SHARED-TRAJECTORY META-LEARNING FOR JOINT SFT AND RL OPTIMIZATION IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) and reinforcement learning (RL) are two synergistic paradigms for large language models, typically applied in a SFT then RL pipeline. However, the staged schedule can cause RL to overwrite the gains acquired from SFT. Jointly optimizing both is challenging because their gradients may conflict, and existing methods usually consider which signal should dominate while overlooking their interaction. We propose a meta-learning framework that places SFT and RL on a shared inner trajectory and applies a Reptile-style outer update. This implicitly encourages alignment across their gradients, guiding the model towards the regions where progress on one objective is less likely to hinder the other. Our method outperforms SFT, RL, and hybrid methods across multiple backbones.

1 INTRODUCTION

Recent advances in scaling model size, training data, and computation have greatly improved the abilities of large language models (LLMs) (Kaplan et al., 2020). Beyond pre-training, post-training has emerged as a critical stage for further improving model behavior and reasoning capabilities.

In post-training, supervised fine-tuning (SFT) and reinforcement learning (RL) (Ouyang et al., 2022; Guo et al., 2025) are two widely used paradigms. They play complementary roles in several perspectives. From the data side, SFT relies on high-quality labeled trajectories, whereas RL can leverage outcome-based rewards without requiring labels. From functional side, SFT can inject expert priors for imitation, while RL enables exploration that leads to strategies beyond human solutions.

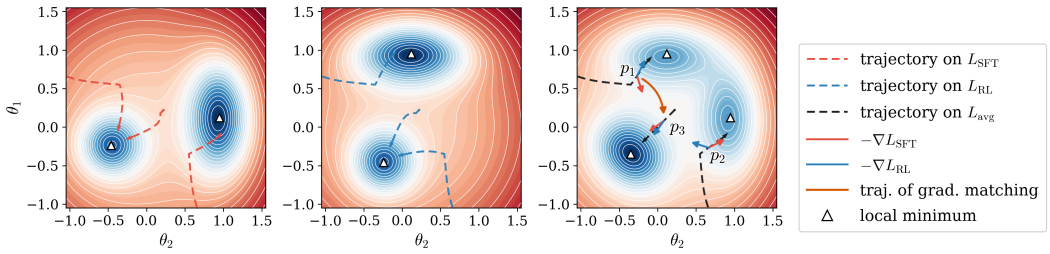
Given this complementarity, a common paradigm is to first perform SFT and then apply RL (Shao et al., 2024; Yang et al., 2024; Yu et al., 2026), with SFT providing an effective initialization that facilitates the subsequent RL training. However, staged training has a well-known failure mode that the model tends to become overly biased toward the last objective (Shi et al., 2025; Fu et al., 2026; Yuan et al., 2025). While RL improves the model through reward exploration, it can also overwrite knowledge acquired during SFT, and further SFT updates may weaken abilities discovered through RL. This makes it difficult to jointly retain the benefits of imitation and exploration.

Jointly leveraging heterogeneous training sources is a commonly discussed topic in machine learning. In semi-supervised learning (Van Engelen & Hoos, 2020), the models are often warmed up on labeled data and subsequently optimized using both labeled and unlabeled data. In transfer learning (Tan et al., 2018; Zhou et al., 2022), models are first trained on one data distribution and then adapt to others, while continuing to use the original data. These suggest a natural alternative to the staged SFT$\rightarrow$RL paradigm: rather than optimizing the two objectives sequentially, one could jointly leverage their complementary training signals during optimization.

However, jointly optimizing SFT and RL is more challenging than combining multiple data sources under similar objectives. SFT and RL impose distinct optimization objectives that provide different learning signals. Their gradients may align on some behaviors but conflict on others (Zhao et al., 2026). When pointing in different directions, naively combining may cause one signal to dominate or lead to mutual interference. The central challenge is thus not merely how to expose the model to both types of training signals, but how to manage their interaction through optimization.

Existing methods for jointly leveraging SFT and RL typically address this challenge by prescribing how the two signals are combined (Yan et al., 2026; Fu et al., 2026; Ma et al., 2026). They either

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Loss surface of L_{SFT} (b) Loss surface of L_{RL} (c) Average loss surface

Figure 1: Gradient interaction on complementary synthetic loss surfaces. (a)(b): Same initialization under L_{SFT} and L_{RL} converge to different basins. The shared basin in the lower-left corner provides small loss for both objectives and is therefore preferable to the two objective-specific basins in the upper and right regions. (c): On average surface $L_{\text{avg}} = \frac{1}{2}(L_{\text{SFT}} + L_{\text{RL}})$, moving in the direction of larger gradient inner product makes it more likely to enter the shared basin.

inject supervised signals into RL rollouts or mix the SFT and RL losses, with binary scheduling as a special case of loss mixing. While enabling the two signals to coexist within training, the relative contribution of the two signals is largely determined by a predefined mechanism, where increasing the influence of one signal can suppress that of the other.

So how can we determine a balance interaction between SFT and RL? One way to access their interplay is through the inner product of gradients (Nichol et al., 2018). The negative product indicates a strong conflict between two objectives, implying that updating one may hinder progress on the other. In contrast, the positive product suggests greater consistency between them, so updating one objective is less likely to damage the other. Therefore, instead of modifying by manually designed rules, one can seek parameter regions in which the gradients of SFT and RL become naturally aligned.

Motivated by this, we propose meta-learning framework for joint SFT and RL training called *shared-trajectory meta-learning*. As shown in Figure 1, rather than explicitly modifying conflicting gradients or prescribing the relative dominance of SFT and RL, our method learns parameter updates that steer the model towards regions with improved cross-objective gradient compatibility. This enables the model to jointly retain the complementary benefits of SFT and RL, therefore achieving a better balance between imitation and exploration. Our contributions is summarized as follows:

1. We extend first-order meta-learning algorithm from within-objective to cross-objective gradient matching (Section 2), to address conflict signals in joint SFT-RL training.
2. We provide a derivation of our claim and revisit general meta-learning from other perspectives, including ensemble and flat minima. (Section 3)
3. We evaluate on math and out-of-domain benchmarks to show that our method outperforms SFT-only, RL-only, and hybrid methods across multiple LLM backbones. (Section 4)

2 Method

2.1 Preliminaries

We consider an autoregressive language model parameterized by θ, which defines a policy π_θ. Given a prompt x, the model generates a response $y = (y_1, \ldots, y_T)$ token by token:

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{1}$$

In this work, we focus on two complementary post-training paradigms: supervised fine-tuning (SFT) and reinforcement learning (RL).

Supervised fine-tuning. SFT (Ouyang et al., 2022) updates the policy on a demonstration dataset $\mathcal{D}_{\text{SFT}} = \{(x, y^\star)\}$, where $y^\star$ is a high-quality response, such as an expert chain of thought. The

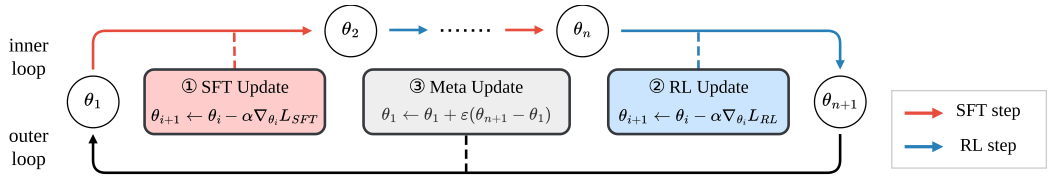


Figure 2: Overview of one iteration of our method. At each inner step, we perform either an SFT or RL update, selected according to a schedule or at random along a shared trajectory. The outer step then moves θ toward θ_{n+1}, thereby encouraging gradient matching across objectives.

objective is to maximize the likelihood of the demonstration under π_θ, or equivalently to minimize the token-level negative log-likelihood:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,y^\star)\sim\mathcal{D}_{\mathrm{SFT}}}\left[\sum_{t=1}^{|y^\star|} \log \pi_\theta(y_t^\star \mid x, y_{<t}^\star)\right]. \tag{2}$$

Thus, SFT provides dense imitation of expert trajectories and injects behavioral priors that may lie outside the current on-policy support.

Reinforcement learning. RL (Guo et al., 2025) instead improves the policy through reward feedback. Given a prompt x, the model generates a group of responses $\{y^{(i)}\}_{i=1}^{G} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x)$. Each response is assigned a verifiable reward $r^{(i)} = r(x, y^{(i)})$. Following GRPO (Shao et al., 2024), we estimate group-relative advantages as $A^{(i)} = \big(r^{(i)} - \mathrm{mean}(\{r^{(j)}\}_{j=1}^{G})\big)/\big(\mathrm{std}(\{r^{(j)}\}_{j=1}^{G}) + \varepsilon\big)$. The RL objective then maximizes the clipped importance-weighted advantage:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{x,\{y^{(i)}\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\min\Big(\rho_t^{(i)}A^{(i)},\ \mathrm{clip}(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon)\,A^{(i)}\Big)\right], \tag{3}$$

where $\rho_t^{(i)} = \pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})/\pi_{\theta_{\mathrm{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})$ is the token-level importance ratio. Unlike SFT, RL needs no step-by-step labels and learns via on-policy exploration with rewards.

Model-agnostic meta-learning. In first-order model-agnostic meta-learning Reptile (Nichol et al., 2018), each iteration consists of an inner loop and an outer loop. The inner loop samples a task τ and performs n updates with learning rate α on batches from the same distribution $p(\tau)$, yielding parameters θ'_τ. The outer loop then moves the original parameters θ toward this solution:

$$\theta \leftarrow \theta + \varepsilon\,(\theta'_\tau - \theta), \tag{4}$$

where ε is the outer learning rate. Expanding the inner-loop losses to first order, Eq. 4 induces the estimated meta-objective as:

$$\arg\min_\theta \sum_{i=1}^{n}\mathcal{L}_i(\theta) - \alpha\sum_{i\neq j}\nabla\mathcal{L}_i(\theta)\cdot\nabla\mathcal{L}_j(\theta), \tag{5}$$

where $\mathcal{L}_i$ denotes the loss of the i-th inner update. Maximizing the pairwise dot products encourages positive alignment between gradient directions, known as gradient matching.

2.2 Our Method

Eqs. 2 and 3 define two distinct objectives over the same parameter space. SFT pulls π_θ toward the supervised distribution, whereas RL reshapes π_θ according to the verifiable rewards. Their gradients $\nabla_\theta\mathcal{L}_{\mathrm{SFT}}$ and $\nabla_\theta\mathcal{L}_{\mathrm{RL}}$ may thus point in different directions, making naive combination unstable.

As shown in Section 2.1, meta-learning can mitigate gradient conflicts by increasing dot products. However, the standard formulation assumes that each inner step draws data from the same distribution. If one simply runs on SFT or RL alone, matching is encouraged only among successive SFT or RL objectives. This within-objective matching does not couple $\nabla\mathcal{L}_{\mathrm{SFT}}$ and $\nabla\mathcal{L}_{\mathrm{RL}}$, and thus can not directly resolve the conflict between the two post-training signals.

Algorithm 1 Meta-learning for joint SFT and RL

Require: Parameters θ; datasets $\mathcal{D}_{\mathrm{SFT}}$, $\mathcal{D}_{\mathrm{RL}}$; inner steps n; rates α, ε

for each outer iteration **do**
  $\theta_1 \leftarrow \theta$
  for $i = 1$ to n **do**
    Choose mode $m_i \in \{\mathrm{SFT}, \mathrm{RL}\}$ by schedule (or at random)
    if $m_i = \mathrm{SFT}$ **then**
      Sample a mini-batch $\mathcal{B}$ from $\mathcal{D}_{\mathrm{SFT}}$
      Compute $\mathcal{L}_{\mathrm{SFT}}(\theta_i; \mathcal{B})$ and set $\theta_{i+1} \leftarrow \theta_i - \alpha \nabla_{\theta_i} \mathcal{L}_{\mathrm{SFT}}(\theta_i; \mathcal{B})$
    else
      Sample prompts from $\mathcal{D}_{\mathrm{RL}}$, generate responses, and obtain rewards
      Compute $\mathcal{L}_{\mathrm{RL}}(\theta_i)$ and set $\theta_{i+1} \leftarrow \theta_i - \alpha \nabla_{\theta_i} \mathcal{L}_{\mathrm{RL}}(\theta_i)$
    end if
  end for
  $\theta \leftarrow \theta_1 + \varepsilon\,(\theta_{n+1} - \theta_1)$
end for

To bridge this gap, we place SFT and RL on a shared inner trajectory, so that matching can occur across objectives rather than only within each of them. As shown in Figure 2, let $\{\theta_i\}_{i=1}^{n+1}$ denote the parameters along the inner loop, where $\theta_1 = \theta$. At step i, we take a mini-batch under either the demonstration objective or the reward objective and update:

$$\theta_{i+1} \leftarrow \theta_i - \alpha \nabla \mathcal{L}_i(\theta_i), \qquad \mathcal{L}_i \in \{\mathcal{L}_{\mathrm{SFT}}, \mathcal{L}_{\mathrm{RL}}\}. \tag{6}$$

The sequence of objectives is alternated between SFT and RL to compose heterogeneous steps into one adaptation path. After n steps, we apply the outer update to the resulting parameters θ_{n+1}:

$$\theta \leftarrow \theta + \varepsilon\,(\theta_{n+1} - \theta). \tag{7}$$

Under the first-order analysis in Section 2.1, Eq. 7 corresponds to the meta-objective:

$$\arg\min_{\theta} \sum_{i=1}^{n} \mathcal{L}_i(\theta) - \alpha \sum_{i \neq j} \nabla \mathcal{L}_i(\theta) \cdot \nabla \mathcal{L}_j(\theta), \qquad \mathcal{L}_i, \mathcal{L}_j \in \{\mathcal{L}_{\mathrm{SFT}}, \mathcal{L}_{\mathrm{RL}}\}. \tag{8}$$

When all $\mathcal{L}_i$ belong to one objective, this reduces to the within-task form in Eq. 5. The key change is that $\mathcal{L}_i$ and $\mathcal{L}_j$ may now come from distinct objectives: when an SFT step is paired with an RL step, they enter the meta-objective that encourages alignment between $\nabla \mathcal{L}_{\mathrm{SFT}}$ and $\nabla \mathcal{L}_{\mathrm{RL}}$. As a result, the outer moves θ towards a region where progress on one is less likely to hinder the other.

This design also differs from common joint-training heuristics that mix the two losses with weights or inject supervised signals into RL rollouts. Such strategies still prescribe how SFT and RL should be combined at each step. In contrast, our method searches for parameters at which the two learning signals are compatible, thereby balancing imitation and exploration at parameter trajectories.

3 Analysis

We discuss the proposed scheme from three aspects: (1) Gradient matching, extending prior theory to multiple objectives; (2) Ensemble interpretation, connecting out loop to output-space interpolation; (3) Flat minima, providing a new intuition of why meta-learning helps generalization.

3.1 Gradient Matching

We show how Eq. 7 induces gradient matching, and why sharing the trajectory between SFT and RL yields cross-objective terms. Substituting the inner updates Eq. 6 into the outer step Eq. 7 gives:

$$\theta \leftarrow \theta - \varepsilon\alpha \sum_{i=1}^{n} \nabla \mathcal{L}_i(\theta_i). \tag{9}$$

This can be interpreted as optimizing the sum of losses along the trajectory:

$$\arg\min_{\theta} \sum_{i=1}^{n} \mathcal{L}_i(\theta_i). \tag{10}$$

We next expand each $\mathcal{L}_i(\theta_i)$ around $\theta_1 = \theta$. For the two-step case, $\theta_2 = \theta - \alpha\nabla\mathcal{L}_1(\theta)$ and:

$$\mathcal{L}_2(\theta_2) = \mathcal{L}_2(\theta) - \alpha\,\nabla\mathcal{L}_2(\theta)\cdot\nabla\mathcal{L}_1(\theta) + O(\alpha^2). \tag{11}$$

By induction on the inner step index, this extends to:

$$\mathcal{L}_i(\theta_i) = \mathcal{L}_i(\theta) - \alpha\sum_{j=1}^{i-1}\nabla\mathcal{L}_i(\theta)\cdot\nabla\mathcal{L}_j(\theta) + O(\alpha^2); \tag{12}$$

see supplementary material for proof. Substituting Eq. 12 into Eq. 10 and dropping $O(\alpha^2)$ yields:

$$\arg\min_{\theta} \sum_{i=1}^{n}\mathcal{L}_i(\theta) - \alpha\sum_{i=1}^{n}\sum_{j=1}^{i-1}\nabla\mathcal{L}_i(\theta)\cdot\nabla\mathcal{L}_j(\theta), \tag{13}$$

which is equivalent to the pairwise form in Eq. 8.

This expansion does not require all $\mathcal{L}_i$ to share one data distribution. In Reptile (Nichol et al., 2018), matching stays within a single objective. In our method, $\mathcal{L}_i \in \{\mathcal{L}_{\mathrm{SFT}}, \mathcal{L}_{\mathrm{RL}}\}$, and when steps i and j select different objectives, $\nabla\mathcal{L}_i(\theta)\cdot\nabla\mathcal{L}_j(\theta)$ becomes a cross term between $\nabla\mathcal{L}_{\mathrm{SFT}}$ and $\nabla\mathcal{L}_{\mathrm{RL}}$. Thus the outer rule that yields within-objective matching also yields cross-objective matching.

3.2 Ensemble Interpretation

Note that this argument use the same mechanism proposed by (Izmailov et al., 2018); we extend it from uniform averaging to a general interpolation.

The outer update can be viewed as a convex interpolation between the model parameters before and after the inner loop. Specifically, Eq. 7 can be written as:

$$\tilde{\theta} = (1-\varepsilon)\,\theta + \varepsilon\,\theta_{n+1}. \tag{14}$$

This parameter interpolation admits a local connection to output-space ensembling. To see this, let $f(\theta)$ denote the model output at a given input. Consider the corresponding output-space interpolation between the two endpoint models:

$$(1-\varepsilon)f(\theta) + \varepsilon f(\theta_{n+1}). \tag{15}$$

Expanding both $f(\theta)$ and $f(\theta_{n+1})$ around $\tilde{\theta}$ yields:

$$\begin{aligned}&(1-\varepsilon)f(\theta) + \varepsilon f(\theta_{n+1})\\ &\quad= f(\tilde{\theta}) + \Big[(1-\varepsilon)(\theta-\tilde{\theta}) + \varepsilon(\theta_{n+1}-\tilde{\theta})\Big]^{\top}\nabla f(\tilde{\theta})\\ &\qquad+ \mathcal{O}\big(\|\theta-\theta_{n+1}\|^2\big)\,.\end{aligned} \tag{16}$$

By the definition of $\tilde{\theta}$,

$$(1-\varepsilon)(\theta-\tilde{\theta}) + \varepsilon(\theta_{n+1}-\tilde{\theta}) = 0. \tag{17}$$

Hence,

$$(1-\varepsilon)f(\theta) + \varepsilon f(\theta_{n+1}) = f(\tilde{\theta}) + \mathcal{O}\big(\|\theta-\theta_{n+1}\|^2\big)\,. \tag{18}$$

Thus, when the inner-loop displacement is small (*i.e.*, $\|\theta-\theta_{n+1}\| \approx 0$), the outer update approximately matches the output of an ensemble of the initial and final models. This provides an additional interpretation of the outer update: rather than fully biased to either endpoint, it retains information from both models through local parameter interpolation.

3.3 FLAT MINIMA

Flat minima (Hochreiter & Schmidhuber, 1997), where loss varies only slightly under small changes of the parameters, are usually related to better generalization. Intuitively, optimizing towards such minima makes the model less sensitive to perturbations. In Eq. 9, the accumulated gradient update can be written as:

$$\theta \leftarrow \theta - \varepsilon\alpha \sum_{i=1}^{n} \nabla_{\theta+\delta_i} \mathcal{L}_i(\theta + \delta_i), \tag{19}$$

where $\theta_i = \theta + \delta_i$ denotes a small displacement of the parameters. Following the first-order update, each δ_i is a stop-gradient quantity, so δ_i is held fixed with respect to θ. By the chain rule,

$$\nabla_\theta \mathcal{L}_i(\theta + \delta_i) = \frac{\partial \mathcal{L}_i(\theta + \delta_i)}{\partial(\theta + \delta_i)} \frac{\partial(\theta + \delta_i)}{\partial\theta} = \nabla_{\theta+\delta_i} \mathcal{L}_i(\theta + \delta_i), \tag{20}$$

which denotes optimizing θ such that the loss is small on $\theta + \delta_i$. Each iteration aggregates this loss from multiple nearby parameters, and thereby the optimization targets not a single point, but a local region around θ. This makes sharp minima less favorable, as their loss could increase rapidly in the neighborhood, thereby guiding the model toward flatter ones.

4 EXPERIMENTS

4.1 SETUP

Model. We use Qwen2.5-Math-7B (Yang et al., 2024), Qwen3-4B-Base, Qwen3-8B-Base (Yang et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as our base models. We do not use DeepSeek-series, since the SFT data are generated by DeepSeek-R1 (Guo et al., 2025).

Dataset. When only one training signal is needed, or when the SFT/RL data depend on each other (*e.g.*, RELIFT (Ma et al., 2026)), we use OpenR1-Math-46k-8192 (Yan et al., 2026). When both signals are needed and are constructed independently, we use OpenR1-Math-46k-8192 for SFT and OpenR1-Math-48k-Complement (Yan et al., 2026) for RL. Both datasets are subsets of OpenR1-Math-220k (Face, 2025). Specifically, the prompts of OpenR1-Math-46k-8192 are collected from NuminaMath 1.5 (Li et al., 2024), and its off-policy reasoning traces are generated by DeepSeek-R1 (Guo et al., 2025). In contrast, OpenR1-Math-48k-Complement contains only the final answers.

Implementation. We implement our method in `verl` (Sheng et al., 2025) with colocated FSDP2 training and vLLM rollouts. Following (Limozin et al., 2026; Chen et al., 2026), we first warm up on SFT and then proceed to hybrid training. For SFT, we use token-level negative log-likelihood loss with batch size 128 and a maximum sequence length of 16,384 tokens. When run as warmup, SFT uses AdamW with learning rate 5×10^{-5} and a cosine schedule with 10% warmup for three epochs. During meta-learning, SFT inner steps share the same inner optimizer as RL. For RL, we follow (Liu et al., 2025b; Yan et al., 2026) which excludes both length normalization and standard error normalization from the GRPO (Shao et al., 2024) loss. We sample 8 responses per prompt at temperature 1.0, with a rollout batch size of 128 and an update batch size of 64. We set the clipping ratios to $[0.2, 0.28]$ and use rule-based answer correctness as rewards. For meta-learning, each outer iteration performs 3 inner steps. We either alternate between SFT and RL, or randomly sample from SFT and RL with probability $p_{\mathrm{SFT}} = 0.5$, followed by a Reptile-style outer update with SGD. We train for 200 iterations using inner learning rate 5×10^{-5} and outer learning rate 3×10^{-1} on 8 H200 GPUs. All models share the same system prompt for training and inference.

Evaluation. We evaluate on six math datasets: AIME 2024, AIME 2025, AMC (Li et al., 2024), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). Following prior work (Yan et al., 2026), we report avg@32 on AIME 2024/2025 and AMC, and pass@1 on MATH-500, Minerva, and OlympiadBench. To evaluate out-of-distribution generalization, we further evaluate with ARC-Challenge (Clark et al., 2018), GPQA-Diamond (Rein et al., 2023), and MMLU-Pro (Wang et al., 2024). For GPQA-Diamond, we exclude one excessively long example containing 6,386 characters. For multiple-choice benchmarks, we shuffle the options to reduce contamination. All evaluations use temperature 0.6 and top-$p = 0.9$.

Table 1: Main results on Qwen2.5-Math-7B **(Best in bold)**.

Method	AI24	AI25	AMC	MATH	MINV	OLYM	Avg.	ARC	GPQA	MMLU	Avg.
Baseline											
Base†	16.6	8.2	49.3	70.4	27.2	34.3	34.3	58.4	29.4	30.1	39.3
SFT*	33.6	28.2	65.4	90.0	40.6	55.7	52.2	82.5	24.6	50.2	52.4
SFT†	33.4	27.0	75.2	88.6	41.9	59.9	54.3	87.3	52.3	62.7	67.4
GRPO†	17.3	15.1	61.3	80.2	33.8	42.6	41.7	83.9	35.5	46.4	55.3
SFT→GRPO*	40.4	29.8	73.2	**92.0**	43.9	62.8	57.0	84.0	40.6	55.1	59.9
SFT→GRPO†	37.5	27.7	78.4	90.2	**44.9**	62.5	56.9	86.6	51.3	61.7	66.5
RL Method											
SimpleRL	27.0	6.8	54.9	76.0	25.0	34.7	37.4	30.2	23.2	34.5	29.3
OpenReasoner	16.5	15.0	52.1	82.4	33.1	47.1	41.0	66.2	29.8	58.7	51.6
PRIME	17.0	12.8	54.0	81.4	39.0	40.3	40.7	73.3	18.2	32.7	41.4
OAT	33.4	11.9	61.2	78.0	34.6	43.4	43.7	70.1	23.7	41.7	45.2
Hybrid Method (w/o SFT warm-up)											
LUFFY	30.7	22.5	66.2	86.8	41.2	55.3	50.4	81.8	49.0	54.7	61.8
ReLIFT	28.3	22.9	65.1	87.4	–	57.8	–	–	–	53.9	–
SRFT	35.3	–	74.3	89.8	39.7	58.3	–	85.3	46.4	55.9	62.5
Prefix-RFT	31.8	26.4	68.2	88.4	40.3	55.7	51.8	84.0	39.1	52.1	58.4
Hybrid Method (w/ SFT warm-up)											
LUFFY†	39.5	30.2	77.0	90.8	43.4	**64.5**	57.6	87.9	**55.8**	61.6	68.4
ReLIFT†	39.7	30.1	78.8	91.4	40.4	63.4	57.3	87.7	55.3	61.6	68.2
SRFT†	34.2	25.0	75.9	91.0	44.1	61.9	55.4	87.7	53.8	62.0	67.8
Prefix-RFT†	38.2	29.2	78.1	90.2	42.6	62.8	56.9	88.3	52.8	61.2	67.4
Ours	**41.8**	**30.5**	**79.7**	**92.0**	43.8	63.6	**58.6**	**89.2**	54.8	**63.3**	**69.1**

† Results with this symbol are reproduced by us, while other results are from previous papers.
* Specially, the previous SFT and SFT→GRPO results are from Limozin et al. (2026).

4.2 Main Results

We perform baselines including SFT-only, RL-only, and staged SFT→RL, and compare them with recent strong results (Limozin et al., 2026), as reported in the second and fifth rows of Table 1. They claim that previous work underestimated SFT due to several bugs in OpenRLHF (Hu et al., 2025).

We compare our method with several RL-based approaches, including SimpleRL (Zeng et al., 2025), OpenReasoner (Hu et al., 2026), PRIME (Cui et al., 2025), and OAT (Liu et al., 2025b). We also reproduce recent hybrid SFT+RL strategies, comprising LUFFY (Yan et al., 2026), SRFT (Fu et al., 2026), Prefix-RFT (Huang et al., 2025), and ReLIFT (Ma et al., 2026). Since our method uses SFT warm-up, we also apply it to these reproduced approaches. The results for Qwen2.5-Math-7B and Qwen3-4B-Base are reported in Table 1 and Table 2, respectively. Results for Qwen3-8B-Base and Llama-3.1-8B-Instruct are provided in the appendix.

(i) Our method achieves the best average performance across all models. (ii) Results obtained with correctly performed SFT outperform those from direct RL training (Limozin et al., 2026). (iii) Our reproduced results for both SFT→GRPO and the hybrid methods outperform those reported in the original paper. This suggest that SFT provides effective initialization for subsequent RL reasoning, while RL can further improve the model's performance after SFT warm-up stage.

4.3 Ablation & Discussion

Meta-learning, disentangling data, and scheduling. Table 3 ablates three design axes under a shared initialization: (i) whether training uses meta-learning; (ii) if so whether the inner loop sees SFT only, RL only, or both; (iii) how the two objectives are scheduled inside the inner loop.

Table 2: Main results on Qwen3-4B-Base **(Best in bold)**.

Method	AI24	AI25	AMC	MATH	MINV	OLYM	Avg.	ARC	GPQA	MMLU	Avg.
Baseline											
Base†	8.3	5.2	26.7	48.0	18.4	24.3	21.8	14.0	10.7	8.9	11.2
SFT†	29.4	26.9	67.5	83.8	41.5	53.7	50.5	91.8	52.8	68.1	70.9
GRPO†	12.5	11.9	53.8	78.4	37.9	45.5	40.0	92.2	56.9	68.0	72.4
SFT→GRPO†	31.3	25.6	69.7	86.6	**42.6**	57.6	52.2	91.8	47.2	65.9	68.3
Hybrid Method (w/ SFT warm-up)											
LUFFY†	29.2	26.3	71.5	86.0	41.2	**60.5**	52.5	94.0	55.3	68.9	72.7
ReLIFT†	32.3	27.1	70.0	82.4	40.4	59.9	52.0	92.6	54.3	67.8	71.6
SRFT†	31.4	25.2	69.7	84.4	38.6	60.4	51.6	93.5	53.8	68.5	71.9
Prefix-RFT†	30.7	27.5	71.2	85.6	40.1	59.9	52.5	94.1	52.8	**70.8**	72.6
Ours	**32.7**	**27.6**	**71.8**	**87.4**	41.9	**60.5**	**53.7**	**95.2**	**57.9**	68.4	**73.8**

Table 3: Ablation study on Qwen2.5-Math-7B **(Best in bold)**.

meta	*sft*	*rl*	*rand*	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
–	–	–	×	33.4	27.0	75.2	88.6	41.9	59.9	54.3
–	–	✓	×	37.5	27.7	78.4	90.2	44.9	62.5	56.9
–	✓	✓	–	39.3	30.4	79.1	90.4	42.6	63.4	57.5
✓	✓	–	×	39.0	27.8	79.6	90.0	40.8	61.6	56.5
✓	–	✓	×	40.2	28.6	**80.3**	91.8	43.8	63.6	58.1
✓	✓	✓	–	**42.0**	29.8	80.2	91.8	**46.3**	63.2	**58.9**
✓	✓	✓	✓	41.8	**30.5**	79.7	**92.0**	43.8	**63.6**	58.6

* All methods start from the same SFT warm-up stage.
* *Meta*: whether to use meta-learning; *sft*: whether to use SFT; *rl*: whether to use RL; *rand*: random selection in the inner loop, otherwise a fixed schedule. ✓ used; – not used; × inactive.

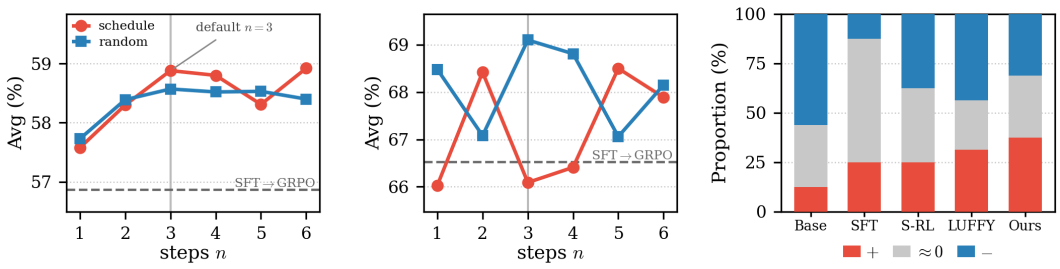
(i) Applying meta-learning to a single objective already provides gains, which lends support in Section 3.2 and 3.3. (ii) A shared trajectory further improves over single-objective run, indicating that the cross-objective interaction is important. (iii) Although schedule outperforms random selection in Table 3, we observe in Figure 3 that the difference between them is small. We therefore use random selection in the main table, as this choice is simpler.

Number of inner loop steps. We vary the number of inner loop steps $n \in \{1, 2, \ldots, 6\}$ under both the alternating schedule and random selection, as shown in Figure 3(a)(b). For in-distribution tasks, the performance of the schedule varies more than that of random. Both strategies initially improve as n increases and then stabilize, reaching their best at around $n = 3$. For out-of-distribution tasks, random selection slightly outperforms the alternating schedule. Overall, three inner-loop steps with random provide a good balance between accuracy and computational cost.

Validating signs of gradient cosines. To verify that our method promotes cross-objective compatibility, we examine the sign of the cosine similarity between $\nabla\mathcal{L}_{\mathrm{SFT}}$ and $\nabla\mathcal{L}_{\mathrm{RL}}$ at fixed checkpoints on Qwen3-4B-Base, as shown in Figure 3(c). We use a deadzone of $|c| < 0.01$ to characterize near-orthogonal gradients. On the base model, gradients of the great majority of SFT and RL pairs are either orthogonal or have negative cosine similarity, indicating conflicts between the two signals. In contrast, our method achieves the highest similarity, suggesting that it improves the compatibility between SFT and RL signals and therefore reduces their gradient conflicts.

5 Related Work

Joint and hybrid SFT–RL training. Current work seeks to combine supervised fine-tuning (SFT) and reinforcement learning (RL) for LLM reasoning, beyond the staged pipeline that first imitates



(a) In-domain (b) Out-of-domain (c) Cosine sign

Figure 3: (a)(b): Accuracy (%) varying with inner-loop steps on Qwen2.5-Math-7B. (c): Proportion (%) of positive, zero, and negative gradient-cosine signs on Qwen3-4B-Base.

SFT data and then optimizes RL rewards (Trung et al., 2024). Existing hybrids largely fall into two families according to where the supervised signal enters training.

The first family mixes the two objectives at the *loss* level. Hard variants route which loss to apply, such as interleaving RL with SFT on hard problems (Ma et al., 2026; Yuan et al., 2025), switching to SFT when rewards vanish or exploration collapses (Liu et al., 2025a; 2026a; Lv et al., 2025; Zhu et al., 2026; Bi et al., 2026), routing examples by conflicts (Zhao et al., 2026), or alternating offline recall and online extension (Guan et al., 2025). Soft variants instead optimize both objectives simultaneously through continuous weights, controlled by entropy (Fu et al., 2026), gradient norms (Chen et al., 2026), or global and token-wise coefficients (Zhang et al., 2026).

The second family aims to inject extra knowledge at the *rollout* stage. Off-policy expert traces can be mixed into on-policy RL with importance shaping or in-context steering (Yan et al., 2026; Huang et al., 2026). Alternatively, demonstrations can also enter as prefixes, hints (Huang et al., 2025; Liu et al., 2026b), or as structured templates that guide generation (Wu et al., 2025). In all these cases, the supervised prior reshapes the trajectories seen by RL rather than only reweighting the SFT loss, which provides a more direct way to transfer supervised information into RL.

Recently, (Limozin et al., 2026) argue that previous mixed-policy methods may underestimated the staged baseline. Specifically, bugs from DeepSpeed (Rasley et al., 2020) and OpenRLHF (Hu et al., 2025) can lead to incorrect gradient accumulation and loss weighting. With a strong and correct SFT pipeline restored, simple staged training can match or even surpass hybrid methods. Our experiments support their claim, so we use similar results as our baseline.

Across both hybrid families, the dominant design is still to prescribe a combination rule, including selecting loss, weighting objectives, or injecting knowledge into rollouts. In contrast, we shift focus to modeling how the two learning signals interact in parameter space.

Meta-learning. Meta-learning (Thrun & Pratt, 2012; Al-Shedivat et al., 2021) aims to find an initialization that enables rapid adaptation for new tasks with minimal updates. MAML (Finn et al., 2017) defines training as cycling between inner and outer loops. The inner loop is for task-specific adaptation, while the outer loop seeks a globally optimal initialization for the inner-loop task. Due to the computational challenges of second-order calculations, Reptile (Nichol et al., 2018) introduce first-order algorithms to reduce the cost while achieving fine-grained sampling. Our outer-loop update follows Reptile, but we integrate both SFT and RL signals within a single inner-loop trajectory, extending gradient matching from a within-objective setting to an cross-objective one.

6 Conclusion

In this paper, we study joint SFT and RL in post-training, where conflicting gradients make staged or naively mixed schedules difficult to balance. We propose a shared-trajectory meta-learning method that unifies both objectives during a single inner loop, implicitly promoting cross-objective gradient matching thus guiding the model towards less conflicting optimization regions. Experiments across math and out-of-distribution benchmarks show consistent gains over SFT-only, RL-only, and hybrid methods across diverse models.

7 AI Assistance Statement

In this work, we use generative AI tools, including Cursor Pro and free ChatGPT web interface, to assist with writing and related work discovery.

Writing. We initially draft the manuscript in a combination of native language and English. The draft is then translated into English and polished with ChatGPT. All AI-assisted text is subsequently reviewed and revised for technical accuracy, clarity, and fidelity to the intended claims.

Related work discovery. Part of the related work retrieval is assisted by Cursor Pro. Firstly, the related work is identified through Google Scholar searches and by following citations in the relevant papers. We then use Cursor Pro to assist in identifying additional potentially relevant papers, which are independently verified by us against the original sources before being cited.

References

Maruan Al-Shedivat, Liam Li, Eric Xing, and Ameet Talwalkar. On data efficiency of meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1369–1377. PMLR, 2021.

Jinhe Bi, Minglai Yang, Xingcheng Zhou, Wenke Huang, Sikuan Yan, Yujun Wang, Zixuan Cao, Michael Färber, Xun Xiao, Volker Tresp, et al. Echorl: Reinforcement learning via rollout echoing. *arXiv preprint arXiv:2605.31228*, 2026.

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. In *2026 29th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 4510–4516. IEEE, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

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. In *International Conference on Learning Representations*, volume 2026, pp. 102430–102459, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zhong Guan, Likang Wu, Hongke Zhao, Jiahui Wang, and Le Wu. Recall-extend dynamics: Enhancing small language models through controlled exploration and refined offline integration. *arXiv preprint arXiv:2508.16677*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

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.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, et al. Openrlhf: A ray-based easy-to-use, scalable and high-performance rlhf framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 656–666, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Hsiu-Yuan Huang, Chenming Tang, Weijie Liu, Clive Bai, Saiyong Yang, and Yunfang Wu. Think outside the policy: In-context steered policy optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 2758–2776, 2026.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. *arXiv preprint arXiv:2507.01679*, 2025.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Alexis Limozin, Eduard Durech, Torsten Hoefler, Imanol Schlag, and Valentina Pyatkin. Sft-then-rl outperforms mixed-policy methods for llm reasoning. *arXiv preprint arXiv:2604.23747*, 2026.

Jiazhen Liu, Yuchuan Deng, and Long Chen. Empowering small vlms to think with dynamic memorization and exploration. In *International Conference on Learning Representations*, volume 2026, pp. 80751–80774, 2026a.

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:101347–101383, 2026b.

Yihao Liu, Shuocheng Li, Lang Cao, Yuhang Xie, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. Superrl: Reinforcement learning with supervision to boost language model reasoning. *arXiv preprint arXiv:2506.01096*, 2025a.

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

Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Xuekai Zhu, Kaiyan Zhang, Bingning Wang, Ning Ding, et al. Towards a unified view of large language model post-training. *arXiv preprint arXiv:2509.04419*, 2025.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *International Conference on Learning Representations*, volume 2026, pp. 80802–80821, 2026.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

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.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7601–7614, 2024.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Haoran Luo, Ling Yang, Huazhe Xu, and Jianhua Tao. Templaterl: Structured template-guided reinforcement learning for llm reasoning. *arXiv preprint arXiv:2505.15692*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Xiangchi Yuan, Xiang Chen, Tong Yu, Dachuan Shi, Can Jin, Wenke Lee, and Saayan Mitra. Mitigating forgetting between supervised and reinforcement learning yields stronger reasoners. *arXiv preprint arXiv:2510.04454*, 2025.

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. *arXiv preprint arXiv:2503.18892*, 2025.

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. In *International Conference on Learning Representations*, volume 2026, pp. 120693–120726, 2026.

Yang Zhao, Yangou Ouyang, Xiao Ding, Hepeng Wang, Bibo Cai, Kai Xiong, Jinglong Gao, Zhouhao Sun, Li Du, Bing Qin, et al. Consolidation or adaptation? prism: Disentangling sft and rl data via gradient concentration. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23929–23941, 2026.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4396–4415, 2022.

Taojie Zhu, Dongyang Xu, Ding Zou, Sen Zhao, Qiaobo Hao, Zhiguo Yang, and Yonghong He. Bridging sft and rl: Dynamic policy optimization for robust reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4468–4484, 2026.