Beyond Two-Stage Training: Integrating SFT and RL for Improved Reasoning in LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reinforcement learning (RL) has proven effective in incentiving the reasoning abilities of large language models (LLMs), but faces significant efficiency challenges due to its extensive trial-and-error nature. A common practice is to employ supervised fine-tuning (SFT) as a warm-up stage; however, this decoupled two-stage approach limits interaction between SFT and RL, thereby constraining overall effectiveness. This study introduces a novel method for learning reasoning models that employs bilevel optimization to facilitate better cooperation between these training paradigms. Specifically, the SFT objective are explicitly conditioned on the optimal solution of the RL objective. During training, lower-level updates enable the model to receive SFT supervision concurrently with RL-based exploration, while upper-level updates are optimized to ensure that the joint training yields higher rewards than RL alone. Empirical evaluations on five reasoning benchmarks demonstrate that our method consistently outperforms baselines and achieves a better balance between effectiveness and efficiency.

1 Introduction

The emergence of OpenAI's o1 [21] and DeepSeek-R1 [7] represents a profound paradigm shift in Large Language Models (LLMs). Test-time scaling enables these models to execute longer Chain-of-Thought reasoning, inducing sophisticated reasoning behaviors. This capability makes them particularly effective in challenging domains such as mathematics [5, 11] and programming problems [2, 6].

The central technique driving this progress is is large-scale, rule-based reinforcement learning (RL), which induces sophisticated reasoning behaviors by exploring the reward signal. However, the inherently trial-and-error nature of RL renders the training process highly inefficient. An alternative approach is supervised fine-tuning (SFT) on curated long chain-of-thought (CoT) datasets, which enables models to rapidly acquire effective reasoning patterns through imitation learning. While more sample-efficient, SFT is typically less generalizable than RL. In practice, state-of-the-art training pipelines often adopt a two or multi-stage paradigm, using SFT as a warm-up phase before applying RL. For example, DeepSeek-R1 [7] undergoes multiple rounds of SFT and RL to refine reasoning performance. However, in these two or multi-stage pipelines, SFT and RL training are typically performed in a fully decoupled manner. This raises a natural question:

Can we design a training method that enables meaningful information exchange between the SFT and RL paradigms?

To investigate this, we first propose a simple baseline that alternates between SFT and RL updates during training. Despite its simplicity, this approach improves both convergence efficiency and final

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

performance. Building on this insight, we further develop a bilevel optimization framework, in which SFT is formulated as the upper-level problem and RL as the lower-level problem. By solving this nested optimization objective, the SFT updates are explicitly conditioned on the RL solution, allowing SFT to provide more targeted guidance to RL. This ultimately yields a model that aligns well with both supervised and reward-driven objectives.

Specifically, we implement this bilevel structure using two learnable components: a base model and a set of LoRA modules, which together form an augmented model. The base model is optimized using RL as the lower-level objective, while the LoRA parameters are updated through a supervised upper-level objective. To make this bilevel optimization tractable, we introduce a penalty-based relaxation strategy, where the relaxed upper-level update ***explicitly encourages cooperation by maximizing the reward gap between joint SFT+RL training and RL-only optimization***. In doing so, the upper-level optimization shapes the lower-level dynamics, fostering tighter alignment between supervised learning and reinforcement learning, and improving overall training efficiency.

To validate the effectiveness of our approach, we conduct experiments using the Qwen-2.5 3B model trained on the LIMR dataset, a challenging mathematical reasoning benchmark constructed from MATH [10]. We evaluate performance across six diverse benchmark datasets covering both standard and competition-level tasks. Our results demonstrate consistent improvements over six strong baselines, including supervised fine-tuning, zero-shot RL, and multi-stage SFT+RL pipelines. Notably, our method achieves superior performance in terms of both accuracy and training efficiency, confirming the benefits of tightly integrating SFT and RL through bilevel optimization.

Our work makes the following three contributions:

1. **Comparative analysis of reasoning training paradigms.** We systematically analyze and compare three prevalent strategies for training reasoning-capable language models: supervised fine-tuning (SFT), reinforcement learning (RL), and multi-stage SFT+RL pipelines. Based on this analysis, we introduce a simple yet effective alternative baseline that addresses the lack of interaction in conventional two-stage training setups.
2. **A bilevel optimization framework for integrating SFT and RL.** To promote meaningful cooperation between SFT and RL, we propose a bilevel optimization method named *BRIDGE*. BRIDGE formalizes SFT as the upper-level objective and RL as the lower-level objective, and employs a penalty-based relaxation to explicitly encourage joint training to achieve higher rewards than RL alone by maximizing the reward gap between the two.
3. **Empirical validation on six mathematical reasoning benchmarks.** We conduct extensive experiments using the Qwen-2.5 3B model trained on the LIMR dataset and evaluated across six diverse reasoning benchmarks. Our method consistently outperforms strong baselines in terms of both accuracy and training efficiency, demonstrating the practical benefits of tightly integrated SFT-RL optimization.

2 Preliminaries

We begin by reviewing three prevalent fine-tuning strategies for training reasoning models and conduct a comparative analysis. We then introduce a simple yet effective improved baseline.

2.1 Fine-tuning Methods for Reasoning Models

We consider a large language model (LLM) parameterized by $\boldsymbol{\theta}$, which defines a conditional distribution $\pi(y|x;\boldsymbol{\theta})$ over output sequences y given input sequences x. This work focuses on three widely used methodologies for tuning $\boldsymbol{\theta}$ to incentivize the model's reasoning capabilities.

Rule-based Reinforcement Learning. Reinforcement learning with verifiable rewards has gained increasing attention for its effectiveness in training advanced reasoning models such as DeepSeek-R1 [7]. Given a dataset $\mathcal{D}_{\mathrm{RL}} := \{(x, y)\}$ with verifiable outputs—such as mathematics competition problems or programming tasks—the objective of rule-based RL is formulated as:

$$\begin{aligned}\max_{\boldsymbol{\theta}} \; J_{\mathrm{RL}}(\boldsymbol{\theta}) := \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}},\, \hat{y}\sim\pi(\cdot|x;\boldsymbol{\theta})}\left[r(\hat{y}, y)\right] \\ - \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}}}\left[D_{\mathrm{KL}}\left(\pi(\cdot \mid x;\boldsymbol{\theta}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\right)\right]\end{aligned} \tag{1}$$

where π_{ref} is a fixed reference model, and $r(\hat{y}, y)$ is a rule-based reward function that evaluates the correctness of predictions using a binary signal:

$$r(\hat{y}, y) = \begin{cases} 1, & \text{if } \hat{y} \equiv y, \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

Here, y denotes the ground-truth answer and $\hat{y}$ is the model's predicted output. The equivalence relation $\hat{y} \equiv y$ is typically computed by a domain-specific verifier (e.g., a symbolic math engine or code interpreter).

Since the KL divergence term in (1) is generally not directly computable, this objective is often solved using policy optimization methods such as Proximal Policy Optimization (PPO) [22] and Group Relative Policy Optimization (GRPO) [7].

Supervised Fine-Tuning. In supervised fine-tuning, we assume access to a curated dataset $\mathcal{D}_{\text{SFT}} := \{(x, r, y)\}$ consisting of input prompts x, intermediate reasoning steps r distilled from larger reasoning models, and final answers y. The training objective maximizes the log-likelihood of generating both the reasoning process and the answer:

$$\max_{\boldsymbol{\theta}} J_{\text{SFT}}(\boldsymbol{\theta}) := \mathbb{E}_{(x,r,y)\sim\mathcal{D}_{\text{SFT}}} \left[\log \pi \left(r, y \mid x; \boldsymbol{\theta}\right)\right]. \tag{3}$$

This approach encourages the model to not only produce correct answers but also to imitate expert reasoning steps that lead to those answers.

Two-Stage Cold Start. In practice, a common recipe is to use SFT as a warm-up stage before applying RL. This two-stage approach, often referred to as a "cold start" for RL. The SFT stage ensures that the model imitate expert reasoning patterns, which provides a good prior for subsequent reward-driven optimization.

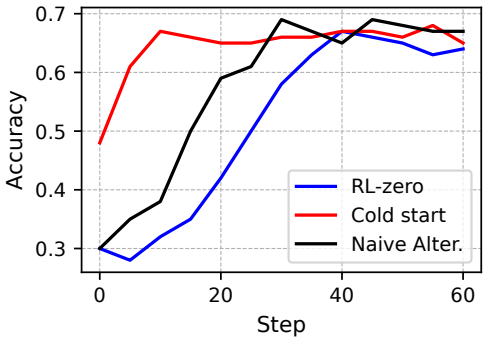

Figure 1: Comparison of Training Methods.

Algorithm 1: A Simple Alternating Method

Initialize parameters θ_0; learning rates α_{SFT}, α_{RL}; datasets D_{SFT}, D_{RL}; total steps T
for $t = 1$ to T **do**
  // RL step
  Sample query $x_t \sim D_{\text{RL}}$
  Generate solution with $\pi_{\theta_{t-1}}(x_t)$
  Compute reward r_t
  $\theta'_{t-1} \leftarrow \theta_{t-1} + \alpha_{\text{RL}} \nabla J_{\text{RL}}(\theta_{t-1})$
  // SFT step
  Sample example $(x_t, y_t) \sim D_{\text{SFT}}$
  $\theta_t \leftarrow \theta'_{t-1} + \alpha_{\text{SFT}} \nabla L_{\text{SFT}}(\theta'_{t-1})$
end for

2.2 Comparative Analysis of Fine-Tuning Strategies

We conduct a comparative study of fine-tuning strategies using the Qwen2.5-base model as the backbone. The training data consists of math problems at the grade 3–5 level, and evaluation is performed across five reasoning benchmarks, including Math500. Detailed experimental settings are provided in Section 4.1. Figure 1 illustrates how test accuracy on Math500 evolves during training.

We observe that **SFT exhibits rapid initial learning**, while **RL achieves better final convergence**. As shown in Figure 1, SFT improves accuracy quickly during the early training stages but plateaus at a suboptimal level. In contrast, RL learns more slowly but eventually surpasses SFT in final performance.

The **two-stage cold start approach combines the strengths of both paradigms**. Figure 1 further shows that the SFT warm-up phase significantly accelerates RL convergence and improves its final performance. This suggests that SFT provides a strong inductive prior, guiding the subsequent RL stage toward better optima.

These results suggest that RL and SFT offer complementary advantages in reasoning tasks, motivating further exploration of their integration.

A Simple Alternating Baseline. To further investigate the supportive role of SFT in reinforcement learning, we design a simple alternating optimization strategy between the two methods, as outlined in Algorithm . This approach alternates between reinforcement learning steps, which explore novel reasoning traces, and supervised fine-tuning steps, which imitate expert reasoning patterns (see Section 4.1 for details on the SFT dataset). As shown in Figure 1, this alternating strategy converges faster than pure RL and achieves better final performance than both SFT and the two-stage Cold-start approach. While this integration leads to empirical performance gains, the current formulation treats SFT and RL as independent update processes, and there is no guarantee that alternating updates consistently outperform either method alone. This limitation raises a important question: *How can we design training strategies that ensure the synergy between SFT and RL leads to guaranteed gains over standalone RL?*

3 Methodology

In this section, we propose BRIDGE, a framework that tightly couples SFT and RL. We will first introduce the formulation, and then the learning algorithm and explanations.

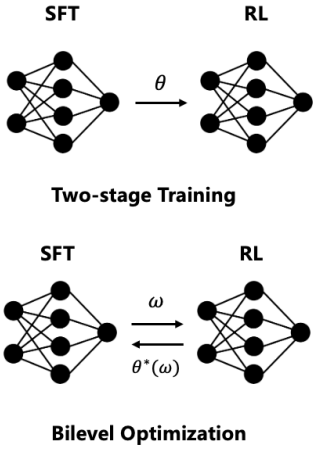

Figure 2: Comparison of two fine-tuning paradigms.

3.1 BRIDGE Framework

We define an augmented model $\bar{\theta} := [\theta, w]$, where θ denotes the base model parameters and w represents the LoRA weights [13]. Given a long-form chain-of-thought (CoT) dataset $\mathcal{D}_{\mathrm{SFT}}$ for supervised fine-tuning and a verifiable dataset $\mathcal{D}_{\mathrm{RL}}$ for reinforcement learning, our objective is to integrate the supervised learning objective in Eq. (3) with the policy optimization problem in Eq. (1). To do this, we propose to solve the following bilevel optimization problem:

$$\begin{aligned} \max_{w} \quad & J_{\mathrm{SFT}}(\theta, w) := \mathbb{E}_{(x,r,y)\sim\mathcal{D}_{\mathrm{SFT}}} \left[\log \pi\left(r, y \mid x;\ \theta^*(w), w\right)\right] \\ \text{s.t.} \quad & \theta^*(w) := \arg\max_{\theta} \Big\{ \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}},\ \hat{y}\sim\pi(\cdot|x;\theta,w)} \left[r(\hat{y}, y)\right] \\ & \qquad - \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}}} \left[D_{\mathrm{KL}}\left(\pi(\cdot \mid x; \theta, w) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\right)\right] \Big\}. \end{aligned} \tag{4}$$

The above problem has a two-level structure that draws inspiration from the leader-follower problem in game theory. SFT serves as the leader with greater decision-making power, capable of predicting the RL component's optimal response $\theta^*(w)$ for any given parameter set w during training. Meanwhile, RL acts as the follower, optimizing the base model parameters θ conditional on the SFT-determined parameters w. During training, these two components interact dynamically to achieve enhanced cooperation, resulting in improved learning outcomes. As shown in Figure 2, this structure enables a more coordinated fine-tuning process compared to the traditional two-stage recipe.

By solving (4), we aim to find a augmented model $\bar{\theta}$ such that: if one trains the base parameter θ on D_{RL}, then the fine-tuned model $\theta^*(w)$ needs to fit well with the long CoT dataset D_{SFT}.

3.2 Learning Algorithm

Following the penalty-based methods [24, 26], we next consider reformulating (4) with penalty functions. Specifically, our first goal is to reformulate (4) to a closely related single-level problem that facilitates efficient gradient-based algorithms.

We define the penalty function for the sub-optimality of the lower-level problem in (4) as:

$$p(w, \theta) = \max_{\theta'} J_{RL}(\theta', w) - J_{RL}(\theta, w) \tag{5}$$

Given a penalty constant $\gamma \in (0, 1)$, penalizing $p(w, \theta)$ onto the upper-level objective yields the following penalized problem of (6):

$$\max_{\theta, w} (1 - \lambda) J_{SFT}(\theta, w) - \lambda [\max_{\theta'} J_{RL}(\theta', w) - J_{RL}(\theta, w)] \tag{6}$$

Note that in E.q. (6), the value of the term $\max_{\theta'} J_{RL}(\theta', w)$ is solely a function of w and is independent of θ. Then we can update θ iteratively by doing stochastic gradient ascent:

$$\theta^{k+1} = \theta^k + \alpha \left[(1-\lambda)\nabla_\theta J_{\mathrm{SFT}}(\theta, w) + \lambda \nabla_\theta J_{\mathrm{RL}}(\theta, w)\right] \tag{7}$$

The penalty strength γ can be scheduled to increase at each epoch from a small value: in earlier epochs, we warm-start the base parameter on the long-CoT examples. Then we gradually increase γ for increasing accuracy in solving for $\theta^*(w)$ and a solution for the original problem in (4).

To evaluate the gradient for w, we need to evaluate $\nabla_\omega \max'_\theta J_{RL}(\theta', w)$. We assume $J_{RL}(\theta', w)$ satisfies the conditions for Danskin's theorem, and then we can write $\nabla_\omega \max'_\theta J_{RL}(\theta', w) \approx \nabla_\omega J_{RL}(\hat{\theta}, w)$, and the above gradient approximation becomes exact if $\hat{\theta} = \theta^*(\omega)$. Given this closed-form gradient, we can update ω with the approximate stochastic gradient ascent:

$$w^{k+1} = w^k + \beta \left[(1-\lambda)\nabla_w J_{\mathrm{SFT}}(\theta, w) + \lambda(\nabla_w J_{\mathrm{RL}}(\theta, w) - \nabla_w J_{\mathrm{RL}}(\hat{\theta}, w))\right] \tag{8}$$

Where $\hat{\theta}$ is the approximation of $\theta^*(\omega)$ obtained by taking one gradient ascend step on θ with respect to the J_{RL} objective:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha \nabla_\theta J_{RL}(\hat{\theta}_k, w) \tag{9}$$

Algorithm 2: Learning Algorithm of BRIDGE

Initialize augmented parameters $\bar{\theta}^0 = (\theta^0, w^0)$, and auxiliary parameters $\hat{\theta}^0 \coloneqq \theta^0$; learning rates α, β; penalty weight λ; number of iterations K
for $k = 0$ to $K-1$ **do**
  Sample mini-batches D_{SFT} and D_{RL}.
  Compute supervised objective $J_{\mathrm{SFT}}(\theta^k, w^k)$ and reinforcement objective $J_{\mathrm{RL}}(\theta^k, w^k)$.
  Compute gradients w.r.t. augmented parameters $\bar{\theta}^k = (\theta^k, w^k)$:
  $\nabla_{\bar{\theta}} J_{\mathrm{SFT}}(\theta^k, w^k) = [\nabla_\theta J_{\mathrm{SFT}},\ \nabla_w J_{\mathrm{SFT}}]$;
  $\nabla_{\bar{\theta}} J_{\mathrm{RL}}(\theta^k, w^k) = [\nabla_\theta J_{\mathrm{RL}},\ \nabla_w J_{\mathrm{RL}}]$.
  // Update lower-level variable (base parameters)
  $\theta^{k+1} \leftarrow \theta^k + \alpha \left[(1-\lambda)\nabla_\theta J_{\mathrm{SFT}}(\theta^k, w^k) + \lambda \nabla_\theta J_{\mathrm{RL}}(\theta^k, w^k)\right]$.
  // Update auxiliary $\hat{\theta}$ for upper-level (meta) gradient
  $\hat{\theta}^{k+1} \leftarrow \hat{\theta}^k - \alpha \nabla_\theta J_{\mathrm{RL}}(\hat{\theta}^k, w^k)$.
  // Update upper-level variable (LoRA parameters)
  $w^{k+1} \leftarrow w^k + \beta \left[(1-\lambda)\nabla_w J_{\mathrm{SFT}}(\theta^k, w^k) + \lambda \left(\nabla_w J_{\mathrm{RL}}(\theta^k, w^k) - \nabla_w J_{\mathrm{RL}}(\hat{\theta}^k, w^k)\right)\right]$.
end for

The overall algorithm of BRIDGE is presented in Algorithm 2.

3.3 Explanations of Update Rules

What does the lower-level update do? The update rule for θ in E.q. (7) is a convex combination of the SFT and RL gradients. As λ increases from 0 to 1 during training, the algorithm gradually shifts from imitation learning to reinforcement learning.

This curriculum learning–like transition [1] is meaningful: in the early training stages, the base model lacks strong reasoning abilities and benefits more from imitating expert reasoning patterns. As training progresses, the model becomes capable of generating the correct answers by exploring the reward signal , making RL updates more valuable.

What does the higher-level update do? The update for w in Eq. (8) aims to solve the bilevel formulation in Eq. (4). Specifically, it seeks a LoRA module w such that, after training the base parameter θ on D_{RL}, the resulting fine-tuned model $\theta^*(w)$ also performs well on the supervised dataset D_{SFT} (i.e., expert demonstrations).

The update in Eq. (8) can be interpreted as performing gradient ascent on the following objective:

$$f(\theta, w) = (1-\lambda) \underbrace{J_{\mathrm{SFT}}(\theta, w)}_{\uparrow \text{ likelihood on expert data}} + \lambda \underbrace{\left[J_{\mathrm{RL}}(\theta, w) - J_{\mathrm{RL}}(\hat{\theta}, w)\right]}_{\uparrow \text{ reward gap between collaboration and solo}} \tag{10}$$

The first term encourages maximizing the likelihood of expert reasoning patterns in D_{SFT}, while the second term increases the gap between the joint optimization using both SFT and RL (parameter θ), versus using only RL (auxiliary parameter $\hat{\theta}$). This contrastive gap explicitly promotes synergy between the SFT and RL objectives—ensuring that their joint optimization yields better performance than optimizing RL alone.

4 Experiment

4.1 Settings

Datasets. We adopt the LIMR dataset [18] for RL training, which is derived from MATH [10]. Following their setup, we use the *Hard* subset (problems with MATH difficulty levels 3–5), which contains approximately 1.3k problems. For the SFT dataset, we follow the procedure used in DeepMath-103k [9] and distill intermediate expert reasoning steps using the R1 model.

For evaluation, we use the MATH500 subset as the primary test set and uniformly sample an additional 500 problems for validation. To assess generalization, we evaluate on a diverse set of mathematical reasoning benchmarks, including MATH500 [10], Minerva Math [16], and OlympiadBench [8], as well as competition-level datasets such as AIME 2024 and AMC 2023.

Models. We conduct zero-shot RL training experiments using Qwen-2.5 models [30], selected for their demonstrated stability on mathematical reasoning tasks. The 3B model is used, with prompt formats consistent with SimpleRL.

Reward. In line with SimpleRL-Zoo [31], we adopt a binary reward function based solely on answer correctness: correct final answers receive a reward of +1, while incorrect answers receive 0. We deliberately exclude format-based rewards, which have been shown to constrain exploration and reduce final performance, particularly for base models.

Implementation Details. All models are trained using the `verl` framework [27] with unified hyperparameters: a prompt batch size of 64, 5 rollouts per prompt, a maximum rollout length of 3000 tokens, and a mini-batch size of 64. For evaluation, we use greedy decoding with a temperature of 0 and a maximum generation length of 5000 tokens. The learning rate is set to 5×10^{-7}, and for LoRA, both the rank and α are set to 16. The weighting coefficient λ is set to 0.5. Following SimpleRL-Zoo [31], we report pass@1 accuracy for most benchmarks. For AIME 2024, due to the limited number of test cases, we additionally report average accuracy over 8 samples (avg@8). All experiments are conducted on four NVIDIA A100 GPUs (80GB).

4.2 Baselines

We evaluate our approach against a comprehensive set of baselines, all built on the same base architecture. These comparisons are designed to isolate the specific contributions of our proposed bilevel optimization framework.

Base / Instruct Model. The original performance of base model and its instruction version, without further reasoning-specific training. This serves as a lower-bound reference for evaluating reasoning capabilities.

Supervised Fine-Tuning (SFT). A model trained solely via supervised fine-tuning on curated chain-of-thought (CoT) data, without any reinforcement learning. This highlights the benefits and limitations of pure imitation-based learning.

RL-Zero. Reinforcement learning applied directly to the base model without any prior fine-tuning. This baseline evaluates the effectiveness of exploration from scratch, without initialization from expert demonstrations.

Cold-Start A two-stage pipeline where SFT is used to pretrain the model, followed by RL fine-tuning. The two phases are fully decoupled, with no interaction between supervised and reward-based updates.

Naive Alternating. A simple training procedure that alternates between SFT and RL updates in fixed intervals, without any explicit coordination or shared optimization objective between the two paradigms.

4.3 Main Results

Method	MATH 500	Minerva Math	Olympiad Bench	AIME24 (Avg@8)	AMC23	Average
Base	32.4	11.8	7.9	0.0	20.0	14.4
Instruct	50.8	14.7	16.7	8.5	32.5	24.6
RL-zero	64.4	26.5	27.0	3.3	40.0	32.2
SFT	53.4	18.8	21.5	3.3	42.5	27.9 **(-13.4%)**
Cold-start	**66.0**	24.3	26.8	9.0	35.0	32.2 **(+0.0%)**
Naive Alter.	65.2	25.3	27.1	6.7	42.5	33.4 **(+3.1%)**
BRIDGE	65.4	**28.3**	**28.4**	**10.0**	**50.0**	**36.4** **(+12.4%)**

Table 1: Performance of our method compared to baselines methods across multiple math benchmarks. The best performance in each column is highlighted in green bold, and performance improvements (%) over RL-zero are shown in blue.

Generalization to Benchmarks. We evaluate the generalization ability of BRIDGE across five diverse mathematical reasoning benchmarks. As shown in Table 1, **BRIDGE** consistently outperforms baseline methods, achieving accuracy improvements of *6.8%, 12.0%, 203.0%*, and *25.0%* over RL-zero on Minerva Math, Olympiad Bench, AIME24, and AMC23, respectively. Overall, BRIDGE yields an average improvement of 12.4%, highlighting its effectiveness and robustness across tasks of varying difficulty.

Baseline methods tend to yield larger improvements on relatively easier benchmarks but generalize poorly to more complex reasoning tasks. For example, the Cold-start method underperforms RL-zero on *Minerva Math*, *Olympiad Bench*, and *AMC23*, potentially due to overfitting during the prior SFT phase. While the Naive Alternative partially mitigates this issue—maintaining performance on harder benchmarks—its gains remain limited. In contrast, BRIDGE, which explicitly encourages cooperative behavior through a reward gap mechanism, achieves consistent and substantial improvements on the more challenging benchmarks. These results underscore BRIDGE's superior generalizability in handling complex mathematical reasoning.

Method	Average Performance			Average
	Epoch=1	Epoch=3	Epoch=6	
RL-zero	14.8	17.5	32.2	21.5
SFT	24.1	26.5	27.9	26.2 **(+21.8%)**
Cold-start	**33.4**	28.5	32.2	31.4 **(+46.0%)**
Naive Alter.	13.0	30.8	33.4	25.7 **(+19.5%)**
BRIDGE	32.3	**33.3**	**36.4**	**34.0** **(+69.3%)**

Table 2: Performance progression across training epochs for different methods.

Performance on varied fine-tuning epochs. To evaluate ***the trade-off between performance and training efficiency***, we assess the effectiveness of BRIDGE across different fine-tuning epochs. We consider the average performance across multiple epochs as a metric to reflect this trade-off. As shown in Table 2, BRIDGE achieves the best balance, with an average performance improvement of *69.3%* over RL-zere.

Among the baselines, Cold-start yields the second-best trade-off. However, its performance becomes unstable as training progresses, eventually converging to the same final result as RL-zero. In contrast, BRIDGE demonstrates consistent improvement throughout training. Overall, nearly all hybrid baselines outperform RL-zero in terms of early-stage efficiency, highlighting the advantage of integrating supervised fine-tuning and reinforcement learning paradigms.

5 Related Work

Reinforcement Learning for Large Reasoning Models. Recent progress has highlighted the critical role of reinforcement learning in enhancing the reasoning capabilities of large language models [21, 7]. DeepSeek-R1 introduced a simple yet effective rule-based reward model and demonstrated further gains through multiple rounds of supervised distillation and RL training. LIMR [18] showed that complex reasoning behaviors can emerge from as few as one thousand curated examples from the MATH dataset [11].

In parallel, substantial advances have been made in training recipes for large reasoning models. Chu et al. [4] compare SFT and RL for reasoning tasks and find that RL generalizes significantly better, whereas SFT is prone to overfitting. SimpleRL [31] observes that fine-tuning on short-CoT datasets can harm reasoning ability, while He et al. [9] find that fine-tuning on long-CoT distilled data can improve the reasoning performance of smaller models—especially when used as a warm-up stage before RL training. In practice, two-stage pipelines that combine SFT and RL are commonly used to balance stability and performance. However, existing approaches often rely solely on supervised fine-tuning, which tends to generalize poorly, or on pure RL, which suffers from sample inefficiency and unstable optimization. In this work, we propose the first unified training framework that enables explicit interaction between SFT and RL via a bilevel optimization formulation. This approach offers a new perspective on integrating imitation and exploration for reasoning-centric large language models.

Bilevel Optimization in LLMs. Bilevel optimization (BLO) is a classical framework for modeling nested learning problems, where an upper-level objective depends on the solution to a lower-level optimization task. Two major classes of methods have been developed to solve BLO problems. Implicit gradient methods [12, 14, 23, 29] compute gradients through the lower-level problem using second-order derivatives. While theoretically robust, these methods are often computationally expensive and memory-prohibitive when applied to large-scale models such as LLMs. In contrast, penalty-based relaxation methods [24, 15, 25, 20] approximate the BLO formulation using only first-order gradients, making them substantially more scalable and thus better suited for LLM applications. Recent work has explored the use of bilevel optimization in LLMs for tasks such as data selection [19, 26], inverse reinforcement learning [17], and meta-learning [3, 28]. To the best of our knowledge, our work is the first to apply bilevel optimization to reasoning-oriented LLM training, providing a principled approach to integrating supervised and reinforcement learning in a unified framework.

6 Conclusion

This work investigates how to effectively integrate supervised fine-tuning and reinforcement learning to improve the reasoning capabilities of large language models. We begin by analyzing three widely used training paradigms and identify a key limitation of existing multi-stage pipelines: the lack of interaction between SFT and RL. To address this, we propose a simple alternating baseline and further introduce *BRIDGE*, a bilevel optimization framework that models SFT as the upper-level objective and RL as the lower-level objective. By employing a penalty-based relaxation, BRIDGE explicitly encourages joint training to outperform standalone RL, fostering tighter synergy between the two learning paradigms. Empirical results on six mathematical reasoning benchmarks demonstrate that our method consistently outperforms strong baselines in both accuracy and training efficiency. These findings underscore the potential of bilevel optimization as a unifying framework for combining supervised and reward-driven learning in complex reasoning tasks.

Limitations. While BRIDGE demonstrates promising results, it introduces additional computational overhead due to its nested bilevel optimization structure. Future work includes extending the framework to larger-scale models and more diverse domains such as program synthesis, theorem proving, and scientific reasoning, as well as exploring more efficient optimization strategies to mitigate the computational cost.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

[2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[3] Sang Keun Choe, Sanket Vaibhav Mehta, Hwijeen Ahn, Willie Neiswanger, Pengtao Xie, Emma Strubell, and Eric Xing. Making scalable meta learning practical. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Xazhn0JoNx.

[4] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025. URL https://openreview.net/forum?id=d3E3LWmTar.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Codeforces. Codeforces - competitive programming platform, 2025. URL https://codeforces.com/. Accessed: 2025-03-18.

[7] 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, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[8] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[9] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025. URL https://arxiv.org/abs/2504.11456.

[10] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[11] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[12] M Hong, HT Wai, Z Wang, and Z Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic, dec. 20. *arXiv preprint arXiv:2007.05170*, 2020.

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[14] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. In *Advances in neural information processing systems*, 2021.

[15] Jeongyeol Kwon, Dohyun Kwon, Steve Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. *arXiv preprint arXiv:2309.01753*, 2023.

[16] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[17] Jiaxiang Li, Siliang Zeng, Hoi-To Wai, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment, 2024. URL `https://arxiv.org/abs/2405.17888`.

[18] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025. URL `https://arxiv.org/abs/2502.11886`.

[19] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 365–374, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657807. URL `https://doi.org/10.1145/3626772.3657807`.

[20] Songtao Lu. Slm: A smoothed first-order lagrangian method for structured constrained nonconvex optimization. 2024.

[21] OpenAI. Learning to reason with llms. urlhttps://openai.com/index/learning-to-reason-with-llms/. Accessed: 15 March 2025.

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

[23] Han Shen and Tianyi Chen. A single-timescale analysis for stochastic approximation with multiple coupled sequences. 2022.

[24] Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International Conference on Machine Learning*, 2023.

[25] Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. 2024.

[26] Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=VHguhvcoM5`.

[27] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

[28] Reza Shirkavand, Qi He, Peiran Yu, and Heng Huang. Bilevel zofo: Bridging parameter-efficient and zeroth-order techniques for efficient llm fine-tuning and meta-training, 2025. URL https://arxiv.org/abs/2502.03604.

[29] Quan Xiao, Han Shen, Wotao Yin, and Tianyi Chen. Alternating implicit projected sgd and its efficient variants for equality-constrained bilevel optimization. 2023.

[30] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[31] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The experiments support claims in the Abstract and Introduction. The abstract and introduction provide a comprehensive overview of the background and motivation of this study, effectively outlining its main contributions point by-point, thus accurately reflecting the paper's scope and significance.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We primarily focused on discussing the limitations associated with this study in Section Conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper includes the full set of correct proofs for each theoretical result, primarily presented in Section **??**. In particular, it covers the formulation of reward function and PPO algorithm, ensuring completeness and accuracy in the theoretical presentation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All information regarding the key contribution of this paper, have been fully disclosed (to the extent that it affects the main claims and/or conclusions of the paper). Data preprocessing steps are provided in Section 4.1. The code will be deanonymized upon acceptance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material submitted with the manuscript includes open access to all source code and scripts necessary to faithfully reproduce the main experimental results. Instructions for running the code are also provided within the scripts. We will also release the code on github after the notification decision.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies detailed experimental configurations in Section **??**, providing readers with essential information to comprehend the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: We did not include an analysis of the statistical significance of the experiments mainly due to the prohibitively expensive training cost of large language models and our limited computing resources. However, we have provided the code, hyperparameters, and random seeds used in our experiments to facilitate the reproducibility of our findings. We would like to point out that, due to the extensive amount of training data, the statistical patterns of the experiment results are likely to remain consistent across different trials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were carried out on an 4 × A100 GPU server, as detailed at the beginning of the experiment section (Section **??**).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: After carefully reviewing the referenced document, we certify that the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper primarily focuses on math LLMs trained using publicly available datasets that have undergone thorough validation. While the code LLMs itself is not directly applicable to everyday scenarios, it serves as a neutral and valuable toolkit for further development and research.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The proposed models are based on Qwen2.5 and trained on benchmark dataset MATH. The pretrain language models and dataset have been extensively used in the large language model community and have undergone comprehensive safety. risk assessments.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In the paper, we specified the pretrain language model, dataset and code sources used (e.g., verl), and provided appropriate citations in the reference section. Additionally, we ensured transparency by including the sources of any modified code files, making the changes traceable.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have included the code, along with detailed usage instructions, in the github. After the review process is completed, we will make the code publicly available to the community.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This study does not involve any crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This study does not involve any crowdsourcing experiments or research with human subjects. All experiments were conducted using code and GPU servers.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed models are based on Qwen2.5 LLMs. The paper specifies detailed experimental configurations, describing the usage of LLMs as a base model and an important component in our study.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.