LEARNING USEFUL SUPERVISION FOR REINFORCEMENT LEARNING IN REASONING MODELS

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 Asia
[3]Shenzhen Campus of Sun Yat-sen University

ABSTRACT

Supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR) are two widely used post-training paradigms for improving the reasoning ability of large language models (LLMs). Recent methods attempt to integrate SFT and RLVR in a single stage by reweighting or scheduling their objectives. However, such coupling can be counterproductive because supervised updates are not uniformly beneficial for reward optimization, which can diminish reward gains of RL. To address this, we propose BRIDGE, a scalable framework in which SFT learns to supervise RL by selectively transferring knowledge that improves reward optimization. Specifically, BRIDGE employs two nested optimization loops during meta-training: the inner loop updates base model parameters using a fused SFT–RL gradient. Concurrently, the outer loop updates a lightweight low-rank adapter (LoRA) to coordinate the two objectives by maximizing a reward-gap signal, defined as the reward of joint SFT–RL training over an RL-only baseline. Across three model scales and five reasoning benchmarks, BRIDGE consistently outperforms two-stage cold start, naive mixing, and representative single-stage integration baselines, yielding over three points average absolute improvement and more stable training dynamics.

1 INTRODUCTION

Large reasoning models (LRMs) have demonstrated strong performance across a range of domains, particularly in challenging tasks such as mathematics (Cobbe et al., 2021; Hendrycks et al., 2021b; Chen et al., 2026) and programming (Chen et al., 2021; Codeforces, 2025). Two post-training paradigms are widely used to elicit such reasoning capabilities: supervised fine-tuning (SFT) (Muennighoff et al., 2025) and reinforcement learning with verifiable rewards (RLVR) (DeepSeek-AI et al., 2025). These paradigms offer complementary strengths. SFT can mimic high-quality expert trajectories efficiently, but it is prone to overfitting (Chu et al., 2025). RLVR, in contrast, encourages the policy to actively explore reward-yielding trajectories, which can improve generalization (Song, 2025; Jiang et al., 2023), but it is inefficient due to trial-and-error search. A common recipe therefore uses a two-stage SFT-then-RL pipeline. However, this pipeline does not consistently outperform pure RL (Table 2), as also reported in prior work (Zhang et al., 2025b;a). These observations motivate more effective approaches to integrating the two paradigms.

Existing methods for integrating SFT and RLVR for reasoning can be grouped into two categories. First, *objective-level combination* integrates SFT and RL by weighting or scheduling their objectives, ranging from interleaved recipes (e.g., alternating RL and SFT when RL stalls) (Ma et al., 2025) to single-stage multi-objective training with adaptive reweighting or gating (Zhang et al., 2025b; Chen et al., 2025a; Fu et al., 2025). Second, *data-augmented RL* incorporates SFT data as off-policy trajectories within the RL objective (Yan et al., 2025), typically weighted by importance-sampling ratios to mitigate distribution mismatch; however, it often underperforms objective-level combination in practice (Zhang et al., 2025b; Chen et al., 2025a). Despite their practical success, objective-level combinations rarely characterize how the two learning signals interact. As shown in Figure 1, we find that a simple combination of SFT and RL updates even decrease the reward of RL, indicating that not all supervised updates are helpful for reward optimization.

*Corresponding to: Xueting Han and Kam-Fai Wong.

In this work, we address the challenges by formulating the integration as a meta-learning problem, BRIDGE, which treats SFT as an upper-level *teacher* and RL as a lower-level *student*. By modeling the hierarchical structure—with the SFT objective explicitly conditioned on the RL solution—we enable SFT to provide *targeted* guidance that directly supports RL optimization, rather than updating SFT and RL independently and balancing them via heuristics.

A direct bilevel solver typically requires second-order derivatives (Finn et al., 2017; Hu et al., 2023), which are prohibitive at LLM scale. To reduce computational overhead, we adopt a first-order, penalty-based relaxation. Concretely, the lower level updates mix two objectives, while the upper level updates maximizes the *reward-gain*—the reward difference of mix SFT-RL training over RL training alone. We further separate these roles across parameter types: the lower level updates the LLM parameters (the student), whereas the upper level updates a newly initialized low-rank adapter (LoRA; Hu et al., 2021) as *meta-parameters* (the teacher). This design yields an efficient, scalable algorithm suitable for large-scale training.

To validate the effectiveness of our approach, we conduct comprehensive experiments with three LLMs of varying scales on five diverse math reasoning benchmarks. Results demonstrate that BRIDGE consistently outperforms all baselines, including SFT, RL, two-stage pipelines, and recent hybrid training methods. Notably, BRIDGE requires less wall-clock training time than the two-stage method while delivering superior performance, highlighting its practical efficiency. Furthermore, extensive ablation studies confirm the necessity of the bilevel design and demonstrate the robustness of our method to hyperparameter variations across different model sizes and task difficulties. These improvements confirm the benefits of coupling SFT and RL through bilevel optimization, enabling the model to selectively learn from supervised signals that contribute to reward maximization.

2 Background and Preliminaries

We review two prevalent post-training paradigms for reasoning in LLMs—supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR)—and discuss a widely used hybrid objective. We show that combining the two objectives does not necessarily improve reward and can sometimes lead to lower reward.

2.1 Fine-tuning Methods for Reasoning Models

Let $\pi_\theta(y \mid x)$ denote a language model with parameters θ that defines a conditional distribution over output sequences y given an input prompt x. We assume a reasoning dataset $\mathcal{D} = \{(x, r, y)\}$, where x is an input question, y is a verifiable target answer, and r is an expert reasoning trace. During training, SFT and RLVR operate on different views of the same dataset: SFT uses (x, r, y), while RLVR uses (x, y) to compute rewards.

SFT. Given a question x, SFT maximizes the log-likelihood of the expert trace r and final answer y jointly:

$$J_{\mathrm{SFT}}(\theta) := \mathbb{E}_{(x,y,r)\sim\mathcal{D}}\big[\log \pi_\theta(r, y \mid x)\big]. \tag{1}$$

This encourages LLMs to imitate expert reasoning patterns and to produce the corresponding answer.

RLVR. RLVR does not require annotated reasoning traces. Given a question x, the policy samples a reasoning trace $\hat{r}$ and an answer $\hat{y}$, and receives a reward based on answer correctness. A standard KL-regularized objective is:

$$\begin{aligned} J_{\mathrm{RL}}(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D},\ (\hat{r},\hat{y})\sim\pi_\theta(\cdot|x)}\big[R(\hat{y}, y)\big] \\ - \beta_{\mathrm{KL}}\, \mathbb{E}_{x\sim\mathcal{D}}\Big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\big)\Big]. \end{aligned} \tag{2}$$

where π_{ref} is a fixed reference policy and $\beta_{\mathrm{KL}} \geq 0$ controls the strength of KL regularization. Here $R(\hat{y}, y)$ is computed by a deterministic, rule-based verifier (e.g., code execution or regular-expression matching). In practice, J_{RL} is optimized using policy-gradient variants such as GRPO (DeepSeek-AI et al., 2025) and DAPO (Yu et al., 2025).

Two-stage pipeline. The prevailing paradigm (DeepSeek-AI et al., 2025) adopts a sequential protocol. The model is first optimized via J_{SFT} to acquire foundational reasoning patterns (SFT warm-up),

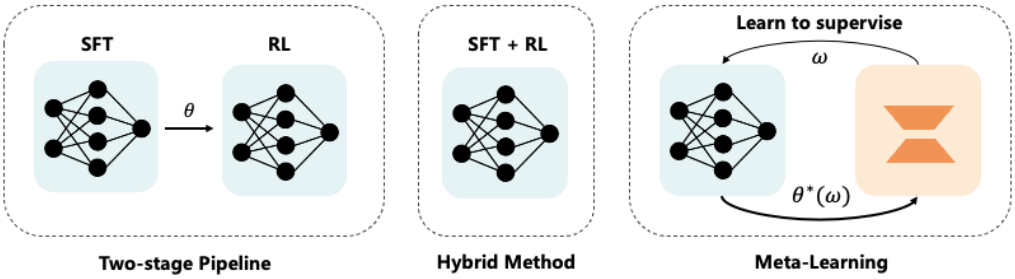

Figure 2: Comparison of three training paradigms. **Left:** The two-stage pipeline first performs SFT then RL, with unidirectional knowledge transfer. **Middle:** Single-stage hybrid training combines SFT and RL objectives via weighting or scheduling on shared parameters, without modeling their interaction. **Right:** Our meta-learning approach introduces a teacher module (w) that learns to supervise the student LLM ($\theta^*(w)$), enabling bidirectional adaptation between the two objectives.

thereby providing a high-quality initialization for the subsequent maximization of J_{RL}. This approach mitigates the exploration difficulties inherent in training from scratch but decouples the supervised signal from the reward optimization phase.

Single-stage hybrid methods. Current methods often integrate SFT and RLVR by optimizing a weighted sum:

$$J_{\mathrm{hyb}}(\theta) := J_{\mathrm{RL}}(\theta) + \mu\, J_{\mathrm{SFT}}(\theta), \qquad \mu \geq 0, \tag{3}$$

where μ trades off reward maximization and imitation learning. In practice, μ is typically set by heuristics (e.g., fixed values, decay schedules (Zhang et al., 2025b), or adaptive rules based on entropy or gradient statistics (Fu et al., 2025)).

2.2 Analysis of Fine-Tuning Methods

To understand the interaction between supervised and reinforcement learning signals, we conduct a preliminary evaluation of three canonical fine-tuning paradigms on mathematics problems (Grades 3–5). We compare **RL** (training from scratch), **Two-Stage (SFT→RL)** (SFT followed by RL), and **Hybrid (SFT + RL)** (multi-task learning with a fixed scalar weight $\lambda = 1$). Figure 1 illustrates the mean reward trajectories throughout training.

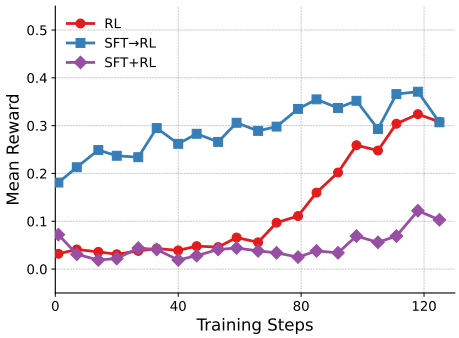

Figure 1: Reward comparison of three different types of training methods.

The results highlight limitations in current methodologies. While **RL** (Red) eventually improves, it suffers from training inefficiency due to the lack of prior knowledge, requiring extensive exploration to discover high-reward regions. Conversely, the **SFT→RL** approach (Blue) leverages SFT for a strong initialization. However, this advantage is primarily static. By discarding the supervised signal after the warm-up, the subsequent training phase reverts to unguided exploration. Consequently, the benefit of the SFT initialization is most pronounced in the early stages but diminishes as the model struggles to navigate the complex reasoning landscape without ongoing directional guidance. Notably, the naive **SFT+RL** method (Purple) yields the worst performance, significantly lagging behind even RL.

This suggests that not all supervised updates are beneficial for reward maximization; thus, we ask a natural question: *How can we dynamically extract the useful components of the supervised signal that actively facilitate the optimization of the RL reward?*

3 METHODOLOGY

We propose a meta-learning method that models the teacher-student relationship between SFT and RL. We first introduce the formulation, then present the learning algorithm.

3.1 BRIDGE: META-LEARNING FOR SFT AND RL

Given a reasoning dataset $\mathcal{D}$ and an LLM parameterized by θ (defined in Section 2.1), our objective is to integrate the SFT objective (Eq.equation 1) with the RL objective (Eq.equation 2) such that SFT updates facilitate reward optimization in RL. We treat SFT as the *teacher*, since it has access to expert reasoning traces, and RL as the *student*, since it relies on policy exploration to discover high-reward traces. We model their relationship through the following bi-level optimization:

$$\begin{aligned} \max_{w} \quad & J_{\mathrm{SFT}}(w, \theta^*(w)), \\ \text{s.t.} \quad & \theta^*(w) := \arg\max_{\theta} J_{\mathrm{RL}}(\theta, w). \end{aligned} \tag{4}$$

where θ denotes the student LLM parameters and w denotes the teacher's meta-parameters, instantiated as a lightweight LoRA module.

This formulation has a hierarchical structure inspired by Stackelberg games: SFT serves as the upper-level leader with access to RL's solution, providing supervision that improves reward optimization, while RL acts as the lower-level follower, optimizing the policy under guidance from SFT. This coupling enables the two objectives to cooperate dynamically, each adapting to the other's feedback.

Figure 2 contrasts our approach with prior paradigms. The two-stage pipeline and single-stage hybrid methods both apply two objectives to the same LLM either sequentially or simultaneously, without modeling their interaction. In contrast, our method introduces a separate teacher module w (implemented as LoRA), which learns to supervise the student LLM. By conditioning each component on the other's parameters, this design enables tighter coordination between the two learning signals.

3.2 LEARNING ALGORITHM VIA PENALTY RELAXATION

To efficiently solve the bi-level problem in Eq. equation 4 at LLM scale, we employ penalty-based methods (Shen & Chen, 2023; Shen et al., 2025) that avoid expensive second-order computations. We first reformulate Eq. equation 4 as a single-level problem, then apply first-order optimization.

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

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

which satisfies $p(w, \theta) = 0$ if and only if $\theta \in \arg\max_{\theta'} J_{\mathrm{RL}}(\theta', w)$. For $\lambda \in (0, 1)$, consider the following penalized objective:

$$\max_{\theta, w} \ \mathcal{L}(\theta, w) := (1 - \lambda) J_{\mathrm{SFT}}(\theta, w) - \lambda p(w, \theta). \tag{6}$$

Update for θ (student). Since $\max_{\theta'} J_{\mathrm{RL}}(\theta', w)$ does not depend on θ, the update of LLM student is as follows:

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

This yields a convex combination of SFT and RL gradients.

Update for w (teacher). For the update of teacher's meta-parameters, we use Danskin's theorem. Assuming $J_{\mathrm{RL}}(\cdot, w)$ satisfies the required regularity conditions (e.g., $J_{\mathrm{RL}}(\theta, w)$ is differentiable in w and $\arg\max_{\theta'} J_{\mathrm{RL}}(\theta', w)$ is non-empty), we have:

$$\nabla_w \max_{\theta'} J_{\mathrm{RL}}(\theta', w) = \nabla_w J_{\mathrm{RL}}(\theta^*(w), w), \tag{8}$$

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: $\hat{\theta} = \theta + \alpha \nabla_\theta J_{\mathrm{RL}}(\theta, w)$, yielding the approximate gradient update for w:

$$\begin{aligned} \nabla_w \mathcal{L}_\lambda(\theta^k, w^k) = (1 - \lambda) \nabla_w J_{\mathrm{SFT}}(\theta^k, w^k) \\ + \lambda \left[\nabla_w J_{\mathrm{RL}}(\theta, w) - \nabla_w J_{\mathrm{RL}}(\hat{\theta}, w) \right] \end{aligned} \tag{9}$$

Algorithm 1: Learning Algorithm of BRIDGE

Initialize student parameters θ^0, teacher parameters w^0, and auxiliary parameters $\hat{\theta}^0 := \theta^0$; reasoning dataset $\mathcal{D}$, penalty coefficient λ, iterations K, learning rates α, β;
for $k = 0$ to $K - 1$ **do**
  Sample mini-batches $\mathcal{B}_{\mathrm{SFT}} \sim \mathcal{D}$ and $\mathcal{B}_{\mathrm{RL}} \sim \mathcal{D}$
  // Compute base objectives
  Compute $J_{\mathrm{SFT}}(\theta^k, w^k)$ on $\mathcal{B}_{\mathrm{SFT}}$
  Compute $J_{\mathrm{RL}}(\theta^k, w^k)$ and $J_{\mathrm{RL}}(\hat{\theta}^k, w^k)$ on $\mathcal{B}_{\mathrm{RL}}$
  // Define composite objectives
  $J_{\mathrm{Joint}}(\theta^k, w^k) = (1 - \lambda)\, J_{\mathrm{SFT}}(\theta^k, w^k) + \lambda\, J_{\mathrm{RL}}(\theta^k, w^k)$
  Compute $J_{\mathrm{meta}}(\theta^k, w^k)$ according to Eq. equation 10
  // Update student via joint objective
  $\theta^{k+1} \leftarrow \theta^k + \alpha\, \nabla_\theta J_{\mathrm{Joint}}(\theta^k, w^k)$
  // Update auxiliary parameters via RL objective
  $\hat{\theta}^{k+1} \leftarrow \hat{\theta}^k + \alpha\, \nabla_{\hat{\theta}} J_{\mathrm{RL}}(\hat{\theta}^k, w^k)$
  // Update teacher to maximize cooperative gain
  $w^{k+1} \leftarrow w^k + \beta\, \nabla_w J_{\mathrm{meta}}(\theta^k, w^k)$
end for

Interpretation of teacher's behaviors The update rule for the meta-parameter w (Eq. equation 9) can be interpreted as gradient ascent on the surrogate objective

$$\begin{aligned} J_{\mathrm{meta}}(\theta, w) = (1 - \lambda) J_{\mathrm{SFT}}(\theta, w) \\ + \lambda \big[J_{\mathrm{RL}}(\theta, w) - J_{\mathrm{RL}}(\hat{\theta}, w) \big] \end{aligned} \tag{10}$$

where the first term preserves supervision from expert trajectories, and the second term is a *cooperative gain* signal measuring how much the joint SFT-RL model (using θ) improves the RL objective relative to an RL-only one (using $\hat{\theta}$). Maximizing J_{meta} therefore encourages the teacher parameters w to shape supervision based on its *utility for reward optimization*, rather than implicitly treating all supervised updates as uniformly beneficial.

Algorithm 1 presents the learning procedure for BRIDGE. At each iteration, we sample SFT and RL mini-batches; update base parameters θ via the joint objective; update auxiliary parameters $\hat{\theta}$ via pure RL; and optimize LoRA parameters w to maximize cooperative gain—the improvement of the joint objective over pure RL. This process enables SFT to meta-learn how to guide RL's optimization.

4 Experiment

4.1 Settings

Datasets. Following the setup of SimpleRL (Zeng et al., 2025), we use the MATH dataset (Hendrycks et al., 2021a) for RL training, and train on the hard split, which contains 8.5K problems with difficulty levels ranging from 3 to 5. For intermediate reasoning traces, we distilled from DeepSeek-R1 (DeepSeek-AI et al., 2025). For evaluation, we adopt five mathematical reasoning benchmarks: MATH500 (Hendrycks et al., 2021a), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and two recent competition-level datasets—AMC 2023 and AIME 2024.

Models. To demonstrate the generality of our approach, we experiment with three LLMs: Qwen2.5-3B (Yang et al., 2024), Llama-3.2-3B-Instruct (Grattafiori et al., 2024), and Qwen3-8B-Base (Yang et al., 2025). All models use prompt formats consistent with SimpleRL (Zeng et al., 2025).

Baselines. We compare BRIDGE against a representative and comprehensive set of baselines, spanning widely used standard training recipes and major families of recent single-stage hybrid training methods. Specifically, we include: (i) **Original Model**: the base or instruction-tuned backbone without reasoning-specific training; (ii) **SFT**: imitation learning on expert reasoning traces; (iii) **RL** (Zeng et al., 2025): GRPO applied directly to the backbone without SFT warm-up; (iv) **SFT→RL (two-stage)** (DeepSeek-AI et al., 2025): the sequential pipeline that first performs SFT and then applies RL with decoupled objectives. We further compare against **single-stage hybrids** that integrate SFT and RL within a unified training stage: (v) **SFT+RL**: a multitask baseline that

Table 1: Performance of BRIDGE compared to baselines on Qwen2.5-3B across five math benchmarks

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	64.4	26.5	27.0	3.3	40.0	32.2
SFT→RL	66.0	24.3	26.8	9.0	35.0	32.2
SFT+RL	55.6	20.6	25.0	3.3	42.5	29.4
LUFFY	65.2	23.5	27.3	3.3	42.5	32.4
SRFT	62.6	22.1	24.4	9.0	37.5	31.1
CHORD	66.0	23.2	25.9	6.7	40.5	32.5
BRIDGE	66.2	23.9	28.9	13.3	47.5	36.0

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

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	48.6	15.1	17.8	10.0	17.5	21.8
SFT→RL	45.0	11.8	12.0	3.3	22.5	18.9
SFT+RL	45.8	13.6	17.3	3.3	20.0	20.0
LUFFY	49.0	14.0	17.1	6.7	22.5	21.9
SRFT	45.4	13.6	15.4	3.3	17.5	19.0
CHORD	46.0	14.3	17.9	3.3	22.5	20.8
BRIDGE	51.8	15.1	19.3	10.0	27.5	24.7

directly combines and optimizes the SFT loss and the RL objective; (vi) **LUFFY** (Yan et al., 2025) (*data-augmented RL*): incorporates demonstration-style information into RL rollouts to stabilize and improve policy optimization; (vii) **CHORD** (Zhang et al., 2025b): balances supervision and RL by dynamically weighting the objectives, including token-wise reweighting; (viii) **SRFT** (Fu et al., 2025): reduces interference between objectives via entropy-aware weighting and clipping.

We use Verl (Sheng et al., 2024) for RL training; full settings are in Appendix A.

4.2 Main Results

Overall performance. Tables 1–3 show that BRIDGE achieves the highest average accuracy across all three LLMs: 36.0% on Qwen2.5-3B, 24.7% on Llama-3.2-3B-Instruct, and 49.9% on Qwen3-8B-Base. These correspond to absolute gains of 3.5, 2.8, and 4.0 points over the respective strongest baselines. Notably, BRIDGE is the only method that consistently surpasses standard recipes (e.g., RL and the two-stage SFT→RL pipeline) across all settings, validating the efficacy of our meta formulation in extracting beneficial supervision without hindering reward optimization.

Comparisons across baseline families. Grouping baselines clarifies the role of different integration strategies. Among standard recipes, RL generally outperforms SFT on final accuracy, though less efficient. The two-stage pipeline (SFT→RL) proves a strong baseline, yielding consistent improvements on the Qwen family at both 3B and 8B. For single-stage hybrids, naive objective-level mixing (SFT+RL) often produces performance between SFT and RL, indicating that simply summing objectives does not reliably improve upon RL—instead yielding a compromise solution. More sophisticated hybrids (e.g., CHORD, SRFT) employ heuristic weighting or scheduling of the two objectives, partially mitigating the adverse effects of supervision on RL exploration. However, these methods still leave a consistent margin to BRIDGE, which explicitly optimizes supervision to improve downstream reward gains rather than assuming that supervised updates are uniformly beneficial.

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

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	76.2	36.0	42.4	10.0	50.0	42.9
SFT→RL	80.4	38.2	39.6	16.7	52.5	45.5
SFT+RL	72.2	34.2	39.2	10.0	45.0	40.1
LUFFY	75.4	36.4	43.1	10.0	55.0	44.0
SRFT	72.2	32.4	40.0	6.7	47.5	39.8
CHORD	76.6	37.5	42.2	13.3	60.0	45.9
BRIDGE	79.0	39.7	44.0	16.7	70.0	49.9

Performance across model families and scales. Baseline rankings vary across backbones, whereas BRIDGE remains consistently strong. For the strongest standard recipe, the two-stage SFT→RL pipeline is competitive on Qwen 3B and 8B but underperforms RL on Llama-3B-Instruct, indicating that this fixed recipe, though strong, is not uniformly reliable across model families. Among hybrid baselines, CHORD is comparatively stronger on the Qwen backbones, while LUFFY is slightly stronger on Llama-3.2-3B-Instruct. Despite this variability, BRIDGE improves over the strongest baseline across different LLM families and preserves its advantage when scaling from 3B to 8B parameters. The gap between BRIDGE and the best hybrid widens from 3.5 points at 3B to 4.0 points at 8B, suggesting that principled SFT–RL coordination compounds with model capacity.

Generalization to challenging benchmarks. A critical limitation of SFT-based baselines is their tendency to plateau on challenging tasks. As shown in Table 3, while the two-stage pipeline (SFT→RL) improves performance on the standard benchmarks like MATH500, it underperforms RL in generalization on the more challenging OlympiadBench and Minerva Math. This indicates that supervised warm-up can restrict the policy to local optima. In contrast, BRIDGE preserves RL's superior generalization ability on more challenging tasks, achieving the highest scores on OlympiadBench and AIME24 across all models. This suggests that our meta objective encourages the model to leverage SFT only when it aids RL learning, ignoring supervised signals that lead to rote memorization and hinder generalization.

5 Analysis and Ablation Study

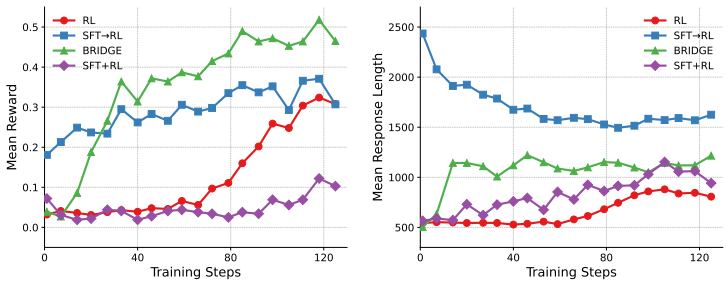

Figure 3: Mean reward and response length on Qwen2.5-3B.

Training Dynamics. We analyze the dynamics of mean reward and response length during training for RL, SFT→RL, BRIDGE, and SFT+RL on Qwen2.5-3B. As shown in Figure 3, the four methods exhibit markedly different patterns. RL suffers from online RL's sample inefficiency, showing slow growth in both response length and reward. Cold-start (SFT→RL) begins with extremely long responses due to SFT warm-up, *causing slow training* (see in Table 4), followed by a sharp decline and gradual recovery. Despite starting with higher rewards, Cold-start's second-phase RL lacks proper guidance, resulting in final rewards similar to RL-Zero. SFT+RL mixes two objectives directly without proper coordination, which slows down the RL training — both the mean reward and response length improve very slowly across the entire training. 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.

Table 4: Cost-benefit analysis on Qwen2.5-3B and Qwen3-8B.

Metric	Qwen 2.5-3B			Qwen 3-8B-Base		
	RL	SFT→RL	BRIDGE	RL	SFT→RL	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

Training Efficiency. 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 4, two-stage pipeline (SFT→RL) requires nearly 2x the training time of RL, despite the short SFT stage. This overhead stems from long sequence lengths induced by the SFT stage (Fig. 3). BRIDGE achieved 44% and 14% time savings compared to the two-stage pipeline 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.

Ablations on the meta-objective, penalty coefficient, and LoRA configurations are in Appendix C.

6 Related Work

Integrating SFT and RL. Recent work has unified supervised fine-tuning (SFT) and reinforcement learning (RL) within a single stage, moving beyond the decoupled "SFT then RL" paradigm. Integration methods typically fall into two types. First, *objective-level combinations* mix SFT and RL losses, using fixed or adaptive weighting schedules (Ma et al., 2025; Zhang et al., 2025b; Chen et al., 2025a; Fu et al., 2025). CHORD and SRFT exemplify this line via token-level reweighting or entropy-aware mixing. Second, *data-augmented RL* injects SFT demonstrations as off-policy data (e.g., LUFFY (Yan et al., 2025)), but often struggles with distribution mismatch. While effective in practice, these approaches typically rely on heuristic coupling and do not characterize when supervision benefits reward learning. Our method, BRIDGE, reframes integration as a bilevel problem that meta-learns *how* SFT should assist RL to maximize total reward gain.

Bilevel Optimization in LLMs. Bilevel optimization (BLO) provides a principled framework for hierarchical learning, with early roots in Stackelberg games. BLO solvers are broadly either implicit-gradient based (Hong et al., 2020; Khanduri et al., 2021; Shen & Chen, 2022; Xiao et al., 2023)—theoretically sound but computationally limited—or penalty-based relaxations (Shen & Chen, 2023; Kwon et al., 2023; Shen et al., 2024; Lu, 2024), which scale to large models via first-order gradients. Recent applications to LLMs include data selection (Lin et al., 2024; Shen et al., 2025), inverse RL (Li et al., 2024), and meta-learning (Choe et al., 2023; Shirkavand et al., 2025). To our knowledge, this is the first to frame reasoning-model training as a BLO problem. Explicitly modeling SFT–RL interaction enables adaptive supervision that improves reward optimization.

Extended related work is provided in Appendix B.

7 Conclusion

This work investigates how to effectively integrate supervised fine-tuning and reinforcement learning to improve the reasoning capabilities of LLMs. We begin by analyzing widely used training paradigms and identify a key limitation of existing baselines: two-stage pipelines decouple SFT from the reward-optimization phase, while naive single-stage objective mixing can be counterproductive because supervised updates are not uniformly beneficial for reward optimization. To address this, we 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 cooperation between the two learning paradigms. Empirical results on five mathematical reasoning benchmarks demonstrate that our method consistently outperforms strong baselines in both accuracy and training efficiency. Furthermore, extensive ablation studies confirm the necessity of the bilevel cooperation signal and the robustness of the method to hyperparameter variations. Overall, this work demonstrates that learning useful supervision is a viable and effective strategy for integrating SFT and RL, and that bilevel optimization offers an effective foundation for advancing reasoning-centric post-training methods.

REFERENCES

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Jack Chen, Fazhong Liu, Naruto Liu, Yuhan Luo, Erqu Qin, Harry Zheng, Tian Dong, Haojin Zhu, Yan Meng, and Xiao Wang. Step-wise adaptive integration of supervised fine-tuning and reinforcement learning for task-specific llms, 2025a. URL https://arxiv.org/abs/2505.13026.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=EMHED4WTHT.

Liang Chen, Li Shen, Yang Deng, Xiaoyan Zhao, Bin Liang, and Kam-Fai Wong. PEARL: Towards permutation-resilient LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=txoJvjfI9w.

Liang Chen, Xueting Han, Qizhou Wang, Bo Han, Jing Bai, Hinrich Schuetze, and Kam-Fai Wong. EEPO: Exploration-enhanced policy optimization via sample-then-forget. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ObF4WIMkY6.

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.

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.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=dYur3yabMj.

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.

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

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.

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.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning, 2025. URL https://arxiv.org/abs/2506.19767.

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 Hosseini, 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.

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.

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.

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*, 2021a.

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*, 2021b.

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.

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.

Nathan Hu, Eric Mitchell, Christopher D. Manning, and Chelsea Finn. Meta-learning online adaptation of language models, 2023. URL https://arxiv.org/abs/2305.15076.

Yiding Jiang, J Zico Kolter, and Roberta Raileanu. On the importance of exploration for generalization in reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=y5duN2j9s6.

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.

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.

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.

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.

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.

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

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Bin Cui, and Wentao Zhang. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions, 2025. URL https://arxiv.org/abs/2506.07527.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

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

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

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.

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.

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.

Meng Song. Good actions succeed, bad actions generalize: A case study on why rl generalizes better, 2025. URL https://arxiv.org/abs/2503.15693.

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

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL https://arxiv.org/abs/2504.14945.

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.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

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.

Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Exploring tool-using language models with reinforced reasoning, 2025a. URL https://arxiv.org/abs/2505.00024.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting, 2025b. URL https://arxiv.org/abs/2508.11408.

APPENDIX

A IMPLEMENTATION DETAILS

All models are trained using the VERL framework (Sheng et al., 2024), employing the GRPO algorithm. We use a batch size of 128, a mini-batch size of 64, a learning rate of 5×10^{-7}, and 8 rollouts, training for 2 epochs. The KL loss and entropy loss coefficient are set to 1×10^{-4} and 1×10^{-5}, respectively. The maximum response length varies by model: up to 4K tokens for Qwen2.5-3B, and up to 6K tokens for both LLaMA-3.2-3B-Instruct and Qwen3-8B-Base. During evaluation, we use greedy decoding to compute pass@1 accuracy. All experiments are conducted on compute clusters equipped with NVIDIA A100 GPUs and AMD MI300 GPUs.

B RELATED WORK

Post-training for reasoning. Post-training typically follows two paradigms: supervised fine-tuning (SFT) and reinforcement learning (RL). RL has been widely used to improve the capabilities of language models (Bai et al., 2022; Rafailov et al., 2023) and to align model outputs with human values across diverse scenarios (Chen et al., 2025c;b), particularly through reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022). More recently, reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; DeepSeek-AI et al., 2025) has replaced subjective preference feedback with automatically verifiable, rule-based rewards. Recent studies analyze the trade-off between the two paradigms; for example, Chu et al. (2025) compare SFT and RL for reasoning tasks and find that RL generalizes significantly better, whereas SFT is prone to overfitting. A common practice therefore adopts a two-stage SFT→RL pipeline, where SFT is often used as a warm-up stage before RL. Within this recipe, the choice of supervised traces matters: SimpleRL (Zeng et al., 2025) observes that fine-tuning on short-CoT datasets can harm reasoning ability, while He et al. (2025) find that long-CoT distilled data can improve the reasoning performance of smaller models when

used as a warm-up stage before RL training. However, the advantage of the two-stage pipeline over pure RL is inconsistent, motivating tighter integration of the two learning signals.

Integrating SFT and RL. Recent efforts move beyond the decoupled "SFT then RL" recipe by mixing two objectives within one stage. Existing integration methods largely fall into two families. *Objective-level combination* mixes SFT and RL objectives via fixed or scheduled weights, including interleaved recipes (Ma et al., 2025) and single-stage hybrid training with adaptive reweighting or gating (Zhang et al., 2025b; Chen et al., 2025a; Fu et al., 2025). Representative examples include CHORD, which stabilizes training with global and token-wise reweighting (Zhang et al., 2025b), and SRFT, which mitigates interference via entropy-aware weighting and clipping (Fu et al., 2025). *Data-augmented RL* instead injects SFT demonstrations as off-policy guidance within RL (e.g., LUFFY) (Yan et al., 2025), often requiring additional mechanisms to handle distribution mismatch and data pairing constraints. Despite empirical progress, most hybrids couple the signals heuristically and rarely characterize *when* a supervised update is actually beneficial for reward optimization. BRIDGE instead treats SFT–RL cooperation as a bilevel problem, meta-adapting the supervision to maximize the reward gain of joint training over RL alone and yields larger and more robust improvements than simple loss mixing. It 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 hierarchical learning problems, originating from Stackelberg leader-follower games. Two major classes of methods have been developed to solve BLO problems. Implicit gradient methods (Hong et al., 2020; Khanduri et al., 2021; Shen & Chen, 2022; Xiao et al., 2023) 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 (Shen & Chen, 2023; Kwon et al., 2023; Shen et al., 2024; Lu, 2024) 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 (Lin et al., 2024; Shen et al., 2025), inverse reinforcement learning (Li et al., 2024), and meta-learning (Choe et al., 2023; Shirkavand et al., 2025). To the best of our knowledge, our work is the first to cast reasoning-oriented LLM training as bilevel optimization, introducing a novel augmented model architecture for modeling and solving this problem. This provides a principled framework for integrating supervised and reinforcement learning, where SFT actively assists RL optimization rather than merely serving as warmup.

C Ablation Study

Impact of the Meta-Objective ($\mathcal{J}$meta). We isolate the contribution of the meta-objective $\mathcal{J}_{\text{meta}}$ (Eq. 10) by disabling the upper-level update to assess the necessity of the bilevel formulation. As shown in Table5, removing this term reduces BRIDGE to a naive multi-task learning between SFT and RL with fixed weighting, resulting in a substantial 9.6-point drop in average accuracy on Qwen3-8B-Base. This degradation confirms that simply combining objectives is insufficient; $\mathcal{J}_{\text{meta}}$ is critical for aligning supervised updates with the ultimate reinforcement learning goal, ensuring that SFT provides targeted assistance.

Table 5: Ablation of $\mathcal{J}_{\text{meta}}$ on Qwen3-8B-Base.

Configuration	Average accuracy
BRIDGE	49.9
- w/o $\mathcal{J}_{\text{meta}}$	40.3

Balancing Supervision and Exploration (λ). Finally, we analyze the weighting parameter λ, which controls the interpolation between the SFT and RL objectives in our penalty formulation (Equation 6). We evaluate average accuracy across five benchmarks for $\lambda \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$. Table 6 reports average accuracy on Qwen3-8B-Base. At $\lambda = 0$, the formulation reduces to pure SFT; at $\lambda = 1$, it reduces to pure RL. Intermediate values perform substantially better, with $\lambda = 0.5$ achieving peak

performance. These results suggest that BRIDGE benefits from a balanced coupling of supervised and reward-driven learning, rather than relying exclusively on either signal.

Table 6: Sensitivity to λ on Qwen3-8B-Base.

λ	**0.0**	**0.3**	**0.5**	**0.7**	**1.0**
Avg.	37.6	47.2	49.9	49.0	42.9

Robustness to LoRA Configuration. Finally, to verify that the gains of BRIDGE are not artifacts of a specific parameter-efficient fine-tuning configuration, we ablate the LoRA rank r and scaling factor α on Qwen3-8B-Base while keeping all other training settings fixed. Table 7 demonstrates that performance is negligible variance across (r, α) choices. This stability suggests that BRIDGE is robust to low-rank hyperparameter settings and that the improvements stem from the objective function itself.

Table 7: LoRA sensitivity ablation on Qwen3-8B-Base.

r/α	MATH500	Minerva	Olym.	AIME24	AMC23	**Avg.**
32/16	79.0	39.7	44.0	16.7	70.0	49.9
16/32	79.0	38.6	44.0	16.0	70.0	49.5