

RIFT: Repurposing Negative Samples via Reward-Informed Fine-Tuning

Zehua Liu, Shuqi Liu[†], Tao Zhong, Mingxuan Yuan

Huawei Noah’s Ark Lab

liuzehua@connect.hku.hk, liu.shuqi1@huawei.com

Abstract

While Supervised Fine-Tuning (SFT) and Rejection Sampling Fine-Tuning (RFT) are standard for LLM alignment, they either rely on costly expert data or discard valuable negative samples, leading to data inefficiency. To address this, we propose Reward Informed Fine-Tuning (RIFT), a simple yet effective framework that utilizes all self-generated samples. Unlike the hard thresholding of RFT, RIFT repurposes negative trajectories, reweighting the loss with scalar rewards to learn from both the positive and negative trajectories from the model outputs. To overcome the training collapse caused by naive reward integration, where direct multiplication yields an unbounded loss, we introduce a stabilized loss formulation that ensures numerical robustness and optimization efficiency. Extensive experiments on mathematical benchmarks across various base models show that RIFT consistently outperforms RFT. Our results demonstrate that RIFT is a robust and data-efficient alternative for alignment using mixed-quality, self-generated data.

1 Introduction

The rapid scaling of Large Language Models (LLMs) has made effective post-training adaptation essential (Zhang et al., 2023b; Chung et al., 2024a; Chu et al., 2025a). Supervised Fine-Tuning (SFT) (Ouyang et al., 2022b; Sanh et al., 2022), which minimizes the negative log-likelihood of expert demonstrations, constitutes the standard approach for aligning models with desired behaviors. However, the efficacy of SFT is heavily dependent upon the availability of high-quality demonstration data, which are generally difficult and costly to curate. More critically, a distributional mismatch between the pre-training data or initial model capabilities and the SFT data can lead to degraded performance,

[†]Corresponding author.

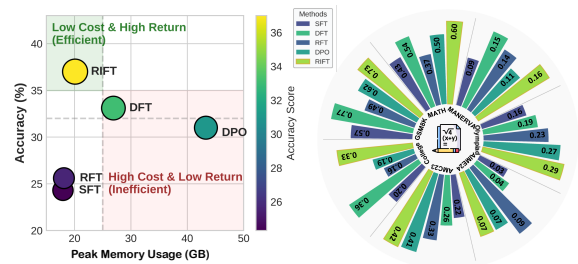


Figure 1: Efficiency and performance of post-training methods for Qwen2.5-Math-1.5B fine-tuned on MATH. **Left:** average accuracy against peak memory utilization (training efficiency); **Right:** per-dataset accuracy (generalization). RIFT surpasses strong baselines in accuracy while requiring less computational memory.

a phenomenon often described as catastrophic forgetting or alignment tax (Korbak et al., 2022; Luo et al., 2023; Huang et al., 2024a; Feng et al., 2025).

To mitigate these data-related limitations in SFT, Rejection Sampling Fine-Tuning (RFT) (Yuan et al., 2023; Chen et al., 2024) has emerged as a lightweight yet effective alternative. The underlying principle of RFT is straightforward: by sampling multiple responses from a base model to a given prompt and subsequently selecting only those that surpass a predefined quality threshold, one can construct a refined, higher-quality dataset for a subsequent round of SFT. Unlike conventional SFT that relies on pre-constructed static datasets, RFT generates its training data through the model’s own sampling process. This self-generation strategy inherently promotes alignment between the data distribution used for fine-tuning and the model’s own output distribution. Furthermore, the quality and correctness of the selected data is ensured through an external verification mechanism or a well-defined scoring function.

Despite its simplicity and advantages, the standard RFT paradigm exhibits a critical shortcoming: it discards all sub-threshold (negative) samples outright. This discard policy neglects the potential

informational value these samples carry regarding model failure modes. Consequently, it not only wastes computational resources expended during generation, but may also impair the model to learn distinctions between correct and incorrect outputs, thereby limiting its capacity to refine its understanding of subtle errors.

To better utilize all generated responses, including those rejected by quality thresholds in RFT, we propose **Reward Informed Fine-Tuning (RIFT)**. RIFT constitutes a simple and efficient extension of standard SFT. In contrast to RFT, which discards low-scoring candidates via hard thresholding, RIFT retains every sampled trajectory and assigns it a scalar reward derived from a quality evaluation metric. The RIFT objective modifies the standard negative log-likelihood loss by reweighting each sample’s contribution proportionally to its assigned reward. This design ensures that positive-reward samples encourage correct behaviors, whereas negative-reward samples provide reduced or negative gradients.

Nevertheless, a naive integration of the reward signal with the logarithmic probability term presents a significant practical challenge. As we will demonstrate in Section 3, directly multiplying these components produces a loss function that is unbounded from below, inevitably leading to severe training collapse. To address this fundamental issue, we introduce a principled framework for loss function formulation, which is designed to ensure guaranteed training stability and maintain optimization efficiency. Figure 1 compares RIFT with strong baselines (e.g., DPO (Rafailov et al., 2023), DFT (Wu et al., 2025)) on Qwen2.5-Math-1.5B: RIFT achieves comparable or superior accuracy at substantially lower peak memory utilization.

Empirically, RIFT delivers consistent and substantial improvements across model scales and alignment settings on mathematical reasoning benchmarks. RIFT outperforms SFT, DFT (Wu et al., 2025), RFT (Yuan et al., 2023) and DPO (Rafailov et al., 2023) in both in-distribution accuracy and out-of-distribution generalization on Qwen2.5-Math (1.5B/7B) (Yang et al., 2024b), Qwen3-1.7B (Yang et al., 2025), and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025). Unlike RFT (Yuan et al., 2023), which critically depends on strong base models to generate high-quality rollouts, RIFT remains stable and effective even with moderately capable models. In off-policy settings, RIFT consistently surpasses DPO (Rafailov et al.,

2023) across models. Notably, RIFT eliminates the need for a reference model, offering a simpler and more resource-efficient alternative for alignment.

2 Related Works

LLM Post-training Supervised Fine-Tuning (SFT) has become the standard post-training paradigm for adapting pretrained models to specific tasks using high-quality, labeled datasets (Zhang et al., 2023a; Chung et al., 2024b). Although the availability of high-quality instruction-following datasets (Cobbe et al., 2021; Mishra et al., 2022; Zhou et al., 2023a; Taori et al., 2023) has significantly enhanced the efficacy of SFT, studies (Dodge et al., 2020; Howard and Ruder, 2018; Ouyang et al., 2022a) indicate that SFT often suffers from overfitting and suboptimal generalization.

To mitigate the challenges of SFT data curation, Reinforcement Learning (RL) has emerged as a powerful alternative for post-training. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022a) and Reinforcement Learning from Verifiable Reward (RLVR) (Guo et al., 2025; Shao et al., 2024) are widely used to align models with human preferences and enhance reasoning, supported by algorithms like DPO (Rafailov et al., 2023), SimPO (Meng et al., 2024a), GRPO (Shao et al., 2024), and DAPO (Yu et al., 2025). However, in contrast to SFT, RL-based training is often more complex, necessitating intricate engineering frameworks (Zheng et al., 2025).

Improving SFT Motivated by the success of RL methods, a growing body of research aims to enhance SFT by integrating RL principles. RFT (Yuan et al., 2023) utilizes self-generated trajectories filtered for correctness as training data. Other approaches re-frame RL objectives within an SFT framework, such as integrating importance sampling (Qin and Springenberg, 2025) or adopting PPO-style clipped surrogates (Zhu et al., 2025). However, these methods typically require a reference model, imposing a procedural complexity that aligns them more closely with RL paradigms than the simplicity of conventional SFT.

In a parallel line of inquiry, other research focuses on directly modifying the SFT loss without a reference model. For example, DFT (Wu et al., 2025) rescales the SFT objective at each token by its probability. Building on this, Li et al. (2025) proposed a unified framework for designing loss objectives, demonstrating that DFT can be considered

a special case within their formulation. Following this direction, our proposed method, RIFT, refines the loss function to enhance performance while preserving the simplicity of standard SFT.

3 Methodology

In this section, we present the theoretical framework and methodology of RIFT (Reward-Informed Fine-Tuning), a generalization of SFT that explicitly leverages mixed-quality demonstrations, i.e., samples with both positive and negative trajectories. An overview of the RIFT framework is depicted in Figure 2. In particular, we address the core challenge of leveraging negative-reward samples without compromising training stability.

3.1 Preliminaries: Generalized Signed-Weighted Objective

Standard SFT relies on Maximum Likelihood Estimation (MLE) over high-quality demonstrations, effectively assigning uniform positive weight to all training samples. However, when both positive and negative feedback are available, it is natural to extend MLE by weighting each sample proportionally to its reward: positive rewards encourage likelihood increase, while negative rewards suppress undesirable outputs. This leads to a generalized signed-weighted objective.

Let $\mathcal{D} = \{(x, y, r)\}$ be a dataset where x denotes the input, y the response sampled from the data distribution $\pi_{ref}(\cdot|x)$, and $r : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ a scalar reward signal indicating the quality of the response. We partition the response space into positive samples $\mathcal{D}^+ = \{(x, y) \mid r(x, y) > 0\}$ and negative samples $\mathcal{D}^- = \{(x, y) \mid r(x, y) < 0\}$.

Definition 3.1 (Naive Signed-Weighted Loss). The naive signed-weighted loss function \mathcal{L}_{naive} for a parameterized policy π_θ is defined as the expectation of the reward-weighted log-likelihood:

$$\mathcal{L}_{naive}(\theta) := -\mathbb{E}_{(x,y,r) \sim \mathcal{D}} [r \cdot \log \pi_\theta(y|x)]. \quad (1)$$

The optimization dynamics of Eq. (1) are determined by the sign of r :

- **Positive Reinforcement** ($r > 0$): Minimizing \mathcal{L}_{naive} is equivalent to **maximizing** $\log \pi_\theta(y|x)$, aligning with the standard SFT objective to promote desirable responses.
- **Negative Suppression** ($r < 0$): Minimizing \mathcal{L}_{naive} is equivalent to **minimizing** $\log \pi_\theta(y|x)$, theoretically suppressing the generation of undesirable responses.

3.2 Theoretical Analysis of Instability

While the naive formulation provides a unified view of reinforcement and suppression, it is ill-posed for negative weights due to the asymptotic behavior of the logarithm function.

Theorem 3.2 (Gradient Explosion and Unboundedness). Consider a negative sample $(x, y) \in \mathcal{D}^-$ with weight $r < 0$. The contribution to the gradient of the loss function \mathcal{L}_{naive} with respect to the probability $\pi_\theta(y|x)$ is:

$$\frac{\partial \mathcal{L}_{naive}}{\partial \pi_\theta} = -\frac{r}{\pi_\theta(y|x)}. \quad (2)$$

As the model successfully suppresses the negative sample (i.e., $\pi_\theta(y|x) \rightarrow 0^+$), the gradient magnitude approaches infinity:

$$\lim_{\pi_\theta \rightarrow 0^+} \left| \frac{\partial \mathcal{L}_{naive}}{\partial \pi_\theta} \right| = \infty. \quad (3)$$

Furthermore, the objective function itself is unbounded from below, as $\lim_{p \rightarrow 0^+} (-r \log p) = -\infty$ for $r < 0$.

Theorem 3.2 reveals a fundamental optimization pathology: the better the model performs at suppressing a negative sample, the more unstable the gradients become. In practice, this singularity leads to numerical overflow and catastrophic forgetting, where the optimizer focuses excessively on driving infinitesimal probabilities to absolute zero, destroying the feature representations learned from positive data.

3.3 Reward Informed Fine-Tuning (RIFT)

Theorem 3.2 shows a key pathology: stronger suppression of negative samples leads to increasingly unstable gradients. To address the instability, RIFT replace the logarithmic objective for negative samples with a bounded surrogate.

3.3.1 Linear Probability Approximation

Motivated by the first-order Taylor expansion of the logarithm function. For a probability $u \in (0, 1]$, the expansion around $u = 1$ is given by:

$$\log u = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} (u-1)^n \approx u-1. \quad (4)$$

Although the linear surrogate $u-1$ is not accurate near $u = 0$, we adopt it for its stable gradient: unlike $\log u$, its constant derivative avoids explosion as $u \rightarrow 0$, ensuring numerical stability while still suppressing negative samples.

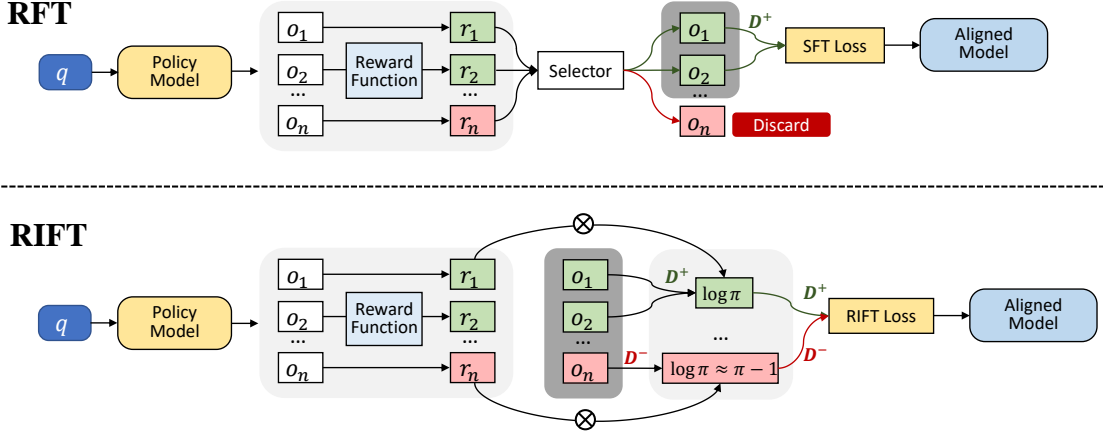


Figure 2: A comparative overview of RFT and RIFT. Unlike RFT rejects negative samples and only trains on positive ones, RIFT repurposes negative samples through a unified reward-informed loss. To ensure stable optimization, a linear surrogate is applied to negative samples to prevent loss collapse.

3.3.2 The RIFT Objective

RIFT decouples positive and negative samples: it retains the log objective for positives (preserving MLE signal) and uses a linear objective for negatives (ensuring stable gradients).

Definition 3.3 (RIFT Loss). Let \mathcal{D}^+ and \mathcal{D}^- be the disjoint sets of positive and negative samples. The RIFT loss function is defined as:

$$\mathcal{L}_{\text{RIFT}}(\theta) := -\mathbb{E}_{(x,y)\sim\mathcal{D}^+} [r(x,y) \cdot \log \pi_{\theta}(y|x)] - \mathbb{E}_{(x,y)\sim\mathcal{D}^-} [r(x,y) \cdot \pi_{\theta}(y|x)]. \quad (5)$$

For $y \in \mathcal{D}^+$, we have $r(x,y) > 0$; thus, minimizing Eq. (5) increases $\pi_{\theta}(y|x)$, thereby enhancing positive samples. In the second term, since $r(x,y) < 0$ for samples in \mathcal{D}^- , the term $-r(x,y)$ is positive. Thus, minimizing Eq. (5) requires minimizing $\pi_{\theta}(y|x)$, effectively suppressing the negative samples.

Theorem 3.4 (Stability and Properties of RIFT). *The RIFT formulation satisfies the following theoretical properties:*

- (i) **Boundedness:** *Since $\pi_{\theta} \in [0, 1]$, the loss contribution from any negative sample is bounded in $[r, 0]$, preventing the divergence to $-\infty$.*
- (ii) **Reward Lower-Bound Maximization:** *Let $\mathcal{J}(\theta) := \mathbb{E}_{y\sim\pi_{\theta}} [r(x,y)]$ denote the expected reward. Optimizing $\mathcal{L}_{\text{RIFT}}$ can be viewed as maximizing a surrogate lower bound of $\mathcal{J}(\theta)$.*

By replacing the unbounded logarithmic penalty with a bounded linear penalty, RIFT provides stable incorporation of negative samples, ensuring that the suppression of undesirable content does not compromise the stability of the fine-tuning process.

4 Experiments

4.1 Experiment Details

Base Models and Off-Policy Data Construction

We evaluate RIFT against four established baselines. We first include supervised and rejection-sampling-based methods: SFT, DFT (Wu et al., 2025), and RFT (Yuan et al., 2023). Furthermore, as models can benefit from contrasting correct and incorrect outcomes, we also compare against DPO (Rafailov et al., 2023), a representative off-policy RL method. Experiments are conducted on Qwen2.5-Math (1.5B, 7B) (Yang et al., 2024b), Qwen3-1.7B (Yang et al., 2025), and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025), with the Qwen3 variant evaluated in non-thinking mode.

| Model | # Num. | # Total | # Pos. (r>0) | # Neg. (r<0) | % Pos. |
|-----------------------------------|--------|---------|--------------|--------------|--------|
| <i>Source: MATH Dataset</i> | | | | | |
| Qwen-2.5-Math-1.5B | 3,000 | 24,000 | 15,941 | 8,059 | 66.4% |
| Qwen-2.5-Math-7B | 3,000 | 24,000 | 16,933 | 7,067 | 70.6% |
| Qwen-3-1.7B | 3,000 | 24,000 | 20,386 | 3,614 | 84.9% |
| <i>Source: NuminaMath Dataset</i> | | | | | |
| Qwen-2.5-Math-1.5B | 4,000 | 32,000 | 11,235 | 20,765 | 35.1% |
| Qwen-2.5-Math-7B | 4,000 | 32,000 | 10,581 | 21,419 | 33.1% |
| Qwen-3-1.7B | 4,000 | 32,000 | 20,352 | 11,648 | 63.6% |

Table 1: Training data statistics across models and datasets, including counts of positive and negative samples and the positive sample ratio.

To construct off-policy training data, we curate two buffers from 3,000 randomly sampled MATH (Hendrycks et al., 2021b) and 4,000 NuminaMath (LI et al., 2024) problems. For each problem, the base model generates 8 candidate responses, each assigned a reward based on final-answer correctness: positive reward for correct responses and

| Model | Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|---|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------------|
| <i>Post-Train on MATH Dataset</i> | | | | | | | | | |
| Qwen-2.5-Math-1.5B | Base | 42.6 | 35.6 | 9.7 | 22.6 | 7.1 | 31.9 | 8.2 | 22.5 |
| | SFT | 57.0 | 42.9 | 9.3 | 16.1 | 3.3 | 21.9 | 19.9 | 24.3 |
| | DFT | 76.8 | 53.9 | 15.6 | 19.1 | 4.2 | 25.6 | 36.4 | <u>33.1</u> |
| | RFT | 48.8 | 37.2 | 13.5 | 22.5 | 8.8 | 33.4 | 15.2 | 25.6 |
| | DPO | 61.8 | 50.3 | 11.3 | 26.7 | 7.1 | 41.2 | 18.6 | 31.0 |
| | RIFT | 72.6 | 59.6 | 15.8 | 28.8 | 7.1 | 41.9 | 33.3 | 37.0 (+11.4) |
| Qwen-2.5-Math-7B | Base | 54.8 | 50.3 | 12.2 | 16.4 | 12.1 | 36.9 | 20.5 | 29.0 |
| | SFT | 67.0 | 48.9 | 10.8 | 16.6 | 2.9 | 25.6 | 26.9 | 28.4 |
| | DFT | 83.3 | 58.5 | 16.9 | 20.9 | 4.6 | 33.8 | 35.2 | 36.2 |
| | RFT | 79.3 | 72.1 | 21.3 | 35.7 | 11.2 | 59.1 | 42.0 | <u>45.8</u> |
| | DPO | 62.0 | 61.7 | 26.3 | 31.3 | 16.2 | 50.3 | 36.8 | 40.7 |
| | RIFT | 84.6 | 74.0 | 25.4 | 36.1 | 17.9 | 58.8 | 43.8 | 48.7 (+2.9) |
| Qwen-3-1.7B (Non-thinking mode) | Base | 77.0 | 42.3 | 19.1 | 13.4 | 1.2 | 22.5 | 30.8 | 29.5 |
| | SFT | 80.0 | 50.1 | 22.5 | 17.7 | 1.2 | 28.4 | 33.8 | 33.4 |
| | DFT | 84.4 | 57.0 | 27.7 | 21.7 | 4.2 | 31.2 | 36.3 | 37.5 |
| | RFT | 87.0 | 67.3 | 30.1 | 27.5 | 5.0 | 39.1 | 41.1 | 42.4 |
| | DPO | 86.6 | 66.0 | 26.1 | 26.5 | 6.7 | 43.4 | 35.7 | 41.6 |
| | RIFT | 87.3 | 69.3 | 32.7 | 29.7 | 7.9 | 41.6 | 41.5 | 44.3 (+1.9) |
| <i>Post-Train on NuminaMath Dataset</i> | | | | | | | | | |
| Qwen-2.5-Math-1.5B | Base | 42.6 | 35.6 | 9.7 | 22.6 | 7.1 | 31.9 | 8.2 | 22.5 |
| | SFT | 67.5 | 51.4 | 11.8 | 18.5 | 5.0 | 29.4 | 30.9 | 30.6 |
| | DFT | 77.4 | 57.8 | 17.3 | 25.2 | 6.7 | 31.2 | 34.2 | 35.7 |
| | RFT | 69.7 | 62.1 | 15.2 | 28.6 | 5.2 | 37.8 | 32.7 | <u>35.9</u> |
| | DPO | 73.5 | 61.9 | 15.8 | 27.6 | 3.3 | 37.7 | 31.1 | 35.8 |
| | RIFT | 75.2 | 62.4 | 18.1 | 27.8 | 7.1 | 40.0 | 33.5 | 37.7 (+1.4) |
| Qwen-2.5-Math-7B | Base | 54.8 | 50.3 | 12.2 | 16.4 | 12.1 | 36.9 | 20.5 | 29.0 |
| | SFT | 71.1 | 60.9 | 21.8 | 32.9 | 9.2 | 43.4 | 37.0 | 39.3 |
| | DFT | 87.0 | 70.6 | 26.1 | 34.7 | 7.5 | 44.7 | 37.9 | 44.1 |
| | RFT | 83.3 | 69.8 | 21.3 | 31.3 | 11.2 | 58.8 | 42.0 | 45.4 |
| | DPO | 84.5 | 71.4 | 27.2 | 32.9 | 16.2 | 56.1 | 38.4 | 46.7 |
| | RIFT | 86.3 | 74.7 | 28.9 | 34.1 | 17.1 | 62.2 | 38.6 | 48.8 (+3.4) |
| Qwen-3-1.7B (Non-thinking mode) | Base | 77.0 | 42.3 | 19.1 | 13.4 | 1.2 | 22.5 | 30.8 | 29.5 |
| | SFT | 84.7 | 62.2 | 22.9 | 24.3 | 2.5 | 37.8 | 34.1 | 38.4 |
| | DFT | 87.6 | 69.9 | 30.5 | 28.3 | 3.3 | 42.5 | 36.2 | 42.6 |
| | RFT | 86.4 | 62.3 | 25.5 | 24.3 | 3.3 | 36.6 | 34.6 | <u>39.0</u> |
| | DPO | 86.8 | 66.7 | 27.7 | 26.3 | 6.7 | 40.8 | 36.0 | 41.6 |
| | RIFT | 88.2 | 69.2 | 28.2 | 28.7 | 3.3 | 46.6 | 36.3 | 42.9 (+3.9) |

Table 2: Mean@8 accuracy (%) on 7 mathematical benchmarks. Best results are in **bold**. (+) indicates the absolute improvement of RIFT compared to RFT.

negative for incorrect ones. Following findings in MGPO (Xu et al., 2025), we set larger magnitude reward (+1.0) for positive responses than for negative ones (-0.2) to emphasize successful reasoning traces. We analyze sensitivity to the negative reward in Section 5.1. The final buffers consist of (x, y, r) triplets, with statistics in Table 1. Regarding learning strategies: SFT and DFT train on the seed problems with their ground-truth solutions; RFT uses only positive-reward responses, discarding all negative ones; DPO forms preference pairs by comparing model responses to ground-truth solutions, preferring the response when correct and the ground truth otherwise; In contrast, RIFT leverages the full training buffer, requiring neither data filtering nor explicit preference pairing.

Implementation Details and Hyperparameter Settings We implement baselines using the built-in recipes of the MS-Swift (Zhao et al., 2024) framework, while RIFT is implemented via TRL (von Werra et al., 2020). Unless otherwise speci-

fied, we adopt the default configurations provided by MS-Swift. For candidate response generation, we sample 8 candidates per problem with a temperature of 0.7 and a maximum sequence length of 4,096. During inference, all models maintain these settings with a top- p of 0.8 and a fixed random seed (0) for reproducibility. Optimization is carried out using the AdamW (Loshchilov and Hutter, 2019) optimizer coupled with a cosine learning rate scheduler featuring a 5% warmup phase. The learning rate is set to 1×10^{-5} for SFT and RFT, and a more conservative 2×10^{-6} for RIFT and DPO to ensure stability during preference-based updates; All experiments are conducted with a global batch size of 64 over three epochs.

Evaluation Benchmarks and Metrics Following prior studies, we adopt mathematical tasks as our primary testbed. Specifically, we evaluate on seven math benchmarks: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), Minerva Math (Lewkowycz et al., 2022), Olympiad Bench

| Model | Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|---|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------------|
| <i>Post-Train on Math Dataset</i> | | | | | | | | | |
| Qwen-2.5-Math-1.5B | Base | 88.0 | 75.1 | 32.0 | 46.5 | 23.3 | 67.5 | 30.1 | 51.8 |
| | SFT | 93.6 | 80.5 | 29.0 | 43.7 | 16.7 | 60.0 | 49.0 | 53.2 |
| | DFT | 94.5 | 76.7 | 38.2 | 43.0 | 20.0 | 55.0 | 52.4 | 54.3 |
| | RFT | 87.0 | 67.3 | 30.1 | 27.5 | 5.0 | 39.1 | 41.1 | 42.4 |
| | DPO | 92.9 | 83.2 | 33.8 | 48.7 | 33.3 | 72.5 | 45.4 | 58.5 |
| | RIFT | 93.9 | 85.9 | 37.1 | 51.7 | 30.0 | 80.0 | 51.9 | 61.5 (+19.1) |
| Qwen-2.5-Math-7B | Base | 92.1 | 83.6 | 36.4 | 42.5 | 30.0 | 70.0 | 48.2 | 50.7 |
| | SFT | 95.8 | 82.8 | 33.5 | 43.0 | 16.7 | 62.5 | 49.6 | 54.8 |
| | DFT | 92.3 | 72.6 | 33.1 | 39.1 | 13.3 | 60.0 | 47.3 | 51.1 |
| | RFT | 95.5 | 90.2 | 46.7 | 58.1 | 33.3 | 85.0 | 54.2 | 66.1 |
| | DPO | 94.6 | 88.9 | 52.2 | 56.1 | 33.3 | 82.5 | 55.1 | 66.1 |
| | RIFT | 96.4 | 90.1 | 49.6 | 59.3 | 36.7 | 85.0 | 55.5 | 67.5 (+1.4) |
| Qwen-3-1.7B (Non-thinking mode) | Base | 90.6 | 67.1 | 34.9 | 31.4 | 10.0 | 45.0 | 41.9 | 45.8 |
| | SFT | 92.7 | 74.1 | 38.2 | 36.4 | 10.0 | 47.5 | 44.8 | 49.1 |
| | DFT | 94.4 | 80.2 | 43.8 | 41.8 | 13.3 | 55.0 | 47.5 | 53.7 |
| | RFT | 94.3 | 84.6 | 42.3 | 40.0 | 20.0 | 60.0 | 48.2 | 55.6 |
| | DPO | 94.7 | 82.6 | 39.3 | 41.6 | 26.7 | 70.0 | 41.2 | 56.7 |
| | RIFT | 94.8 | 85.6 | 45.6 | 45.8 | 20.0 | 65.0 | 48.4 | 57.9 (+2.3) |
| <i>Post-Train on NuminaMath Dataset</i> | | | | | | | | | |
| Qwen-2.5-Math-1.5B | Base | 88.0 | 75.1 | 32.0 | 46.5 | 23.3 | 67.5 | 30.1 | 51.8 |
| | SFT | 94.0 | 82.8 | 36.8 | 45.3 | 16.7 | 70.0 | 54.2 | 57.1 |
| | DFT | 93.3 | 85.3 | 40.1 | 50.4 | 16.7 | 62.5 | 52.0 | 57.2 |
| | RFT | 93.6 | 86.1 | 38.2 | 52.4 | 23.3 | 75.0 | 45.4 | 59.1 |
| | DPO | 93.6 | 85.9 | 39.3 | 51.7 | 23.3 | 77.5 | 46.1 | 59.6 |
| | RIFT | 94.2 | 86.4 | 41.9 | 49.5 | 26.7 | 80.0 | 45.8 | 60.6 (+1.5) |
| Qwen-2.5-Math-7B | Base | 92.1 | 83.6 | 36.4 | 42.5 | 30.0 | 70.0 | 48.2 | 50.7 |
| | SFT | 96.0 | 89.5 | 45.2 | 60.3 | 23.3 | 80.0 | 56.4 | 64.4 |
| | DFT | 91.7 | 81.8 | 37.5 | 48.7 | 16.7 | 62.5 | 42.7 | 54.5 |
| | RFT | 95.5 | 90.1 | 49.6 | 56.1 | 33.3 | 82.5 | 51.9 | 65.6 |
| | DPO | 95.8 | 90.2 | 52.2 | 58.1 | 36.7 | 85.0 | 54.2 | 67.5 |
| | RIFT | 96.3 | 90.6 | 53.3 | 58.4 | 43.3 | 87.5 | 48.3 | 68.2 (+2.6) |
| Qwen-3-1.7B (Non-thinking mode) | Base | 90.6 | 67.1 | 34.9 | 31.4 | 10.0 | 45.0 | 41.9 | 45.8 |
| | SFT | 93.5 | 81.3 | 34.9 | 41.2 | 10.0 | 67.5 | 40.8 | 52.7 |
| | DFT | 93.4 | 81.6 | 39.3 | 43.4 | 16.7 | 67.5 | 41.0 | 54.7 |
| | RFT | 94.8 | 81.3 | 38.6 | 41.0 | 13.3 | 62.5 | 40.7 | 53.2 |
| | DPO | 94.1 | 81.6 | 38.6 | 41.6 | 13.3 | 62.5 | 41.2 | 53.3 |
| | RIFT | 94.9 | 85.6 | 42.6 | 43.1 | 13.3 | 70.0 | 41.8 | 55.9 (+2.7) |

Table 3: Pass@8 accuracy (%) on 7 mathematical benchmarks. Best results are in **bold**. (+) indicates the absolute improvement of RIFT compared to RFT.

(Huang et al., 2024b), AIME 2024 (Mathematical Association of America, 2024), AMC 2023 (Mathematical Association of America, 2023), and College Math (Hendrycks et al., 2021a). We use the standardized Qwen2.5-Math-Eval pipeline (Yang et al., 2024a) and report Mean@8 and Pass@8.

4.2 Main Results

Mean@8 Performance Table 2 reports Mean@8 accuracy across seven mathematical reasoning benchmarks. RIFT consistently achieves the highest average performance in all settings, surpassing SFT, DFT, RFT, and DPO without requiring explicit preference pairs or data filtering. Our analysis yields the following findings:

(1) SFT and DFT: limited OOD generalization. Trained solely on MATH, SFT and DFT underperform the base model on harder OOD tasks (e.g., DFT: 19.1 vs. 22.6 on Olympiad; 4.6 vs. 12.1 on AIME24), but recover with NuminaMath whose distribution better aligns with the benchmarks. In

contrast, by leveraging mixed-reward responses, RIFT consistently outperforms the base across all benchmarks, even under MATH-only training.

(2) RFT scales with model capacity. On Qwen-Math-1.5B, RFT underperforms DPO (25.6 vs. 31.0), but surpasses it on Qwen-Math-7B (45.8 vs. 40.7), indicating that RFT requires sufficient high-quality positive samples for self-improvement. However, RIFT stabilizes the refinement process even when self-generation quality is moderate.

| Model | # Num. | # Mixed-Reward Num. | % Mixed |
|-----------------------------------|--------|---------------------|---------|
| <i>Source: MATH Dataset</i> | | | |
| Qwen-2.5-Math-1.5B | 3,000 | 2,541 | 84.7% |
| Qwen-2.5-Math-7B | 3,000 | 1,947 | 64.9% |
| Qwen-3-1.7B | 3,000 | 971 | 32.4% |
| <i>Source: NuminaMATH Dataset</i> | | | |
| Qwen-2.5-Math-1.5B | 4,000 | 2,060 | 51.5% |
| Qwen-2.5-Math-7B | 4,000 | 2,305 | 57.6% |
| Qwen-3-1.7B | 4,000 | 3,462 | 86.6% |

Table 4: Fraction of problems with mixed correct and incorrect responses (out of 8) per model and dataset.

| Model | Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------------|
| DeepSeek-R1-Distill-Qwen-1.5B (Mean@8) | Base | 80.5 | 70.4 | 19.5 | 30.2 | 12.9 | 47.5 | 39.6 | 43.0 |
| | SFT | 50.5 | 38.2 | 10.8 | 9.7 | 0.4 | 14.4 | 25.2 | 21.3 |
| | DFT | 76.3 | 70.6 | 23.5 | 30.1 | 13.3 | 42.5 | 37.8 | 42.0 |
| | RFT | 63.6 | 63.5 | 14.7 | 27.4 | 13.8 | 48.1 | 33.9 | 37.9 |
| | DPO | 80.9 | 69.2 | 16.9 | 28.8 | 14.6 | 50.3 | 39.2 | 42.8 |
| | RIFT | 82.1 | 71.1 | 22.3 | 30.3 | 13.3 | 48.8 | 40.1 | 44.0 (+6.1) |
| DeepSeek-R1-Distill-Qwen-1.5B (Pass@8) | Base | 95.1 | 89.8 | 36.4 | 40.9 | 33.3 | 70.0 | 50.2 | 59.4 |
| | SFT | 85.0 | 71.2 | 31.2 | 31.0 | 3.3 | 32.5 | 46.1 | 45.8 |
| | DFT | 93.8 | 89.4 | 43.4 | 50.7 | 30.0 | 77.5 | 48.2 | <u>61.9</u> |
| | RFT | 92.6 | 89.2 | 31.2 | 47.9 | 30.0 | 75.0 | 50.0 | 59.4 |
| | DPO | 94.9 | 89.1 | 32.7 | 47.0 | 30.0 | 77.5 | 49.5 | 60.1 |
| | RIFT | 95.2 | 89.9 | 40.4 | 50.7 | 33.3 | 82.5 | 50.3 | 63.2 (+3.8) |

Table 5: Mean@8 and Pass@8 accuracy (%) on 7 mathematical benchmarks for DeepSeek-R1-Qwen-1.5B model. Best results are in **bold**. (+) indicates the absolute improvement of RIFT compared to RFT.

| Reward Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|-----------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| <i>Group Normalization Reward</i> | | | | | | | | |
| GPG-Mean | 68.8 ± 0.76 | 57.0 ± 0.49 | 14.8 ± 1.07 | 27.7 ± 2.02 | 10.0 ± 2.69 | 43.3 ± 4.25 | 25.4 ± 0.29 | 35.3 ± 1.65 |
| GPG-Scaled | 70.2 ± 0.67 | 57.5 ± 0.22 | 15.5 ± 0.19 | 28.6 ± 0.53 | 10.0 ± 2.69 | 43.8 ± 7.07 | 26.4 ± 0.91 | 36.0 ± 1.75 |
| Gaussian Norm | 69.4 ± 0.46 | 57.8 ± 0.21 | 16.5 ± 2.36 | 28.6 ± 0.62 | 10.0 ± 2.69 | 48.3 ± 5.14 | 25.4 ± 0.66 | 36.6 ± 1.73 |
| <i>Constant Negative Reward</i> | | | | | | | | |
| $r_{neg} = -0.2$ | 73.2 ± 0.29 | 59.6 ± 0.22 | 18.5 ± 1.14 | 28.8 ± 0.70 | 11.1 ± 1.91 | 43.3 ± 2.89 | 27.6 ± 0.29 | 37.5 ± 1.06 |
| $r_{neg} = -0.5$ | 72.0 ± 0.96 | 58.8 ± 0.91 | 12.4 ± 1.52 | 28.2 ± 1.31 | 10.0 ± 3.30 | 46.7 ± 2.29 | 29.6 ± 0.40 | 36.8 ± 1.53 |
| $r_{neg} = -0.8$ | 72.8 ± 1.00 | 59.4 ± 0.76 | 17.6 ± 1.15 | 27.3 ± 1.06 | 10.0 ± 3.82 | 45.0 ± 2.90 | 28.6 ± 0.38 | 37.2 ± 1.58 |

Table 6: Performance comparison of reward methods across 7 mathematical reasoning benchmarks. Mean score (± standard deviation) over three runs is reported. Best results are **bolded**.

(3) **DPO exhibits greater robustness.** DPO achieves consistent improvements over the base model via pairwise preference learning (e.g. +8.5 on Qwen2.5-1.5B, +11.7 on Qwen2.5-7B, +8.0 on Qwen3-1.7B, trained on MATH), but its advantage over RFT diminishes with stronger base models. RIFT, in contrast, dominates across all scales by explicitly modeling reward signals, proving more effective than pair preference alignment alone.

(4) **Mixed-reward responses drive RIFT gains.** As Table 4 shows, on MATH the mixed-reward rate (rollouts with both correct and incorrect responses) drops with model scale (84.7% to 32.4%), and the gain of RIFT over RFT declines accordingly (+11.4 to +1.9). On NuminaMath, however, larger models yield higher mixed-reward rates (86.6% for Qwen3-1.7B) and the largest RIFT gains (+3.9), confirming that RIFT benefits most when correct and incorrect responses coexist.

Pass@8 Performance Table 3 reports Pass@8 (probability of more than 1 correct solution in 8 generations). RIFT consistently achieves the highest and most stable Pass@8 across all settings. (1) **RFT prioritizes correctness at the cost of solution diversity.** While RFT achieves competitive Mean@8, its Pass@8 consistently lags behind DPO, as RFT relies solely on correct rollouts, yielding high-quality but low-diversity solutions. In contrast, DPO achieves higher Pass@8 by contrasting

correct and incorrect outcomes, forcing the model to explore a wider strategy space. (2) **RIFT outperforms implicit pairwise comparisons.** RIFT further surpasses DPO in Pass@8 (+3.0 on Qwen-2.5-Math-1.5B, +1.4 on Qwen-2.5-Math-7B, and +1.2 on Qwen-3 1.7B), showing that the explicit use of the reward signal enables more effective exploration than implicit pairwise comparison.

4.3 Extending to Reasoner Model

To evaluate RIFT on models with intrinsic reasoning, we adopt DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025), a distilled reasoner that generates explicit reflective traces, unlike non-thinking models that depend on prompt-based thinking. Given the extended reasoning traces, we set the maximum length of self-generated responses to 8,192. As Table 5 shows, this setting reveals key alignment failures in baseline methods.

(1) **SFT breaks the built-in reasoning.** SFT on MATH severely degrades performance (21.3 vs. Base 43.0), indicating direct SFT on non-reflective data actively degrades the inherent capacity for step-by-step thinking of the reasoner model. (2) **RFT and DPO only approach base performance.** While RFT recovers Pass@8 (59.4), reaching levels comparable to the base model, it simultaneously degrades the Mean@8 (37.9 vs. Base 43.0). DPO, in contrast, maintains a comparable Mean@8 while achieving slightly higher Pass@8. (3) **RIFT deliv-**

| Method | Qwen-2.5-Math-1.5B | | DeepSeek-R1-Distill-Qwen-1.5B | | Qwen3-1.7B | |
|-------------|--------------------------|---------------------|-------------------------------|--------------------|--------------------------|--------------------|
| | Peak Memory usage (GB) ↓ | Acc (%) ↑ | Peak Memory usage (GB) ↓ | Acc (%) ↑ | Peak Memory usage (GB) ↓ | Acc (%) ↑ |
| SFT | 17.95 | 24.3 | 15.59 | 21.3 | 19.40 | 33.4 |
| DFT | 26.90 | 33.1 | 21.57 | 42.0 | 21.04 | 37.5 |
| RFT | 18.12 | 25.6 | 18.15 | 37.9 | 19.47 | 42.4 |
| DPO | 43.31 | 31.0 | 43.36 | 42.8 | 41.24 | 41.6 |
| RIFT | 20.10 (+1.98) | 37.0 (+11.4) | 22.28 (+4.13) | 44.0 (+6.1) | 22.00 (+2.53) | 44.3 (+1.9) |

Table 7: Computational efficiency and performance trade-off. Accuracy (Acc) represents the mean@8 score averaged across 7 mathematical benchmarks. (+) and (+) indicate the absolute improvement in Acc and the absolute increase in peak computational memory of RIFT compared to RFT.

ers robust gains. RIFT achieves the highest performance across all metrics: 44.0 Mean@8 (+1.0 over Base) and 63.2 Pass@8 (+3.8 over Base), representing the largest absolute improvement observed among all tested methods. Notably, RIFT significantly outperforms strong baselines like DPO by +1.2 in Mean@8 and +3.1 in Pass@8.

4.4 Generalization to General Preference Alignment

While RIFT is primarily motivated by tasks with verifiable rewards (e.g., mathematical reasoning), we further evaluate its generalizability on broader preference alignment. We fine-tune the Llama3.2-3B (Grattafiori et al., 2024) model on the Ultra-Feedback (Cui et al., 2024) dataset and compare RIFT against the standard DPO baseline across three diverse benchmarks: IFEval (Zhou et al., 2023b), TruthfulQA (Lin et al., 2022), and Hellaswag (Zellers et al., 2019).

As shown in Table 8, RIFT consistently outperforms DPO across all three benchmarks, demonstrating its effectiveness beyond verifiable reward tasks. Notably, RIFT achieves the improvement with approximately 50% less GPU memory usage than DPO. We observe that the performance gains on RLHF tasks are more modest compared to mathematical reasoning. We attribute this to the inherent noise in preference datasets (e.g., annotator bias, model-generated preferences), which provide less clean supervision signals than the deterministic rewards available in math tasks. Nevertheless, RIFT’s consistent improvements confirm its robustness even under noisy feedback conditions.

| Llama-3.2-3B | IFEval | TruthfulQA | Hellaswag | Avg. |
|--------------|--------------|--------------|--------------|--------------|
| Base | 12.23 | 19.42 | 52.32 | 27.99 |
| DPO | 18.34 | 22.12 | 53.74 | 31.40 |
| RIFT | 18.63 | 24.94 | 55.89 | 33.15 |

Table 8: Zero-shot evaluation on general preference alignment tasks. RIFT outperforms DPO while using ~50% less memory.

5 In-depth Analysis

5.1 Reward Strategy and Robustness Analysis

To assess how the reward design impacts the effectiveness of RIFT, we compare two distinct classes of reward strategies: (1) constant negative rewards ($r_{\text{neg}} \in \{-0.2, -0.5, -0.8\}$) and (2) group-wise normalization, which rescales rewards per problem based on its self-generated response set. We evaluate three normalization variants, \hat{r} represents the reward r after normalization:

- Gaussian Normalization: Standardizes rewards within each problem’s solution group:

$$\hat{r} = (r - \mu) / \sigma \quad (6)$$

where μ and σ are the mean and standard deviation of rewards for that problem.

- GPG Normalization (Mean-Centered): Adapts the advantage formulation of GPG (Chu et al., 2025b):

$$\hat{r} = \alpha \cdot (r - \mu), \quad \alpha = N^+ / N, \quad (7)$$

N^+ and N denote the numbers of correct and total responses in each group. The scaling factor α acts as an adaptive gain controller, amplifying the learning signal for problems with higher success rates.

- GPG Normalization (Raw-Scaled): Preserves the original reward sign and relative magnitude:

$$\hat{r} = \alpha \cdot r, \quad \alpha = N^+ / N. \quad (8)$$

Table 6 evaluates different reward mechanisms: **(1) Constant negative reward outperforms dynamic normalization.** Surprisingly, simple constant negative rewards consistently surpass group normalization methods, suggesting absolute reward can be more effective than relative intra-group rewards. **(2) RIFT is remarkably robust to negative reward magnitude.** Average performance remain stable within the $[-0.2, -0.8]$ range, indicating a consistent rejection signal enables stable and superior performance over RFT.

| Model | Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|-----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>Post-Train on MATH Dataset</i> | | | | | | | | | |
| Qwen-2.5-Math-1.5B (Mean@8) | Base | 42.6 | 35.6 | 9.7 | 22.6 | 7.1 | 31.9 | 8.2 | 22.5 |
| | SFT | 57.0 | 42.9 | 9.3 | 16.1 | 3.3 | 21.9 | 19.9 | 24.3 |
| | DFT | 76.8 | 53.9 | 15.6 | 19.1 | 4.2 | 25.6 | 36.4 | 33.1 |
| | RFT | 48.8 | 37.2 | 13.5 | 22.5 | 8.8 | 33.4 | 15.2 | 25.6 |
| | DPO | 61.8 | 50.3 | 11.3 | 26.7 | 7.1 | 41.2 | 18.6 | 31.0 |
| | SimPO | 65.8 | 50.0 | 18.0 | 18.4 | 5.8 | 20.6 | 35.0 | 30.5 |
| | KTO | 69.9 | 58.5 | 17.2 | 28.3 | 8.8 | 40.9 | 26.9 | <u>35.8</u> |
| | RIFT | 72.6 | 59.6 | 15.8 | 28.8 | 7.1 | 41.9 | 33.3 | 37.0 |
| Qwen-2.5-Math-1.5B (Pass@8) | Base | 88.0 | 75.1 | 32.0 | 46.5 | 23.3 | 67.5 | 30.1 | 51.8 |
| | SFT | 93.6 | 80.5 | 29.0 | 43.7 | 16.7 | 60.0 | 49.0 | 53.2 |
| | DFT | 94.5 | 76.7 | 38.2 | 43.0 | 20.0 | 55.0 | 52.4 | 54.3 |
| | RFT | 87.0 | 67.3 | 30.1 | 27.5 | 5.0 | 39.1 | 41.1 | 42.4 |
| | DPO | 92.9 | 83.2 | 33.8 | 48.7 | 33.3 | 72.5 | 45.4 | 58.5 |
| | SimPO | 94.5 | 81.8 | 40.8 | 45.6 | 16.7 | 62.5 | 55.5 | 56.8 |
| | KTO | 93.5 | 82.1 | 44.9 | 50.2 | 30.0 | 77.5 | 42.8 | <u>60.1</u> |
| | RIFT | 93.9 | 85.9 | 37.1 | 51.7 | 30.0 | 80.0 | 51.9 | 61.5 |

Table 9: Comparison with SimPO and KTO on Qwen2.5-Math-1.5B. Results show Mean@8 and Pass@8 accuracy (%) across 7 mathematical benchmarks.

| Qwen-2.5-Math-1.5B | SFT | DFT | RFT | DPO | SimPO | KTO | RIFT |
|--------------------------|--------------|-------|--------------|-------|-------|--------------|--------------|
| Peak Memory usage (GB) ↓ | 17.95 | 26.90 | <u>18.12</u> | 43.31 | 21.40 | 42.93 | 20.10 |
| Average Accuracy (%) ↑ | 24.30 | 33.10 | 25.60 | 31.00 | 30.50 | <u>35.80</u> | 37.00 |

Table 10: Efficiency and performance trade-off against SimPO and KTO on Qwen2.5-Math-1.5B. Results show peak GPU memory usage and average mathematical reasoning accuracy.

5.2 Computational Efficiency

We evaluate the computational efficiency of RIFT by measuring peak computational memory usage during training alongside average accuracy across seven benchmarks. As demonstrated in Table 7, RIFT maintains a highly favorable performance-efficiency trade-off across all backbones. Specifically, while DPO incurs substantial memory overhead (exceeding 41 GB) due to the necessity of loading a reference model, RIFT requires only 20.1 to 22.3 GB. This represents nearly a 50% reduction in peak VRAM usage compared to DPO, while consistently yielding higher accuracy (e.g., 44.3% vs. 41.6% on Qwen3-1.7B). Furthermore, the computational cost of RIFT is comparable to that of SFT and RFT, introducing only marginal overhead.

5.3 Comparison with Efficient Alignment Baselines

To further evaluate the efficiency and effectiveness of RIFT, we extend our comparison to include recent reference-free and low-memory alignment methods, specifically SimPO (Meng et al., 2024b) and KTO (Ethayarajh et al., 2024). We conduct experiments using the Qwen2.5-Math-1.5B model to assess both performance gains and computational overhead. As illustrated in Table 9 and 10, RIFT achieves a superior trade-off between mathemat-

ical reasoning performance and computational overhead. Although SimPO achieves memory savings similar to RIFT by removing the reference model, its performance gains over DPO are marginal, trailing RIFT by 6.5 points. Similarly, while KTO shows a notable improvement over DPO (+4.8 points), it still lags behind RIFT by 1.2 points. Crucially, unlike KTO which maintains a high memory usage, RIFT delivers superior performance with significantly higher resource efficiency.

6 Conclusion

We propose RIFT, a simple yet effective post-training framework that leverages the full distribution of self-generated samples. Unlike RFT, which discards valuable negative samples via hard thresholding, RIFT leverages both high- and low-reward trajectories. To ensure optimization stability, we introduce a principled loss formulation that effectively prevents training collapse during reward integration. Extensive evaluation across seven mathematical reasoning benchmarks shows that RIFT consistently outperforms established baselines. This demonstrates that explicitly learning from mixed-quality data allows models to better internalize correct reasoning and failure modes. As a robust and data-efficient alignment method, RIFT enables scalable self-improvement without reliance on extensive expert-labeled data.

Limitations

While RIFT demonstrates substantial gains in data efficiency and performance, several limitations remain for further refinement.

First, as a reward-informed framework, RIFT is designed to effectively bridge the gap between reward signals and policy optimization. While it maximizes the utility of self-generated feedback, the performance upper-bound is naturally influenced by the discriminative power of the reward source. This is a shared challenge across all reward-driven alignment methodologies. Our results show that RIFT is robust to mixed-quality data, and exploring uncertainty-aware reward weighting to further mitigate potential feedback noise remains a compelling direction.

Second, our evaluation primarily focuses on verifiable reasoning tasks characterized by objective success criteria and deterministic outcomes. Although mathematical benchmarks provide a high-fidelity environment to validate the core mechanics of RIFT, extending this framework to subjective or open-ended generative domains remains an open challenge. In such contexts, where correctness is harder to define, and the rewards are more difficult to measure, which might require a more complex reward setup.

Finally, RIFT currently treats each sample as a single unit. In complex, multi-step problems, a model might fail just because of one small mistake in a long, mostly correct path. At the moment, we do not look inside the steps to find these almost correct parts. Future versions of RIFT could use step-by-step rewards to learn from these partial successes, which could help the model improve even faster.

Regarding safety and ethical considerations, while the base models may occasionally generate reasoning errors, the practical risk is minimal as all model outputs are utilized strictly as internal training signals rather than for real-world deployment. Furthermore, all evaluations are conducted on standard mathematical benchmarks using objective metrics, ensuring a controlled experimental environment. In terms of manuscript preparation, an AI assistant is employed to enhance the linguistic clarity. However, all AI-generated suggestions are carefully reviewed and refined by the authors, ensuring that the final manuscript accurately reflects our own judgment and contains no harmful or misleading content.

References

- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. [Self-play fine-tuning converts weak language models to strong language models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025a. [SFT memorizes, RL generalizes: A comparative study of foundation model post-training](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025b. [GPG: A simple and strong reinforcement learning baseline for model reasoning](#). *CoRR*, abs/2504.02546.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024a. [Scaling instruction-finetuned language models](#). *J. Mach. Learn. Res.*, 25:70:1–70:53.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024b. [Scaling instruction-finetuned language models](#). *Journal of Machine Learning Research*, 25(70):1–53.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. 2024. [Ultrafeedback: Boosting language models with scaled ai feedback](#). In *Forty-first International Conference on Machine Learning*.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. [Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping](#). *Preprint*, arXiv:2002.06305.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. [Kto: Model alignment as prospect theoretic optimization](#). *arXiv preprint arXiv:2402.01306*.

- Tao Feng, Wei Li, Didi Zhu, Hangjie Yuan, Wendi Zheng, Dan Zhang, and Jie Tang. 2025. [ZeroFlow: Overcoming catastrophic forgetting is easier than you think](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024a. [Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1416–1428. Association for Computational Linguistics.
- Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, and 9 others. 2024b. [Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent AI](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. [On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. [Solving quantitative reasoning problems with language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Gaotang Li, Ruizhong Qiu, Xiusi Chen, Heng Ji, and Hanghang Tong. 2025. Beyond log likelihood: Probability-based objectives for supervised fine-tuning across the model capability continuum. *arXiv preprint arXiv:2510.00526*.
- Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. NuminaMath. [<https://huggingface.co/AI-MO/NuminaMath-CoT>](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. [An empirical study of catastrophic forgetting in large language models during continual fine-tuning](#). *CoRR*, abs/2308.08747.
- Mathematical Association of America. 2023. [2023 american mathematics competitions \(amc 10a/10b/12a/12b\)](#). Problems and official solutions available at <https://maa.org/math-competitions/amc-1012>.
- Mathematical Association of America. 2024. [2024 american invitational mathematics examination \(aime i\)](#). Problems and official solutions available at <https://maa.org/math-competitions>.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024a. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024b. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022a. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022b. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Chongli Qin and Jost Tobias Springenberg. 2025. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, and 21 others. 2022. [Multitask prompted training enables zero-shot task generalization](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galouédec. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.
- Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2025. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*.
- Sen Xu, Yi Zhou, Wei Wang, Jixin Min, Zhibin Yin, Yingwei Dai, Shixi Liu, Lianyu Pang, Yirong Chen, and Junlin Zhang. 2025. [Tiny model, big logic: Diversity-driven optimization elicits large-model reasoning ability in vibethinker-1.5b](#). *Preprint*, arXiv:2511.06221.
- 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, and 40 others. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024b. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. [Scaling relationship on learning mathematical reasoning with large language models](#). *Preprint*, arXiv:2308.01825.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and 1 others. 2023a. Instruction tuning for large language models: A survey. *ACM Computing Surveys*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023b. [Instruction tuning for large language models: A survey](#). *CoRR*, abs/2308.10792.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. [Swift:a scalable lightweight infrastructure for fine-tuning](#). *Preprint*, arXiv:2408.05517.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, An Yang, Jingren Zhou, and Junyang Lin. 2025. [Stabilizing reinforcement learning with llms: Formulation and practices](#). *arXiv preprint arXiv:2512.01374*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023a. [Lima: Less is more for alignment](#). *Advances in Neural Information Processing Systems*, 36:55006–55021.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. [Instruction-following evaluation for large language models](#). *arXiv preprint arXiv:2311.07911*.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. 2025. [Proximal supervised fine-tuning](#). *arXiv preprint arXiv:2508.17784*.

A Extended Analysis and Empirical Explorations

In this section, we provide additional empirical evidence and in-depth analyses to further elucidate the behavior of RIFT. Specifically, we investigate how the scale of self-generated data influences the alignment performance and examine the underlying dynamics of policy convergence facilitated by our framework.

A.1 Ablation Study on the Number of Self-Generated Responses

To evaluate how the quantity of self-generated data affects the alignment performance of RIFT, we conduct an ablation study by varying the number of sampled responses per problem (K). While the main experiments utilize $K = 8$, we investigate the performance across $K \in \{2, 4, 8, 16\}$, using Qwen-2.5-Math-1.5B trained on MATH. All other hyperparameters remain consistent with the main training setup.

As shown in Table 11, increasing K from 2 to 8 consistently improves both Mean@3 (+8.2 points) and Pass@3 (+6.4 points), with $K = 8$ achieving the highest scores across nearly all benchmarks. However, further increasing K to 16 leads to a noticeable drop particularly in Pass@3 (−3.0 points), suggesting that more samples do not always translate to better alignment.

The performance trend is intriguing because the underlying data statistics, reported in Tables 12 and 13, show little variation across K : the proportion of positive responses remains stable at around 66.4%, and the fraction of problems with mixed reward signals plateaus near 85% for $K \geq 4$. In other words, simply generating more responses does not significantly alter the overall composition of the training data.

The resolution lies in the quality of exploration, not just its quantity. While the aggregate statistics appear similar, larger K enables richer coverage of the solution space to capture a wider variety of reasoning patterns, including subtle failure modes. At $K = 8$, this diversity is sufficient for RIFT to learn robust distinctions between correct and flawed reasoning, without overwhelming the model with redundant or low-signal trajectories. By contrast, $K = 16$ introduces diminishing returns: the marginal gain in mixed-reward problems (from 84.7% to 92.3%) comes at the cost of increased noise, which disproportionately harms the reliabil-

| Model | Num. K | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|--------------------------------|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Qwen-2.5-Math-1.5B (Mean@3) | Base | 41.6 | 35.2 | 9.3 | 22.4 | 10.0 | 32.5 | 8.3 | 22.8 |
| | 2 | 57.7 | 46.4 | 11.3 | 25.1 | 11.1 | 39.2 | 14.2 | 29.3 |
| | 4 | 62.0 | 50.2 | 13.5 | 27.1 | 7.8 | 41.7 | 18.8 | 31.6 |
| | 8 | 73.2 | 59.6 | 18.5 | 28.8 | 11.1 | 43.3 | 27.6 | 37.5 |
| | 16 | 71.9 | 59.2 | 19.4 | 28.5 | 5.6 | 39.2 | 28.1 | 36.0 |
| Qwen-2.5-Math-1.5B (Pass@3) | Base | 69.0 | 58.1 | 13.5 | 37.2 | 20.0 | 52.5 | 17.8 | 38.3 |
| | 2 | 81.3 | 69.4 | 23.5 | 37.6 | 26.7 | 60.0 | 26.2 | 46.4 |
| | 4 | 83.2 | 71.4 | 27.6 | 41.5 | 20.0 | 62.5 | 31.1 | 48.2 |
| | 8 | 88.6 | 77.3 | 33.5 | 41.9 | 20.0 | 70.0 | 38.6 | 52.8 |
| | 16 | 88.6 | 76.6 | 32.0 | 41.2 | 13.3 | 57.5 | 38.3 | 49.8 |

Table 11: Ablation study on the number of self-generated responses per problem K in RIFT training for Qwen-2.5-Math-1.5B. For each $K \in \{2, 4, 8, 16\}$, we sample K responses per problem from the base model. We report Mean@3 and Pass@3 accuracy (%) across 7 mathematical benchmarks.

ity of top predictions, as reflected in the sharper decline of Pass@3.

| K | # Num. | # Total | # Pos. (r>0) | # Neg. (r<0) | % Pos. |
|-----|--------|---------|--------------|--------------|--------|
| 2 | 3,000 | 6,000 | 3,992 | 2,008 | 66.5% |
| 4 | 3,000 | 12,000 | 7,964 | 4,036 | 66.4% |
| 8 | 3,000 | 24,000 | 15,941 | 8,059 | 66.4% |
| 16 | 3,000 | 48,000 | 31,943 | 16,057 | 66.5% |

Table 12: Statistics of self-generated rollouts for 3,000 problems from the MATH dataset, sampled using the Qwen-2.5-Math-1.5B base model with $K \in \{2, 4, 8, 16\}$ responses per problem.

As shown in Table 12, the overall proportion of positive samples (% Pos.) remains remarkably consistent at approximately 66.5% as K increases from 2 to 16. More importantly, Table 13 shows that the percentage of problems with mixed-reward responses (containing both positive and negative outcomes) scales with K , rising from 78.5% to 92.3%. This trend demonstrates that a larger K effectively augments intra-problem diversity.

| K | # Num. | # Mixed-Reward Num. | % Mixed |
|-----|--------|---------------------|---------|
| 2 | 3,000 | 2,356 | 78.5% |
| 4 | 3,000 | 2,548 | 84.9% |
| 8 | 3,000 | 2,541 | 84.7% |
| 16 | 3,000 | 2,768 | 92.3% |

Table 13: Proportion of problems containing both positive- and negative-reward responses in their K self-generated responses.

The performance drop when increasing from $K = 8$ to $K = 16$ may stem from overfitting to noisy negative patterns in the limited MATH dataset. With more self-generated samples, the model encounters repeated or low-quality negatives, potentially leading to spurious correlations.

This is consistent with observations in rejection sampling, where excessive sampling can degrade performance.

A.2 Exploration Study: RIFT Drives Policy Convergence

To further evaluate whether RIFT serves as a modular enhancement or constitutes an indispensable component in the training pipeline, we conduct a comparison study using Qwen-2.5-Math-1.5B trained on MATH, under three distinct protocols: (i) SFT followed by RIFT, (ii) DFT followed by RIFT, and (iii) Iterative RIFT (RIFT followed by RIFT). In each setting, the previously trained SFT, DFT, or RIFT models serves as the base policy to generate a new corpus of self-generated responses. These responses are then utilized to perform a subsequent round of RIFT training. The central hypothesis is that if RIFT functions as a plug-and-play refiner, performance should vary with initialization quality; conversely, if RIFT itself drives capability gains, final performance should converge across initialization strategies.

As shown in Table 14, the results support the hypothesis that RIFT itself drives capability gains. Across all previously trained models, applying a second round of RIFT yields consistent improvements in Mean@3, outperforming their single-round counterparts: notable gains of +11.5 (SFT), +5.6 (DFT), and +1.2 (RIFT), even when starting from a strong RIFT-initialized policy. Gains diminish as the base policy strengthens, indicating convergence toward a shared high-reward policy. However, the re-application of RIFT primarily boosts Mean@3 rather than Pass@3, which shows little to no further improvement compared to single-round trained RIFT. Together, these findings demonstrate

| Model | Method | GSM8K | MATH | Minerva | Olympiad | AIME24 | AMC23 | College | Avg. |
|--------------------------------|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------------|
| Qwen-2.5-Math-1.5B (Mean@3) | Base | 41.6 | 35.2 | 9.3 | 22.4 | 10.0 | 32.5 | 8.3 | 22.8 |
| | SFT | 56.3 | 43.3 | 11.6 | 16.5 | 8.9 | 32.5 | 20.3 | 27.1 |
| | + RIFT | 74.7 | 62.9 | 18.5 | 29.4 | 8.9 | 39.2 | 36.8 | 38.6 (+11.5) |
| | DFT | 77.1 | 54.4 | 16.3 | 19.1 | 2.2 | 25.8 | 35.9 | 33.0 |
| | + RIFT | 74.7 | 61.8 | 18.3 | 30.4 | 4.4 | 45.0 | 35.5 | 38.6 (+5.6) |
| | RIFT | 73.2 | 59.6 | 18.5 | 28.8 | 11.1 | 43.3 | 27.6 | 37.5 |
| | + RIFT | 74.8 | 62.3 | 17.5 | 29.8 | 10.0 | 42.5 | 34.1 | 38.7 (+1.2) |
| Qwen-2.5-Math-1.5B (Pass@3) | Base | 69.0 | 58.1 | 13.5 | 37.2 | 20.0 | 52.5 | 17.8 | 38.3 |
| | SFT | 84.2 | 67.1 | 23.2 | 31.6 | 16.7 | 60.0 | 36.6 | 45.6 |
| | + RIFT | 90.0 | 78.7 | 34.6 | 42.4 | 13.3 | 60.0 | 47.4 | 52.3 (+6.7) |
| | DFT | 89.9 | 68.4 | 29.0 | 31.1 | 6.7 | 40.0 | 45.5 | 44.4 |
| | + RIFT | 87.9 | 78.5 | 29.4 | 44.3 | 13.3 | 70.0 | 46.7 | 52.9 (+8.5) |
| | RIFT | 88.6 | 77.3 | 33.5 | 41.9 | 20.0 | 70.0 | 38.6 | 52.8 |
| | + RIFT | 88.8 | 79.3 | 28.7 | 44.0 | 16.7 | 65.0 | 45.8 | 52.6 (-0.2) |

Table 14: Mean@3 and Pass@3 accuracy (%) on 7 mathematical benchmarks for Qwen-2.5-Math-1.5B under different sequential training protocols, starting from models trained via SFT, DFT, or RIFT, we generate self-sampled responses and apply a second round of RIFT (denoted “+ RIFT”). Best results are in **bold**. (+) indicates the absolute improvement over the respective single-phase baseline.

| K | # Num. | # Total | # Pos. ($r > 0$) | # Neg. ($r < 0$) | % Pos. |
|------|--------|---------|-----------------------|-----------------------|--------|
| SFT | 3,000 | 24,000 | 8,300 | 15,700 | 65.4% |
| DFT | 3,000 | 24,000 | 7,964 | 16,859 | 70.2% |
| RIFT | 3,000 | 24,000 | 15,941 | 16,287 | 67.9% |

Table 15: Statistics of self-generated responses across previously trained SFT, DFT and RIFT models.

| K | # Num. | # Mixed-Reward Num. | % Mixed |
|------|--------|---------------------|---------|
| SFT | 3,000 | 2,686 | 89.5% |
| DFT | 3,000 | 2,697 | 89.9% |
| RIFT | 3,000 | 2,669 | 89.0% |

Table 16: Proportion of problems containing both positive- and negative-reward responses in self-generated responses across previously trained SFT, DFT and RIFT models.

that RIFT acts not as a plug-and-play refiner whose efficacy depends on initialization, but as a self-convergent alignment phase that capable of steering diverse initial policies toward comparable final performance. Tables 15 and 16 provide a detailed statistical overview of the self-generated responses produced by the previously trained SFT, DFT, and RIFT models.

B Discussion on Loss Function and Gradient Stability

In this section, we provide a detailed analysis of our choice of using a Taylor approximation for the loss function, particularly in the context of penalizing negative samples.

B.1 Design Intuition: Stability over Global Precision

The primary motivation for employing a Taylor-based linear penalty (i.e., $(1 - p)$) rather than the standard negative log-likelihood ($-\log(1 - p)$) for negative samples lies in the *bounded and stable gradients* it provides across the entire probability range $[0, 1]$. While the Taylor approximation is locally accurate near $p \rightarrow 1$, its global behavior is empirically more conducive to stable optimization in model alignment.

For negative trajectories, our design intuition follows a specific priority:

- **Confident Errors:** Tokens assigned high probability ($p \rightarrow 1$) on negative samples indicate cases where the model is “confidently wrong.” These require strong, linear penalties to steer the model away from incorrect reasoning paths.
- **Correct Uncertainty:** Tokens with low probability ($p \rightarrow 0$) on negative examples reflect appropriate uncertainty. In such cases, the linear approximation provides a gentler correction signal, preventing the model from over-correcting on already unlikely tokens.

B.2 Comparison with Alternative Formulations

We evaluated several alternative functions to penalize negative samples, but found them less suitable for the following reasons:

- **Logarithmic Loss** ($-\log(1 - p)$): This formulation diverges as $p \rightarrow 1$. In practical training, this leads to gradient instability and

a significant magnitude imbalance between positive and negative loss components, often resulting in training divergence.

- **Sigmoid-based Variants:** While sigmoid transformations can bound the loss, they introduce unnecessary saturation regions (where gradients vanish) without offering clear performance benefits in our preliminary experiments.

In summary, the choice of the Taylor approximation in RIFT is empirically motivated by its ability to maintain robust gradient signals and prevent optimization instability. We acknowledge that exploring other functional forms with similar bounded properties remains a promising direction for future work.

B.3 Proofs and Theoretical Analysis in Section 3

Proof of Theorem 3.2. Let the dataset \mathcal{D} be finite. The naive signed-weighted loss function can be explicitly written as:

$$\mathcal{L}_{\text{naive}}(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} r(x,y) \log \pi_{\theta}(y | x). \quad (9)$$

By the theorem’s premise, the subset of negative samples $\mathcal{D}^- = \{(x,y) \in \mathcal{D} \mid r(x,y) < 0\}$ is non-empty. Let $(x_0, y_0) \in \mathcal{D}^-$ be a specific negative sample with weight $r_0 := r(x_0, y_0) < 0$.

We invoke the assumption of *sufficient expressivity*, which implies that the model parameterization θ allows for the arbitrary manipulation of the probability mass $\pi_{\theta}(\cdot | x_0)$ on the support \mathcal{Y} . Specifically, we can construct a sequence of parameters $\{\theta_n\}_{n=1}^{\infty}$ such that the probability of the negative sample decays to zero:

$$\pi_{\theta_n}(y_0 | x_0) = \epsilon_n, \text{ where } \epsilon_n > 0 \text{ and } \lim_{n \rightarrow \infty} \epsilon_n = 0. \quad (10)$$

To ensure the well-posedness of the remaining terms, we stipulate that the probability mass removed from y_0 is redistributed to other tokens $y' \in \mathcal{Y} \setminus \{y_0\}$ uniformly, such that for all other samples $(x,y) \in \mathcal{D} \setminus \{(x_0, y_0)\}$, the probabilities satisfy $\pi_{\theta_n}(y | x) \geq \delta$ for some constant $\delta > 0$. This ensures that $\log \pi_{\theta_n}(y | x)$ remains bounded from below.

Then, we have

$$\lim_{n \rightarrow \infty} \left| \frac{\partial \mathcal{L}_{\text{naive}}}{\partial \pi_{\theta_n}} \right| = \left| \frac{r}{\pi_{\theta_n}(y|x)} \right| = \infty. \quad (11)$$

Now, we decompose the loss function for the sequence θ_n :

$$\begin{aligned} \mathcal{L}_{\text{naive}}(\theta_n) &= -\frac{1}{|\mathcal{D}|} \underbrace{r_0 \log \epsilon_n}_{T_1} \\ &\quad - \frac{1}{|\mathcal{D}|} \underbrace{\sum_{(x,y) \in \mathcal{D} \setminus \{(x_0, y_0)\}} r(x,y) \log \pi_{\theta_n}(y | x)}_{T_2}. \end{aligned} \quad (12)$$

We analyze the asymptotic behavior of the two terms T_1 and T_2 as $n \rightarrow \infty$:

1. **The Negative Sample Term (T_1):** Since $r_0 < 0$ and $\lim_{n \rightarrow \infty} \log \epsilon_n = -\infty$, the product behaves as:

$$\begin{aligned} \lim_{n \rightarrow \infty} T_1 &= \lim_{n \rightarrow \infty} r_0 \log \epsilon_n \\ &= (-|r_0|) \cdot (-\infty) = +\infty. \end{aligned} \quad (13)$$

2. **The Remaining Terms (T_2):** The sum T_2 consists of a finite number of terms.

- For any sample with $r(x,y) \geq 0$, since $\pi_{\theta_n}(y|x) \leq 1$, we have $r(x,y) \log \pi_{\theta_n}(y|x) \leq 0$. Thus, these terms are bounded from above by 0.
- For any sample with $r(x,y) < 0$, since we enforced $\pi_{\theta_n}(y|x) \geq \delta$, the term $r(x,y) \log \pi_{\theta_n}(y|x)$ is finite.

Crucially, since we ensure other probabilities do not vanish (i.e., $\pi \geq \delta$), the logarithm $\log \pi$ is bounded below by $\log \delta$. Consequently, the entire sum T_2 is bounded, i.e., there exists a constant M such that $|T_2| < M$.

Combining these results, the limit of the total loss is:

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathcal{L}_{\text{naive}}(\theta_n) &= -\frac{1}{|\mathcal{D}|} \left(\lim_{n \rightarrow \infty} T_1 + \lim_{n \rightarrow \infty} T_2 \right) \\ &= -\frac{1}{|\mathcal{D}|} (+\infty + O(1)) \\ &= -\infty. \end{aligned} \quad (14)$$

Thus, we have identified a sequence in the parameter space along which the loss diverges to negative infinity. This proves that $\mathcal{L}_{\text{naive}}$ is unbounded from below. \square

First, we give a formal reformulation of Theorem 3.4 as follows.

Theorem B.1 (Formal Statement of Theorem 3.4). *The RIFT formulation satisfies the following theoretical properties:*

- (i) **Boundedness:** Assume that the reward r is bounded with constant $|r| \leq M$ for all the data. Then, the loss objective is bounded from below.
- (ii) **Reward Lower-Bound Maximization:** Let the expected reward objective be $\mathcal{J}(\theta) := \mathbb{E}_{y \sim \pi_\theta}[r(x, y)]$. Assume that all the data sampled from the reference distribution π_{ref} has a lower bound probability, i.e., $\pi_{ref}(y|x) \geq C_2$. Then, there exists a constant $C_1 > 0$, such that $\mathcal{J}(\theta) \geq -\frac{1}{C_2} \mathcal{L}_{RIFT}(\theta) + C_1$. Thus, minimizing the RIFT loss objective is equivalent to maximize the reward objective.

Proof of Theorem 3.4. (i) Boundedness: Recall the formulation of the RIFT loss:

$$\mathcal{L}_{RIFT}(\theta) = -\mathbb{E}_{\mathcal{D}^+}[r(x, y) \log \pi_\theta(y|x)] + \mathbb{E}_{\mathcal{D}^-}[r(x, y) \pi_\theta(y|x)]. \quad (15)$$

We analyze the boundedness of the two terms separately.

For the positive sampled term, we have

$$-\mathbb{E}_{\mathcal{D}^+}[r(x, y) \log \pi_\theta(y|x)] \geq 0, \quad (16)$$

since $\log \pi_\theta \leq 0$ for the probability distribution.

For the negative sampled term, the absolute value is bounded by

$$\begin{aligned} & |-\mathbb{E}_{\mathcal{D}^-}[r(x, y) \pi_\theta(y|x)]| \\ & \leq \mathbb{E}_{\mathcal{D}^-}[|r(x, y) \pi_\theta(y|x)|] \\ & \leq \mathbb{E}_{\mathcal{D}^-}[|r(x, y)|] \leq M. \end{aligned} \quad (17)$$

Hence, the second term is bounded from below.

Thus, \mathcal{L}_{RIFT} is bounded from below.

(ii) **Reward Lower-Bound Maximization:** We aim to show that maximizing the negative RIFT loss (i.e., minimizing \mathcal{L}_{RIFT}) is equivalent to maximizing a surrogate lower bound of the expected reward $\mathcal{J}(\theta)$.

First, we rewrite the expected reward objective using Importance Sampling (IS) to shift the expectation from the policy distribution π_θ to the reference data distribution π_{ref} :

$$\begin{aligned} \mathcal{J}(\theta) &= \mathbb{E}_{y \sim \pi_\theta}[r(x, y)] \\ &= \mathbb{E}_{y \sim \pi_{ref}} \left[\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} r(x, y) \right]. \end{aligned} \quad (18)$$

Let $\rho(y|x) = \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$ be the likelihood ratio. We utilize the fundamental inequality relating linear and logarithmic functions: for any $u > 0$, $\log u \leq u - 1$, which implies $u \geq 1 + \log u$.

We decompose the objective into contributions from positive (\mathcal{D}^+) and negative (\mathcal{D}^-) domains:

Case 1: Positive Samples ($r > 0$). Applying the inequality $\rho \geq 1 + \log \rho$:

$$\begin{aligned} \mathbb{E}_{\mathcal{D}^+}[\rho \cdot r] &\geq \mathbb{E}_{\mathcal{D}^+}[r(1 + \log \rho)] \\ &= \mathbb{E}_{\mathcal{D}^+}[r(1 + \log \pi_\theta - \log \pi_{ref})] \\ &= \mathbb{E}_{\mathcal{D}^+}[r \log \pi_\theta] + C_1, \end{aligned} \quad (19)$$

where $C_1 = \mathbb{E}_{\mathcal{D}^+}[r(1 - \log \pi_{ref})]$ is a constant with respect to θ . This recovers the positive component of $-\mathcal{L}_{RIFT}$.

Case 2: Negative Samples ($r < 0$). For negative samples, we seek a lower bound for the term $\rho \cdot r$. Note that (x, y) are data sampled from the distribution π_{ref} , there exists a constant $C_2 > 0$, such that $\pi_{ref}(y|x) \geq C_2$ for all (x, y) . Thus,

$$\mathbb{E}_{\mathcal{D}^-}[\rho \cdot r] \geq \frac{1}{C_2} \mathbb{E}_{\mathcal{D}^-}[\pi_\theta(y|x) r(x, y)] \quad (20)$$

Synthesis: Combining the results, we define a global surrogate objective \mathcal{J}_{surr} (the IS-derived logarithmic lower bound for positive samples and the linear lower bound for negative samples):

$$\begin{aligned} \mathcal{J}(\theta) &\geq \mathbb{E}_{\mathcal{D}^+}[r \log \pi_\theta] + \frac{1}{C_2} \mathbb{E}_{\mathcal{D}^-}[r \pi_\theta] + C_1 \\ &\geq -\frac{1}{C_1} \mathcal{L}_{RIFT}(\theta) + C_2. \end{aligned} \quad (21)$$

Therefore, maximizing $-\mathcal{L}_{RIFT}$ (or minimizing \mathcal{L}_{RIFT}) effectively maximizes a rigorous surrogate lower bound of the true expected reward $\mathcal{J}(\theta)$. \square