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France. PMLR.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *Preprint*, arXiv:2504.05520.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Annual Meeting of the Association for Computational Linguistics*.

DeepSeek Team. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in large language models. *arXiv preprint arXiv:2402.03300*.

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, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

Zhenting Wang, Guofeng Cui, Kun Wan, and Wentian Zhao. 2025. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *Preprint*, arXiv:2504.09710.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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. 2025. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *Preprint*, arXiv:2503.10460.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Preprint*, arXiv:2504.13837.

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.

Xuemiao Zhang, Liangyu Xu, Feiyu Duan, Yongwei Zhou, Sirui Wang, Rongxiang Weng, Jingang Wang, and Xunliang Cai. 2025. Preference curriculum: Llms should always be pretrained on their preferred data. *Preprint*, arXiv:2501.13126.

A Pseudocode for ADCL

Algorithm 1 Adaptive Difficulty Curriculum Learning

```
Input: Dataset $\mathcal{D} = \{x_i\}_{i=1}^{M}$; initial policy model $\pi_{\theta_0}$; number of batches $K$;.

$\forall x_i \in \mathcal{D}$, compute difficulty score $\delta_0(x_i) = f(\pi_{\theta_0}, x_i)$    ▷ Initial difficulty estimation
$\mathcal{D}_{sorted} \leftarrow$ Sort($\mathcal{D}, \delta_0$)    ▷ Sort based on increasing difficulty
Partition $\mathcal{D}_{sorted}$ into $\{B_1, B_2, \ldots, B_K\}$ where $|B_k| = \lfloor M/K \rfloor$ or $\lceil M/K \rceil$
for $k = 1$ to $K$ do
    $\theta_k \leftarrow$ TrainModel($\theta_{k-1}, B_k$)    ▷ Update $\pi_\theta$ using RL algorithm (e.g., GRPO)
    if $k < K$ then
        $\forall x_i \in B_{k+1}$, compute $\delta_k(x_i) = f(\pi_{\theta_k}, x_i)$    ▷ Re-estimate difficulty
        $B_{k+1} \leftarrow$ Sort($B_{k+1}, \delta_k$)    ▷ Re-sort next batch
Output: $\pi_{\theta_K}$.
```

B Pseudocode for EGSR

Algorithm 2 Expert-Guided Self-Reformulation

```
Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models $r_\phi$; Dataset $\mathcal{D} = \{(q_i, s_i, a_i)\}_{i=1}^{N}$, where $q_i$ is the question, $s_i$ is the solution, and $a_i$ is the answer;
policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, ..., I do
    for step = 1, ..., M do
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample $G$ trajectories $\{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{R(\tau_i)\}_{i=1}^{G}$ for each sampled trajectory $\tau_i$ using $r_\phi$
        if $\sum_{i=1}^{G} R(\tau_i) = 0$ then    ▷ All rewards are zero
            Generate guided trajectories $\{\tau_i^{'}\}_{i=1}^{M} \sim \pi_{\theta_{old}}(\cdot \mid f(q, s, a))$
            Create mixed trajectory $\tau_{\text{mixed}} = \{\tau_k\}_{k=1}^{G-M} \cup \{\tau_k^{'}\}_{k=1}^{M}$
            Recompute rewards for $\tau_{\text{mixed}}$
        Compute advantages $A_{i,t}$ for each token $t$ in each trajectory $\tau_i$ from $\mathcal{T}_{\text{mixed}}$
        for GRPO iteration = 1, ..., $\mu$ do
            Update $\pi_\theta$ by maximizing the GRPO-EGSR objective in Eq. 2
Output $\pi_\theta$
```

C Training Hyperparameters

The complete hyperparameters used during training are shown in Table 4.

Training Setup	
global_batch_size	1024
per_device_batch_size	8
num_machines	4
num_devices	32 (4 × 8 H20s)
Rollout Configuration	
rollout_backend	vllm
tensor_parallel_size	1
data_parallel_size	8
num_generation	8
max_completion_length	4096
temperature	0.7
GRPO Training Configuration	
learning rate	1e-6 (constant)
beta	0
epsilon	0.2
reward functions	format, accuracy
reward weights	1.0, 2.0
num_iteration	4
gradient_accumulation_steps	4
SFT Training Configuration	
learning rate	2e-5 (decay)
max_length	8192
gradient_accumulation_steps	1

Table 4: Training Hyperparameters

Detailed training curves can be found in Figure 4, Figure 5, and Figure 6.

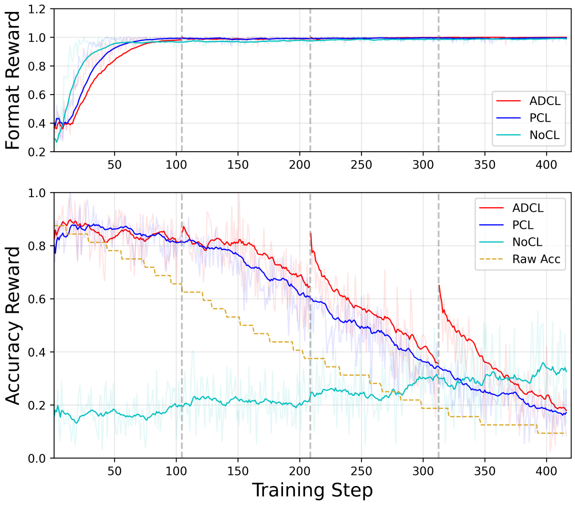

Figure 4: Reward curves from training the ADCL strategy and relevant baseline methods

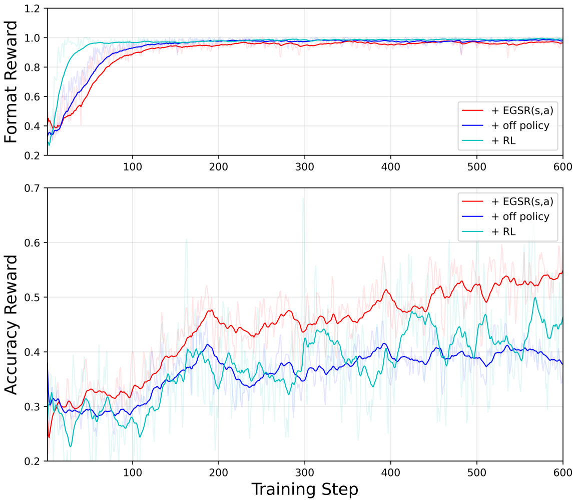

Figure 5: Reward curves from training the EGSR strategy and relevant baseline methods

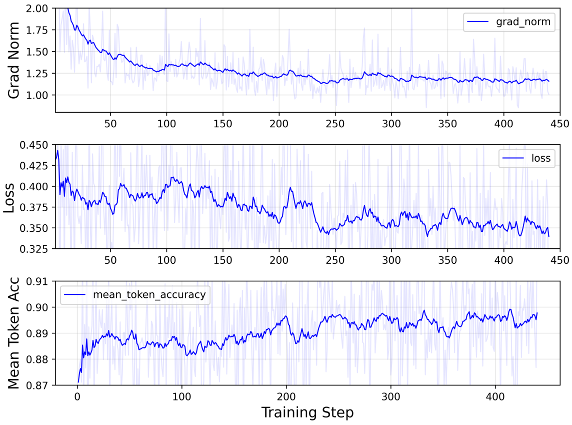

Figure 6: Supervised Fine-Tuning (SFT) training curve

D Quantifying ADCL Re-sorting Dynamics

Curriculum Batch	NIR
Batch 1 (No Re-estimation)	0.000
Batch 2 (After 1st Re-estimation)	0.331
Batch 3 (After 2nd Re-estimation)	0.363
Batch 4 (After 3rd Re-estimation)	0.369

Table 5: Normalized inversion rates for the subsequent batch, quantifying the re-sorting difference after each ADCL difficulty re-estimation event.

To verify ADCL's re-sorting mechanism, we measured the disagreement between the predefined and the dynamically re-estimated sample orders within batches using the Normalized Inversion Rate (NIR). This metric calculates the fraction of sample pairs whose relative order is flipped (0=identical, higher=more different). Table 5 shows the NIR measured for each curriculum batch after the preceding difficulty re-estimation event. The rate increased from 0 for the initial batch (Batch 1, where no re-estimation was applied) to 0.331, 0.363, and 0.369 for Batches 2, 3, and 4 (processed after the 1st, 2nd, and 3rd re-estimations, respectively). These substantial, non-zero rates confirm re-sorting occurs (affecting 33-37% of pairs). This increasing NIR trend indicates that the underlying *Difficulty Shift* becomes more pronounced as training progresses, necessitating greater adaptation by ADCL.

E Prompts for EGSR

Prompt for EGSR(a): Generating Trajectory $\tau(a)$

```
Problem:
{problem}

Think step–by–step to reach the solution. Output
your reasoning process and final answer using this
format:
<think>
[Write your complete reasoning process here,
showing each step of your thinking]
</think>

boxed{{answer}}
"""
```

Prompt for EGSR(s, a): Generating Trajectory $\tau(s, a)$

```
Problem:
{problem}

Reference Solution:
{solution}

You already know the answer, but it's important to
understand the solution in your own way. Like a
student who learns by reformulating rather than
memorizing, develop your own step–by–step
reasoning that leads to a similar conclusion.

IMPORTANT: Do NOT copy the reference solution
 verbatim. Instead, reconstruct the reasoning path
using your own understanding and expression style.
 Your goal is to demonstrate comprehension by
explaining the solution in your natural reasoning
pattern.

Use this format:
<think>
[Your original reasoning process here, showing how
 YOU would reach this solution]
</think>

boxed{{answer}}
```

F Case Study: Emergence of Flawed Reasoning Steps under Answer-Only Guidance in EGSR

We observe that if EGSR guides self-reformulation using only the final answer (a), the model, despite lower perplexity (PPL) in its outputs, tends to generate flawed reasoning pathways. It appears the model may construct its solution to forcibly arrive at the provided correct answer, a behavior reminiscent of reward hacking.

Case Study

Question:
Find the maximum value of $\frac{x+2y+3}{\sqrt{x^2+y^2+1}}$ over all real numbers x and y.

Solution:
Because we want to find the maximum value of the expression, we can assume that both x and y are positive; if not, then replacing x and y with $|x|$ and $|y|$ would strictly increase the value of the expression. By Cauchy-Schwarz, $(1^2+2^2+3^2)(x^2+y^2+1) \geq (x+2y+3)^2$, or $14(x^2+y^2+1) \geq (x+2y+3)^2$. Hence,

$$\frac{x+2y+3}{\sqrt{x^2+y^2+1}} \leq \sqrt{14}.$$

Equality occurs when $x = \frac{y}{2} = \frac{1}{3}$, so the maximum value is $\boxed{\sqrt{14}}$.

Answer: $\boxed{\sqrt{14}}$.

Flawed $\tau(a)$ **Example (Correct Answer)**:
[...] (Attempted Lagrange Multipliers, noted system was complex to solve) [...] Since the problem asks for the maximum value..., we can use the fact that the maximum value of a function of the form $\frac{ax+by+c}{\sqrt{x^2+y^2+d}}$ is given by $\sqrt{a^2+b^2+c^2}$. (Error: Cited formula is generally incorrect, ignores ’d’.) Applying this to our function: $a = 1, b = 2, c = 3, d = 1$. The maximum value is: $\sqrt{1^2+2^2+3^2} = \sqrt{1+4+9} = \sqrt{14}$. </think>
Thus, the maximum value is $\boxed{\sqrt{14}}$. (Correct answer via incorrect method.)

Correct $\tau(s, a)$ **Example**:
<think> Applying the Cauchy-Schwarz inequality with vectors $(1, 2, 3)$ and $(x, y, 1)$, we get: $(1^2 + 2^2 + 3^2)(x^2 + y^2 + 1) \geq (x \cdot 1 + 2y \cdot 1 + 3 \cdot 1)^2$, This simplifies to: $14(x^2+y^2+1) \geq (x+2y+3)^2$. [...] Taking the square root of both sides and rearranging yields:

$$\sqrt{14} \geq \frac{|x + 2y + 3|}{\sqrt{x^2 + y^2 + 1}}$$

Since we seek the maximum value, the expression is bounded above by $\sqrt{14}$. Equality holds if and only if the vectors $(1, 2, 3)$ and $(x, y, 1)$ are proportional, i.e., when $\frac{x}{1} = \frac{y}{2} = \frac{1}{3}$. Thus, the maximum value is indeed achievable. </think>
The final answer is: $\boxed{\sqrt{14}}$.