Beyond Two-Stage Training: Cooperative SFT and RL for LLM Reasoning

Liang Chen[1] **Xueting Han**[2†] **Li Shen**[3] **Jing Bai**[2] **Kam-Fai Wong**[1†]
[1]The Chinese University of Hong Kong [2]Microsoft Research
[3]Shenzhen Campus of Sun Yat-sen University

Abstract

Reinforcement learning (RL) has proven effective in incentiving the reasoning abilities of large language models (LLMs), but suffers from severe efficiency challenges due to its trial-and-error nature. While the common practice employs supervised fine-tuning (SFT) as a warm-up stage for RL; however, this decoupled two-stage approach limits interaction between SFT and RL, thereby constraining overall effectiveness. This study introduces BRIDGE, a novel method for learning reasoning models that employs bilevel optimization to facilitate better cooperation between these training paradigms. By conditioning the SFT objective on the optimal RL policy, our approach allows SFT to dynamically adapt its guidance based on RL's exploration needs. During training, the lower-level performs RL updates while continuously receiving SFT supervision, while the upper-level explicitly maximizes the cooperative gain—the performance advantage of joint SFT-RL training over RL alone. Empirical evaluations across three LLMs and 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 [23] and DeepSeek-R1 [7] represents a profound paradigm shift in 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, 13] 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?

[†]Corresponding authors: chrihan@microsoft.com, kfwong@se.cuhk.edu.hk.
[‡]The code will be available at `https://github.com/ChanLiang/BRIDGE`

Preprint. Under review.

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 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 three LLMs trained on the two datasets and evaluate performance across five diverse benchmark datasets covering both standard and competition-level tasks. Our results demonstrate consistent improvements over five strong baselines, including SFT, RL-zero, cold-start and our proposed naive alternating. Notably, BRIDGE 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*. It 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 three LLMs evaluated across six diverse reasoning benchmarks. Our method consistently outperforms strong baselines in 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, conduct a comparative analysis, and discuss limitations of the popular two-stage method. We then introduce a simple yet effective baseline that improves upon it.

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 fine-tuning $\boldsymbol{\theta}$ to enhance the model's reasoning capabilities.

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

$$\max_{\boldsymbol{\theta}} \; J_{\mathrm{SFT}}(\boldsymbol{\theta}) := \mathbb{E}_{(x,r,y)\sim\mathcal{D}_{\mathrm{SFT}}} \left[\log \pi\left(r, y \mid x; \boldsymbol{\theta}\right)\right]. \tag{1}$$

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

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—the objective of rule-based RL is formulated as:

$$\begin{aligned}\max_{\boldsymbol{\theta}} J_{\mathrm{RL}}(\boldsymbol{\theta}) := \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}},\ (\hat{r},\hat{y})\sim\pi(\cdot|x;\boldsymbol{\theta})}\left[R(\hat{y}, y)\right] \\ - \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RL}}}\left[D_{\mathrm{KL}}\left(\pi(\cdot \mid x;\boldsymbol{\theta}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\right)\right]\end{aligned} \tag{2}$$

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

$$R(\hat{y}, y) = \begin{cases} 1, & \text{if } \hat{y} \equiv y, \\ -1, & \text{otherwise} \end{cases} \tag{3}$$

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). This objective is commonly solved using policy optimization methods such as Proximal Policy Optimization (PPO) [24] or Group Relative Policy Optimization (GRPO) [7].

Two-Stage Cold Start. In practice, the common recipe uses SFT as a warm-up stage before applying RL. This two-stage approach, often referred to as "cold start," ensures that the model first learns to imitate expert reasoning patterns, providing a strong initialization for subsequent RL training.

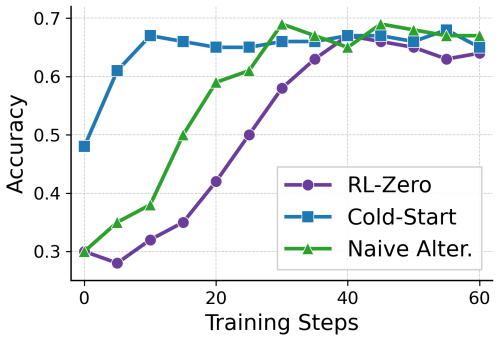

Figure 1: Comparison of Training Methods.

Algorithm 1: A Simple Alternating Method

Initialize parameters θ_0; datasets D_{SFT}, D_{RL}; learning rates α_{SFT}, α_{RL}; total steps T
for $t = 1$ to T **do**
  // RL step
  Sample $(x_t, y_t) \sim D_{\mathrm{RL}}$
  Generate solution with $\pi_{\theta_{t-1}}(x_t)$
  Compute RL objective J_{RL} using (2)
  $\theta'_{t-1} \leftarrow \theta_{t-1} + \alpha_{\mathrm{RL}} \nabla J_{\mathrm{RL}}(\theta_{t-1})$
  // SFT step
  Sample example $(x_t, r_t, y_t) \sim D_{\mathrm{SFT}}$
  Compute SFT objective J_{SFT} using (1)
  $\theta_{t-1} \leftarrow \theta'_{t-1} + \alpha_{\mathrm{SFT}} \nabla J_{\mathrm{SFT}}(\theta'_{t-1})$
end for

2.2 Comparison of Fine-Tuning Methods

We evaluate these methods on mathematics problems at the grade 3–5 level. Figure 1 illustrates the evolution of test accuracy during training. We observe that *while SFT provides effective initialization and rapid early convergence for cold-start training, it contributes little to final convergence performance*. This results in faster initial accuracy improvements, but performance plateaus with minimal gains in the later stages of the two-phase pipeline. In contrast, RL alone converges more slowly but eventually achieves comparable final performance.

These results suggest that SFT and RL offer complementary strengths in reasoning tasks: SFT facilitates rapid initial learning, while RL enables better asymptotic performance. However, the naïve two-stage combination in cold-start training fails to fully exploit these complementary advantages. We identify two key limitations:

1. **Catastrophic forgetting:** The two-stage paradigm suffers from catastrophic forgetting—the model loses valuable SFT-acquired knowledge when transitioning to RL training. This phenomenon is evident in the response length dynamics during cold-start's second stage (see Figure 3). Response lengths initially drop sharply before gradually recovering, exhibiting a "dip-then-rise" pattern that indicates the model first forgets some expert behaviors before slowly exploring new strategies.

2. **Inefficient exploration:** Despite effective SFT initialization, online RL frequently encounters inefficient exploration, particularly on challenging problems where LLMs fail to generate reward-yielding solutions. LLMs often become trapped in local optima, unable to discover trajectories that yield positive rewards (see Figure 3). Moreover, once the initial SFT phase concludes, it cannot provide continued guidance for difficult problems.

These limitations motivate integrating SFT and RL training within a unified framework.

2.3 A Simple Alternating Baseline

To investigate the potential synergy between SFT and RL, we design a simple alternating optimization strategy, as outlined in Algorithm 1. This approach alternates between RL steps, which explore novel reasoning strategies, and SFT steps, which imitate expert reasoning patterns.

As shown in Figure 1, this alternating strategy converges faster than pure RL and achieves better final performance than both standalone SFT and two-stage cold-start training. While this integration yields empirical gains, the current formulation treats SFT and RL as ***independent update*** processes with ***no guarantee*** that alternating updates will consistently outperform RL method alone. This limitation raises a natural question: *How can we design training strategies that ensure better cooperation between SFT and RL leads to guaranteed superior performance compared to standalone RL?*

3 Methodology

In this section, we propose BRIDGE, a framework that tightly couples SFT and RL through a cooperative meta-learning approach. We first introduce the mathematical formulation, then present the learning algorithm and explanations.

3.1 BRIDGE: Cooperative Meta-Learning for SFT and RL

Given an SFT dataset $\mathcal{D}_{\mathrm{SFT}}$ and an RL dataset $\mathcal{D}_{\mathrm{RL}}$ (defined in Section 2.1), our objective is to integrate policy optimization (Eq. (2)) with supervised learning (Eq. (1)). We propose the following cooperative meta-learning formulation:

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

where θ denotes the base model parameters and w represents the Low-Rank Adaptation (LoRA) weights [15]. Together, they form an augmented model with parameters $\bar{\theta} := [\theta, w]$.

For clarity, we express Equation (4) in simplified notation:

$$\begin{aligned} \max_{w} \quad & J_{\mathrm{SFT}}(w, \theta^*(w)), \\ \text{s.t.} \quad & \theta^*(w) := \arg\max_{\theta} J_{\mathrm{RL}}(\theta, w). \end{aligned} \tag{5}$$

This formulation exhibits a *bilevel optimization* structure inspired by leader-follower game. SFT acts as the leader (teacher) with access to the RL follower's (student's) optimal response $\theta^*(w)$, enabling it to provide targeted guidance. Conversely, RL optimizes the base parameters θ given the auxiliary support from SFT through w. During training, these components interact dynamically, resulting in better cooperation. As illustrated in Figure 2, this structure enables bidirectional information flow—where RL's optimal solution becomes visible to SFT—in contrast to the unidirectional flow of traditional two-stage approaches. From a *meta-learning* perspective, BRIDGE implements cooperative framework where, at each iteration, the upper-level SFT provides an improved initialization for RL exploration, while the lower-level RL refines this initialization through reward-based optimization. This framework *adaptively extracts the most beneficial information from SFT to enhance RL training*, as SFT guidance may not always be uniformly beneficial.

The single-stage cooperative meta-learning design provides three distinct advantages: (1) eliminates catastrophic forgetting of the two-stage pipeline through unified single-stage training; (2) improves

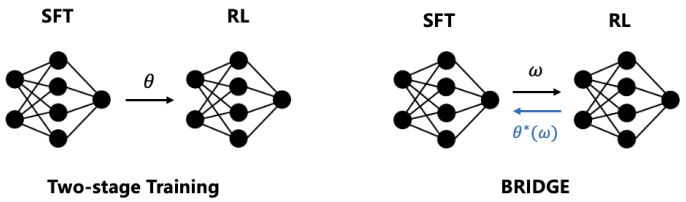

Figure 2: Comparison of two training methods.

exploration efficiency via continuous supervised guidance; and (3) guarantees RL performance gains by learning to learn from SFT signals.

Architectural Design Rationale. The augmented model architecture, comprising base model parameters θ and LoRA parameters w, is essential for enabling cooperative learning. This separation allows the upper- and lower-level objectives to *co-adapt* during training, as illustrated in Figure 2. Without this architectural separation, our formulation (Equation (4)) would collapse to a Model-Agnostic Meta-Learning (MAML)-style setup [8], where the lower-level solution reduces to a single gradient step used to update the upper-level SFT parameters. In this case, RL learning is disabled, and the cooperation between SFT and RL is lost.

3.2 Learning Algorithm

To solve the bilevel optimization problem in Eq. (5), we employ penalty-based methods [26, 28] to avoid expensive second-order derivative computations. We first reformulate (5) as a single-level problem amenable to efficient first-order optimization.

We define the penalty function measuring the sub-optimality of the lower-level problem as:

$$p(w, \theta) = \max_{\theta'} J_{\mathrm{RL}}(\theta', w) - J_{\mathrm{RL}}(\theta, w). \tag{6}$$

This penalty quantifies the optimality gap: $p(w, \theta) = 0$ if and only if θ maximizes $J_{\mathrm{RL}}(\cdot, w)$.

Given a penalty weight $\lambda \in (0, 1)$, we obtain the penalized reformulation:

$$\max_{\theta, w} \ \mathcal{L}(\theta, w) := (1 - \lambda) J_{\mathrm{SFT}}(\theta, w) - \lambda p(w, \theta). \tag{7}$$

The penalty weight λ follows an annealing schedule: starting from a small value to warm-start training on supervised data, then gradually increasing to enforce the bilevel constraint more strictly.

Since $\max_{\theta'} J_{\mathrm{RL}}(\theta', w)$ depends only on w, the gradient with respect to θ simplifies to:

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

For the gradient with respect to w, we invoke Danskin's theorem. Assuming $J_{\mathrm{RL}}(\cdot, w)$ satisfies the required regularity conditions, we have:

$$\nabla_w \max_{\theta'} J_{\mathrm{RL}}(\theta', w) = \nabla_w J_{\mathrm{RL}}(\theta^*(w), w), \tag{9}$$

where $\theta^*(w) = \arg\max_\theta J_{\mathrm{RL}}(\theta, w)$. In practice, we approximate $\theta^*(w)$ by taking a single gradient ascent step with respect to the RL objective:

$$\hat{\theta} = \theta + \alpha \nabla_\theta J_{\mathrm{RL}}(\theta, w), \tag{10}$$

yielding the approximate gradient update for w:

$$\nabla_w \mathcal{L}(\theta, w) \approx (1 - \lambda) \nabla_w J_{\mathrm{SFT}}(\theta, w) + \lambda \left[\nabla_w J_{\mathrm{RL}}(\theta, w) - \nabla_w J_{\mathrm{RL}}(\hat{\theta}, w) \right]. \tag{11}$$

The overall algorithm of BRIDGE is presented in Algorithm 2.

Algorithm 2: Learning Algorithm of BRIDGE

Initialize augmented parameters $\bar{\theta}^0 = (\theta^0, w^0)$, and auxiliary parameters $\hat{\theta}^0 := \theta^0$; learning rates α, β; penalty weight λ; number of iterations K
for $k = 0$ to $K - 1$ **do**
  Sample mini-batches $\mathcal{B}_{\mathrm{SFT}} \sim \mathcal{D}_{\mathrm{SFT}}$ and $\mathcal{B}_{\mathrm{RL}} \sim \mathcal{D}_{\mathrm{RL}}$
  // Compute base objectives
  Compute $J_{\mathrm{SFT}}(\theta^k, w^k)$, $J_{\mathrm{RL}}(\theta^k, w^k)$ and $J_{\mathrm{RL}}(\hat{\theta}^k, w^k)$ on $\mathcal{B}_{\mathrm{SFT}}$ and $\mathcal{B}_{\mathrm{RL}}$
  // Define composite objectives
  $J_{\mathrm{Joint}}(\theta^k, w^k) = (1-\lambda) J_{\mathrm{SFT}}(\theta^k, w^k) + \lambda J_{\mathrm{RL}}(\theta^k, w^k)$
  $J_{\mathrm{Gain}}(w^k) = (1-\lambda) J_{\mathrm{SFT}}(\theta^k, w^k) + \lambda [J_{\mathrm{RL}}(\theta^k, w^k) - J_{\mathrm{RL}}(\hat{\theta}^k, w^k)]$
  // Update base parameters via joint objective
  $\theta^{k+1} \leftarrow \theta^k + \alpha \nabla_\theta J_{\mathrm{Joint}}(\theta^k, w^k)$
  // Update auxiliary parameters via pure RL
  $\hat{\theta}^{k+1} \leftarrow \hat{\theta}^k + \alpha \nabla_{\hat{\theta}} J_{\mathrm{RL}}(\hat{\theta}^k, w^k)$
  // Update LoRA parameters to maximize cooperative gain
  $w^{k+1} \leftarrow w^k + \beta \nabla_w J_{\mathrm{Gain}}(w^k)$
end for

3.3 Intuition Behind the Update Rules

Lower-level update: Curriculum-weighted gradient fusion. The update rule for θ in Eq. (8) performs a convex combination of SFT and RL gradients. As λ increases from 0 to 1 during training, the algorithm smoothly transitions from pure imitation learning to pure reinforcement learning.

This adaptive curriculum [1] reflects the model's evolving capabilities: early in training, when the base model lacks strong reasoning abilities, it benefits primarily from imitating expert demonstrations. As the model develops competence in generating correct solutions, it can increasingly leverage reward signals through exploration, making RL updates progressively more valuable.

Upper-level update: Maximizing cooperative gain. The update for w in Eq. (11) solves the bilevel problem by finding LoRA parameters w that ensure the RL-optimized model $\theta^*(w)$ also excels on the supervised dataset $\mathcal{D}_{\mathrm{SFT}}$.

The update in Eq. (11) can be interpreted as performing gradient ascent on the following objective:

$$f(\theta, w) = (1-\lambda) \underbrace{J_{\mathrm{SFT}}(\theta, w)}_{\uparrow \text{ likelihood on expert data}} + \lambda \underbrace{\left[J_{\mathrm{RL}}(\theta, w) - J_{\mathrm{RL}}(\hat{\theta}, w) \right]}_{\uparrow \text{ cooperative gain: SFT-RL vs RL-only}} \tag{12}$$

The first term maintains alignment with expert reasoning patterns, while the second term—**the cooperative advantage**—quantifies how much the joint SFT-RL optimization (using θ) outperforms pure RL training (using $\hat{\theta}$). By maximizing this advantage term, the algorithm *explicitly encourages cooperation between supervised and reinforcement learning, ensuring their combination yields superior performance compared to RL alone.*

4 Experiment

4.1 Settings

Datasets. We use two datasets for RL training: LIMR [20] containing 1.3k unique problems and MATH [12] with 8.5k problems. For the SFT dataset, we pair queries from LIMR and MATH with corresponding intermediate reasoning traces extracted from DeepSeekMath-103k [11], which were distilled from the DeepSeek-R1 model.

We evaluate on seven benchmarks: three core mathematical reasoning datasets (MATH500 [12], Minerva Math [18], and OlympiadBench [10]), two competition-level benchmarks (AIME 2024 and AMC 2023).

Models. To demonstrate the generality of our approach, we experiment with three LLMs: Qwen2.5-3B [32], Llama-3.2-3B-Instruct [9], and Qwen2-8B-Base [33]. All models use prompt formats consistent with SimpleRL [34].

Reward Function. Following SimpleRL [34], we employ a binary reward based on answer correctness: +1 for correct final answers and 0 otherwise. We deliberately exclude format-based rewards, which can constrain exploration and reduce performance, particularly for base models.

Implementation Details. All models are trained using the VERL framework [29]. We use a prompt batch size of 64, mini-batch size of 64, and learning rate of 5×10^{-7}. For LoRA, we set both rank and α to 16. The penalty weight λ is set to 0.5. We employ two configurations: (1) for 3B models: 5 rollouts per prompt with 3k maximum tokens; (2) for 8B models: 8 rollouts per prompt with 8k maximum tokens. During evaluation, we use greedy decoding (temperature 0) with a 5k or 8k token limit and report pass@1 accuracy. Experiments are conducted on 4×NVIDIA A100 GPUs (80GB) for 3B models and 8×AMD MI300 GPUs (192GB) for 8B models.

4.2 Baselines

We compare BRIDGE against five baselines on the same base architectures:

Base/Instruct Model. The base model or its instruction-tuned variant without additional reasoning-specific training, serving as performance lower bounds.

Supervised Fine-Tuning (SFT). Models trained exclusively on curated reasoning traces without reinforcement learning, demonstrating the capabilities and limitations of pure imitation learning.

RL-Zero. Reinforcement learning applied directly to the base model without prior fine-tuning, evaluating the effectiveness of exploration from scratch.

Cold-Start A two-stage pipeline with SFT pretraining followed by RL fine-tuning, where phases are fully decoupled with no interaction between objectives.

Naive Alternating. We introduce this baseline as an ablation study, which alternates between SFT and RL updates without the cooperative optimization. Despite being simple and effective, this straightforward independent alternating optimization approach allows us to isolate the additional gains from BRIDGE's cooperative mechanism.

4.3 Experimental Results

Method	MATH 500	Minerva Math	Olympiad Bench	AIME24	AMC23	Average
Base	32.4	11.8	7.9	0.0	20.0	14.4
SFT	53.4	18.8	21.5	3.3	42.5	27.9
RL-zero	64.4	26.5	27.0	3.3	40.0	32.2
Cold-start	66.0	24.3	26.8	9.0	35.0	32.2
Naive Alter.	65.2	25.3	27.1	6.7	42.5	33.4 **(+3.7)**
BRIDGE	66.2	23.9	28.9	13.3	47.5	36.0 **(+11.8)**

Table 1: Performance of BRIDGE compared to baseline methods across five math benchmarks. Average performance improvements (%) over Cold-start are highlighted 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 consistently accuracy improvements on Minerva Math, Olympiad Bench, AIME24, and AMC23. Overall, BRIDGE yields an average improvement of 11.8% over RL-zero and Cold-start, 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 achieves consistent and substantial improvements on the more challenging benchmarks. These results underscore BRIDGE's superior generalizability in handling complex mathematical reasoning.

Method	MATH 500	Minerva Math	Olympiad Bench	AIME24	AMC23	Average
Instruct	38.0	14.3	13.0	13.3	25.0	20.7
SFT	38.4	10.3	11.9	27.5	3.3	18.3
RL-zero	48.6	15.1	17.8	10.0	17.5	21.8
Cold-start	45.0	11.8	12.0	3.3	22.5	18.9
Naive Alter.	49.8	17.6	17.2	20.0	0.0	20.9 **(+10.6%)**
BRIDGE	51.8	15.1	19.3	10.0	27.5	24.7 **(+30.7%)**

Table 2: Performance on Llama3.2-3B-Instruct.

Method	MATH 500	Minerva Math	Olympiad Bench	AIME24	AMC23	Average
Base	55.4	24.3	22.5	3.3	27.5	26.6
SFT	67.8	32.0	29.8	45.0	13.3	37.6
RL-zero	76.2	36.0	42.4	10.0	50.0	42.9
Cold-start	80.4	38.2	39.6	16.6	52.5	45.5
Naive Alter.	78.2	37.5	40.6	65.0	13.3	46.9 **(+3.1%)**
BRIDGE	79.0	39.7	44.0	16.7	70.0	49.9 **(+9.7%)**

Table 3: Performance on Qwen3-8B-Base.

Effectiveness across LLMs. We expand our experiments to additional LLMs: Qwen3-8B-Base and Llama3.2-3B-Instruct. As shown in Tables 3 and 2, BRIDGE consistently outperforms all baselines across diverse architectures. On Qwen3-8B-Base, BRIDGE achieves 16.3% improvement over RL-zero and 9.7% over Cold-start. On Llama3.2-3B-Instruct, gains are more pronounced with 13.5% over RL-zero and 30.9% over Cold-start. These results demonstrate BRIDGE's robust effectiveness across different model families and training configurations.

Performance on varied fine-tuning epochs. We assess BRIDGE's effectiveness across different fine-tuning epochs on Qwen2.5-3B using average performance across epochs as the metric. As shown in Table 4, BRIDGE achieves the highest average performance.

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.

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

Table 4: Performance progression across training epochs for different methods.

Training Dynamics Analysis. We analyze the dynamics of mean reward and response length during training for BRIDGE, Cold-start, and RL-Zero on Qwen2.5-3B. As shown in Figure 3, the three methods exhibit markedly different patterns. RL-Zero suffers from online RL's sample inefficiency, showing slow growth in both response length and reward. Cold-start begins with extremely long responses due to SFT warm-up, causing slow initial training, followed by a sharp decline and gradual recovery. This "dip-then-rise" pattern indicates the model initially loses expert behavior acquired during SFT, then slowly explores new strategies—a mismatch that contributes to training inefficiency.

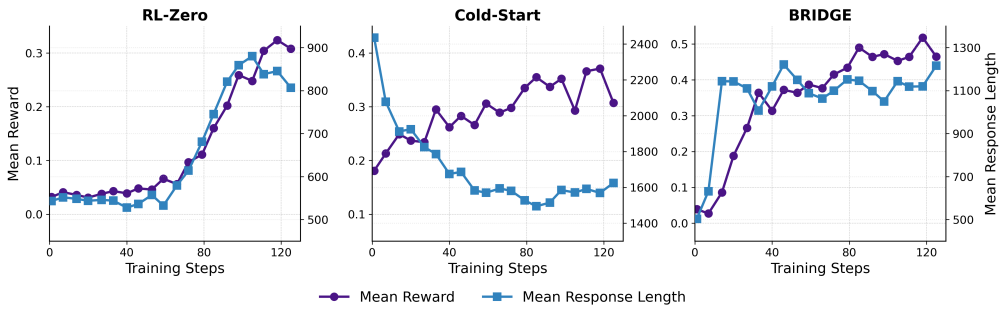

Figure 3: Training dynamics of mean reward and response length for BRIDGE, Cold-start, and RL-zero on Qwen2.5-3B.

Despite starting with higher rewards, Cold-start's second-phase RL lacks proper guidance, resulting in convergence similar to RL-Zero. In contrast, BRIDGE benefits from continuous SFT guidance throughout training, enabling rapid reward growth that surpasses Cold-start and achieving superior convergence. These dynamics demonstrate that BRIDGE's bilevel optimization enables more efficient policy learning through sustained and targeted expert guidance.

Cost-Benefit Analysis. We evaluated the cost-performance trade-offs by measuring wall-clock training time, average GPU memory usage per device, and final convergence performance across two model scales: Qwen2.5-3B (4×A100-80GB) and Qwen3-8B-Base (8×MI300-192GB). As shown in Table 5, Cold-start requires nearly 2x the training time of RL-zero, despite the short SFT stage. This overhead stems from long sequence lengths induced by the SFT stage (see Figure 3). BRIDGE achieved 44% and 14% time savings compared to Cold-start for the 3B and 8B models, respectively. Despite a modest 11% increase in memory usage for the larger model, BRIDGE consistently delivered superior performance improvements (13% for 3B and 9.7% for 8B models), demonstrating favorable cost-benefit trade-offs for practical deployment.

Metric	Qwen 2.5-3B			Qwen 3-8B-Base		
	RL-zero	Cold-start	BRIDGE	RL-zero	Cold-start	BRIDGE
Time (hr)	6.1	12.3	6.9	38.5	39.1	33.5
Mem. (GB)	52.2	45.9	59.3	50.7	60.8	67.4
Acc. (%)	32.2	32.2	36.4	42.9	45.5	49.9

Table 5: Cost-performance analysis on Qwen2.5-3B and Qwen3-8B-Base

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 [23, 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 [20] showed that complex reasoning behaviors can emerge from as few as one thousand curated examples from the MATH dataset [13].

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 [34] observes that fine-tuning on short-CoT datasets can harm reasoning ability, while He et al. [11] 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 large reasoning 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 [14, 16, 25, 31] 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 [26, 17, 27, 22] 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 [21, 28], inverse reinforcement learning [19], and meta-learning [3, 30]. 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.

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] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/finn17a.html`.

[9] 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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hos-

seini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu

Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

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

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

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

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

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

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[34] 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.