ON THE GENERALIZATION OF SFT: A REINFORCEMENT LEARNING PERSPECTIVE WITH REWARD RECTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we present a simple yet theoretically motivated improvement to Supervised Fine-Tuning (SFT) for the Large Language Model (LLM), addressing its limited generalization compared to reinforcement learning (RL). Through mathematical analysis, we reveal that standard SFT gradients implicitly encode a problematic reward structure that may severely restrict the generalization capabilities of model compared to RL. To rectify this, we propose Dynamic Fine-Tuning (DFT), stabilizing gradient updates for each token by dynamically rescaling the objective function with the probability of this token. With just a single-line change, the method outperforms standard SFT on multiple difficult benchmarks and base models, from math reasoning to code generation and multi-modal tasks, demonstrating improved generalization. Additionally, DFT achieves competitive results in offline RL settings, and further boosts the effectiveness of subsequent RL training, providing an effective yet streamlined alternative. The experiments further demonstrate that DFT not only strengthens SFT performance but also consistently improves the effectiveness of subsequent RL training. By bridging theoretical insights with practical solutions, this work advances the state of SFT. The source code will be publicly released.

1 INTRODUCTION

Supervised Fine-Tuning (SFT), which adapts models to expert demonstrations, has become the standard post-training paradigm for Large Language Models (LLMs). It enables efficient task adaptation and capability enhancement (Chung et al., 2024; Zhang et al., 2024b; Sanh et al., 2022; Ouyang et al., 2022), and is popular for its ease of implementation and rapid acquisition of expert-like behaviors (Wei et al., 2022; Zhou et al., 2023). Despite these advantages, SFT often shows limited generalization compared to reinforcement learning (RL) (Chu et al., 2024; Ouyang et al., 2022; Christiano et al., 2017; Bai et al., 2022; Huan et al., 2025; Swamy et al., 2025). RL leverages explicit reward or verification signals to explore diverse strategies and thus generalizes better. However, RL requires substantial computation, careful hyperparameter tuning, and explicit reward signals—conditions often impractical in real-world settings (Schulman et al., 2017; Ouyang et al., 2022; Sheng et al., 2025; Strubell et al., 2019; Liu & Yin, 2024; Winsta, 2025). Moreover, RL can struggle to recover expert-like behaviors that SFT captures efficiently (Mandlekar et al., 2022; Chen et al., 2025b).

To exploit the complementary strengths of both approaches, many hybrid methods combine SFT with RL (Ouyang et al., 2022; Sheng et al., 2025; Rafailov et al., 2023; Liu et al., 2025; Qiu et al., 2025). Yet a key question remains: can SFT itself be fundamentally improved? This is crucial, as SFT remains the only viable option when datasets contain only positive demonstrations, with no negative samples or reward model available.

In this work, we address this gap with a mathematical analysis of the connection between SFT and RL. We show that the gradient update in SFT can be interpreted as a form of policy gradient with a specific, implicitly defined reward under certain assumptions. Crucially, this reward is (i) sparse, and (ii) inversely proportional to the model's probability of expert actions (see equation 6). As a result, when the model assigns low probability to expert actions, the gradient becomes excessively large, yielding an ill-posed reward structure and unstable optimization (Pascanu et al., 2013; Yang et al., 2019).

Building on this insight, we propose Dynamic Fine-Tuning (DFT), a principled fix. Our method rescales the SFT objective at each token by its probability, canceling the distortion introduced by inverse-probability weighting. This reframing turns the SFT gradient from a potentially unstable and biased estimator into a more stable, more uniformly weighted update rule that behaves closer to an RL-style.

Empirically, DFT delivers substantial improvements. On the Qwen-2.5-Math series (Qwen Team et al., 2024b) fine-tuned with NuminaMath-CoT (LI et al., 2024), DFT yields gains several times larger than standard SFT. More importantly, unlike SFT, which often degrades on challenging benchmarks such as OlympiadBench (He et al., 2024), AIME 2024 (American Institute of Mathematics, 2024), and AMC 2023 (Mathematical Association of America, 2023), our method consistently improves performance and generalization. These improvements hold across models, scales, and data sizes (Table 1, Figure 1), and extend to code generation and multimodal reasoning (Tables 3, 4).

We further test DFT in off-policy RL settings (Table 2), where dense rewards are available (Levine et al., 2020). Our method not only outperforms offline RL approaches such as DPO (Rafailov et al., 2023) and RAFT (Dong et al., 2023; Ahn et al., 2024), but also achieves competitive or superior performance to online methods like GRPO and PPO on math tasks with Qwen2.5-Math-1.5B. Unlike these RL methods, DFT requires neither a reference model nor large batch sizes, making it a simpler and more resource-efficient alternative. Besides, our experiments further show that DFT not only yields stronger SFT performance, but also reliably enhances the effectiveness of subsequent RL training.

To understand its effect, we analyze token probability distributions after training (Figure 2). While traditional SFT uniformly pushes probabilities toward the training set, DFT selectively increases some while reducing others. In particular, the proportion of less strongly fitted tokens rises, suggesting improved regularization. We provide further discussion in Appendix A.3.

The contributions of this work are theoretical and practical. On the theoretical side, we mathematically establish LLM SFT as a special RL in policy gradient space, pinpoint the underlying reasons for the limited generalization of SFT, and derive a method to improve it. On the experimental side, we show that such a simple solution, just one line of code, can enhance the performance and generalization capabilities of SFT across various tasks and models.

2 Related Work

The trade-off between supervised fine-tuning (SFT) and reinforcement learning (RL) is central to the alignment of large language models. SFT is widely adopted due to its simplicity and efficiency in imitating expert demonstrations (Chung et al., 2024; Zhou et al., 2023; Wei et al., 2022), analogous to behavioral cloning in robotics (Sammut, 2011; Mandlekar et al., 2022). However, the literature consistently highlights its limitations, particularly the tendency to overfit and generalize poorly compared to RL, which leverages reward signals to discover more robust policies (Ouyang et al., 2022; Christiano et al., 2017; Bai et al., 2022; Swamy et al., 2025; Zhang et al., 2025). A recent systematic comparison by Chu et al. (2024) across textual and visual domains confirms this distinction, concisely summarized as "SFT memorizes while RL generalizes." They further show that SFT remains indispensable as an initialization step, stabilizing output formatting prior to effective RL training. Nonetheless, RL faces significant practical hurdles, including computational expense, sensitivity to hyperparameters, and the requirement of an explicit reward function, all of which constrain its applicability (Schulman et al., 2017; Strubell et al., 2019; Sheng et al., 2025).

To combine the strengths of both paradigms, much recent work has pursued hybrid approaches. The most common strategy involves SFT pretraining followed by RL-based refinement with a learned reward model, as popularized by InstructGPT (Ouyang et al., 2022). More recent methods interleave SFT and RL updates to improve stability and performance (Sheng et al., 2025; Liu et al., 2025; Qiu et al., 2025). Other approaches, such as Direct Preference Optimization (DPO) (Rafailov et al., 2023), bypass reward modeling entirely by directly optimizing policies on preference data, thereby unifying imitation and reinforcement signals within a single loss function. Chen et al. (2025a) introduce Negative-aware Fine-Tuning (NFT), which models incorrect generations via an implicit negative policy, enabling self-improvement without explicit feedback. While powerful, these methods rely on reward signals, preference pairs, or negative samples. They enrich the training pipeline but do not fundamentally improve SFT in its native setting, where only positive demonstrations are available. Our work instead focuses on enhancing SFT itself without requiring external feedback.

A complementary line of theoretical research seeks to unify SFT and RL under a common formalism. Du et al. (2025) reinterpret RLHF as a reward-weighted variant of SFT, preserving reliance on an explicit reward. Wang et al. (2025) show that SFT can be cast as RL with an implicit reward, proposing adjustments such as smaller learning rates to manage the vanishing KL constraint. Abdolmaleki et al. (2025) analyze learning from both positive and negative feedback, studying how their balance affects convergence. Qin & Springenberg (2025) view SFT as a lower bound of RL and introduce importance weighting based on the data-generating policy. While these works establish connections between SFT and RL through weighting, they do not provide a precise mathematical equivalence between the SFT gradient and the offline policy gradient. Some methods approximate this connection in practice by reweighting training losses. For instance, MixCE (Zhang et al., 2023) combines the forward and reverse KL divergences to form a unified objective, while GOLD (Pang & He, 2021) adopts offline RL with demonstrations, introducing reliance on an unknown demonstration distribution π_b and a restrictive $1/N$ assumption. Kantharaju & Sankar (2022) also provide a clear and insightful exposition of GOLD's motivation and mechanics from an alternative perspective, offering useful intuition for understanding its underlying design. In contrast, our work offers a more formal perspective on this connection, highlighting the role of the inverse-probability weighting term in shaping the difference between SFT and RL-like updates. This perspective motivates a simple adjustment: multiplying the loss by the model's token probability to neutralize the weighting.

Interestingly, our method modifies the standard cross-entropy (CE) loss in a way that inverts the weighting philosophy of the widely used Focal Loss (Lin et al., 2017). Specifically, our modified CE takes the form $-p\log(p)$, whereas focal loss is defined as $-(1-p)^\gamma\log(p)$. Focal Loss deliberately downweights well-classified samples to emphasize underrepresented or hard cases, whereas we deliberately downweight poorly classified samples to encourage generalization. This inversion reflects a fundamental shift in the LLM era: while underfitting was once a central challenge, overfitting and memorization now dominate, demanding a rethinking of objective design.

3 Method

3.1 Preliminaries

Supervised Fine-Tuning. Let $\mathcal{D} = \{(x, y^\star)\}$ denote a corpus of expert demonstrations, where $y^\star$ is the complete reference response to the query x. SFT minimizes the sentence-level cross-entropy:

$$\mathcal{L}_{\text{SFT}}(\theta) \;=\; \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\big[-\log \pi_\theta\big(y^\star \mid x\big)\big]. \tag{1}$$

Its gradient is:

$$\nabla_\theta \mathcal{L}_{\text{SFT}}(\theta) \;=\; \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\big[-\nabla_\theta \log \pi_\theta\big(y^\star \mid x\big)\big]. \tag{2}$$

Reinforcement Learning. Let y denote a response sampled from the policy $\pi_\theta(\cdot \mid x)$ for query x. Given a reward function $r(x, y) \in \mathbb{R}$, the policy objective is

$$J(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{D}_x,\; y\sim\pi_\theta(\cdot|x)}\big[r(x,y)\big]. \tag{3}$$

Its policy gradient at the sentence level is

$$\nabla_\theta J(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{D}_x,\; y\sim\pi_\theta(\cdot|x)}\big[\nabla_\theta \log \pi_\theta(y \mid x)\; r(x,y)\big]. \tag{4}$$

3.2 Unify SFT and RL Gradient Expression

Rewriting SFT Gradient as Policy Gradient via Importance Sampling. The SFT gradient in equation 2 is taken under the *fixed* demonstration distribution. We convert it to an on-policy expectation by inserting an importance weight that compares the expert (Dirac Delta) distribution with the model distribution.

$$\mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\left[-\nabla_\theta \log \pi_\theta\big(y^\star \mid x\big)\right] = \mathbb{E}_{x\sim\mathcal{D}_x}\; \underbrace{\mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \frac{\mathbf{1}[y=y^\star]}{\pi_\theta(y \mid x)} \left[-\nabla_\theta \log \pi_\theta\big(y \mid x\big)\right]}_{\text{resample + reweight}} \tag{5}$$

Define the auxiliary variables (importance sampling weight) as

$$w(y \mid x) = \frac{\mathbf{1}}{\pi_\theta(y \mid x)}, \quad r(x, y) = \mathbf{1}[y = y^\star].$$

Reorganizing equation 5 and rewriting it using the above auxiliary variables, we obtain the form

$$\nabla_\theta \mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\big[w(y \mid x)\, \nabla_\theta \log \pi_\theta(y \mid x)\, r(x, y)\big]. \tag{6}$$

This form of the SFT gradient closely resembles the policy gradient in Equation equation 4. Under this formulation, conventional SFT can be interpreted as an on-policy gradient method, where the reward is a sparse indicator function matching the expert trajectory, but biased by an importance weighting term $1/\pi_\theta$. We emphasize that this RL-style characterization serves solely as a theoretical lens: both the analysis and subsequent modifications are developed within the RL framework, while the final method remains fully implementable in standard SFT form for computational efficiency. Detailed derivations are provided in Appendix A.2.

Due to the inherently sparse reward signal in the SFT setting, we identify the importance weight $1/\pi_\theta$ as a key contributor to SFT's generalization limitations compared to RL. When the model assigns low probability to the expert response, the resulting weight becomes excessively large, introducing an ill-posed reward landscape. This leads to disproportionately large gradients and training instability. The issue is compounded by the fact that the reward function $r(x, y) = \mathbf{1}[y = y^\star]$ is non-zero only for exact matches to the expert outputm causing optimization to overfit rare exact-match samples and weakening the model's ability to generalize beyond the training data.

3.3 Proposed Method

Reward Rectification via Dynamic Reweighting. To neutralize the skewed reward issue identified when viewing SFT under the RL objective, we dynamically reweight the reward by multiplying by a corrective inverse ratio given by the policy probability $1/w$. The resulting "dynamically fine-tuned" gradient is then

$$\nabla_\theta \mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\big[\text{sg}(\frac{1}{w}) \cdot w(y \mid x)\, \nabla_\theta \log \pi_\theta(y \mid x)\, r(x, y)\big]. \tag{7}$$

where $\text{sg}(\cdot)$ denotes the stop gradient operator, ensuring that gradients do not flow through the reward scaling term w. To facilitate transition to later equations, we directly write $1/w$ to be $\pi_\theta(y^\star \mid x)$ instead of $\pi_\theta(y \mid x)$ because the indicator function in equation 5 or equation 6 would leave all cases where $y \neq y^\star$ is 0. Now since the gradient does not flow, the corrected SFT loss also becomes a simple reweighted loss, called Dynamic Fine-tuning (DFT).

$$\mathcal{L}_{\text{DFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[-\,\text{sg}\left(\pi_\theta(y^\star \mid x)\right) \log \pi_\theta(y^\star \mid x)\Big]. \tag{8}$$

However, in practice, computing importance weights over the entire trajectory can induce numerical instability. A common treatment of this issue is to simply apply importance sampling at the token level, as was adopted in PPO (Schulman et al., 2017). This leads to the final DFT loss version:

$$\mathcal{L}_{\text{DFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[-\sum_{t=1}^{|y^\star|} \text{sg}\left(\pi_\theta(y_t^\star \mid y_{<t}^\star, x)\right) \log \pi_\theta(y_t^\star \mid y_{<t}^\star, x)\Big]. \tag{9}$$

Note that the reward of this corrected SFT (in RL form), i.e., DFT, now becomes 1 uniformly for all expert trajectory. This is akin to contemporary verification based reward approach RLVR (DeepSeek-AI et al., 2025) that assigns uniform reward to all correct samples. Consequently, it avoids over-concentration on specific low-probability reference tokens, leading to more stable updates and improved generalization without introducing any additional sampling or reward models.

Table 1: Average@16 accuracy of five state-of-the-art large language models on mathematical reasoning benchmarks. The best performance of each model across benchmarks is bold.

	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
LLaMA-3.2-3B	1.63	1.36	1.01	0.41	1.56	1.19
LLaMA-3.2-3B w/SFT	8.65	2.38	2.06	0.00	3.13	3.24
LLaMA-3.2-3B w/DFT	**12.79**	**2.84**	**2.90**	**0.83**	**3.91**	**4.65**
LLaMA-3.1-8B	1.86	0.98	0.94	0.21	1.01	1.00
LLaMA-3.1-8B w/SFT	16.85	5.78	3.88	0.00	5.16	6.33
LLaMA-3.1-8B w/DFT	**27.44**	**8.26**	**6.94**	**0.41**	**12.03**	**11.02**
DeepSeekMath-7B	6.15	2.15	1.74	0.21	2.97	2.64
DeepSeekMath-7B w/SFT	26.83	7.26	6.33	0.41	8.28	9.82
DeepSeekMath-7B w/DFT	**41.46**	**16.79**	**15.00**	**1.24**	**16.25**	**18.15**
Qwen2.5-Math-1.5B	31.66	8.51	15.88	4.16	19.38	15.92
Qwen2.5-Math-1.5B w/SFT	43.76	13.04	12.63	1.87	18.75	18.01
Qwen2.5-Math-1.5B w/DFT	**64.89**	**20.94**	**27.08**	**6.87**	**38.13**	**31.58**
Qwen2.5-Math-7B	40.12	14.39	17.12	6.68	27.96	21.25
Qwen2.5-Math-7B w/SFT	53.96	16.66	18.93	2.48	26.09	23.62
Qwen2.5-Math-7B w/DFT	**68.20**	**30.16**	**33.83**	**8.56**	**45.00**	**37.15**

4 Experiments

We design four groups of experiments to comprehensively evaluate DFT. We first study the standard SFT setting on mathematical reasoning tasks to establish its core advantage over SFT (Section 4.1). We then extend to an offline RL setting, comparing DFT with representative offline and online RL methods (Section 4.2). To test cross-domain robustness, we further examine DFT on code generation benchmarks (Section 4.3) and its applicability to multi-modal reasoning math datasets (Section 4.4).

4.1 Main Experiment - Mathematical Reasoning Task

To examine whether DFT can outperform vanilla SFT across tasks, architectures, and scales, we use mathematical reasoning as a representative testbed.

DFT consistently yields average performance improvements over base models compared to standard SFT across all benchmarks. Table 1 shows that, for Qwen2.5-Math-1.5B, DFT achieves an average gain of +15.66 points over the base model, which is over 5.9× larger than the +2.09 point improvement from SFT. This pattern generalizes across other model families and sizes: LLaMA-3.2-3B benefits from a +3.46 point gain with DFT, exceeding the SFT gain (+2.05) by approximately 1.4×; LLaMA-3.1-8B achieves +10.02 from DFT, surpassing SFT's +5.33 by 1.88×; DeepSeekMath-7B sees a +15.51 point improvement via DFT, which is 1.58× larger than SFT's +7.18; and Qwen2.5-Math-7B reaches a +15.90 point gain, nearly 3.8× higher than the SFT improvement of +2.37.

DFT demonstrates generalization and robustness, especially on challenging benchmarks where standard SFT yields minimal or even negative impact. For instance, on Olympiad Bench, SFT degrades performance for Qwen2.5-Math-1.5B, dropping accuracy from 15.88 to 12.63, while DFT boosts it to 27.08, +11.20 point improvement over base model. On AIME24, SFT reduces accuracy for Qwen2.5-Math-7B by 4.20 points (from 6.68 to 2.48), whereas DFT improves performance to 8.56, achieving a +1.88 point gain over the base model despite the difficulty of the benchmark. A similar trend is observed on AMC23. SFT reduces the performance of Qwen2.5-Math-1.5B from 19.38 to 18.75, while DFT raises it to 38.13, a +18.75 point gain over base. For Qwen2.5-Math-7B, SFT yields only a marginal improvement (+1.86), whereas DFT achieves a +17.04 point gain. These results underscore that DFT not only scales more effectively across models of varying capacities, but also exhibits better resilience on difficult reasoning tasks where traditional SFT struggles.

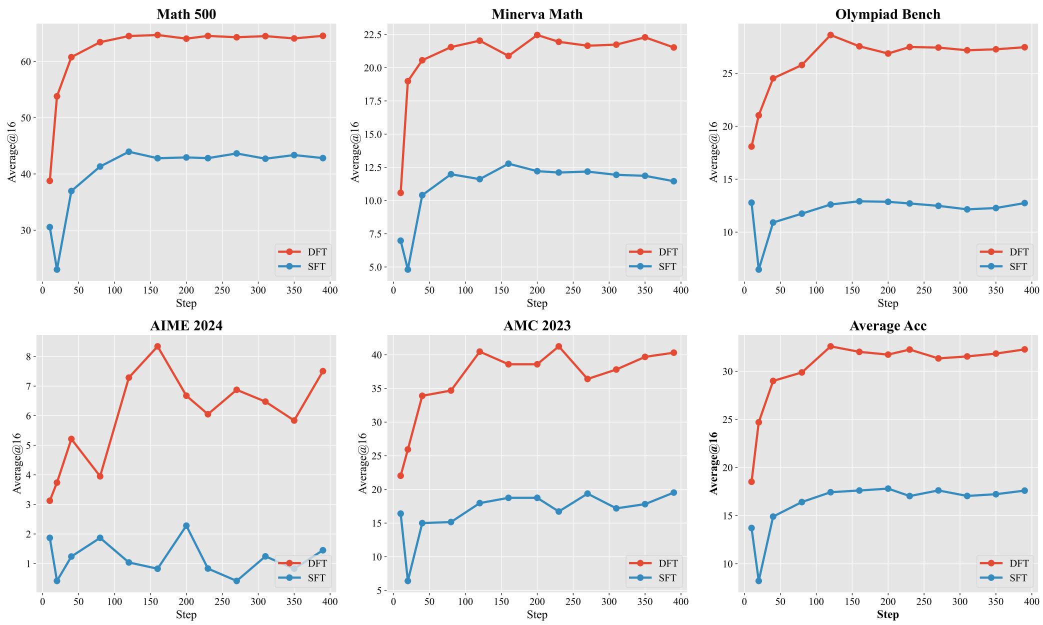

Figure 1: Accuracy progression for Qwen2.5-Math-1.5B across mathematical benchmarks, illustrating faster convergence and better performance achieved by DFT relative to SFT.

DFT exhibits better learning efficiency and faster convergence characteristics. Figure 1 reveals clear differences in learning dynamics between DFT and standard SFT on Qwen2.5-Math-1.5B across all math reasoning benchmarks. Compared to SFT, our method demonstrates three distinct advantages: (1) Faster convergence, achieving peak performance within the first 120 training steps on most benchmarks; (2) Better early-stage performance, with DFT already outperforming best final accuracy of SFT within the first 10–20 steps; and (3) Higher sample efficiency, consistently requiring fewer updates to reach relatively optimal results. This accelerated convergence shows that the dynamic reweighting mechanism in DFT leads to more informative gradient updates, guiding the model toward high-quality solutions early in training. It also suggests that DFT helps avoid the optimization plateaus or noise-prone regions often encountered in standard SFT, thereby enabling more efficient acquisition of complex mathematical reasoning patterns.

We also report the results of parameter-efficient fine-tuning (PEFT) training setting (Hu et al., 2022) and training on the OpenR1-Math dataset (Hugging Face, 2025) with better quality in Appendix A.8 and Appendix A.7, respectively. Comparison and Discussion with the concurrent method iw-SFT (Qin & Springenberg, 2025) is provided in Appendix A.6.

4.2 Exploratory Experiment - Offline RL Setting

Equation 7 shows that SFT suffers from reward sparsity, since in a constructed dataset each query x has only a single reference answer $y^\star$. From the perspective of RL, RFT/RAFT (Dong et al., 2023; Ahn et al., 2024) can be viewed as alleviating the sparse reward issue by effectively increasing reward density, thereby enhancing model performance. Motivated by this observation, we conduct an exploratory study applying DFT in an offline RL setting, where the reward sparsity problem is inherently less severe compared to standard SFT, to further validate the effectiveness.

DFT demonstrates competitive performance in the offline RL setting, outperforming both offline and online RL baselines. Table 2 shows DFT achieves an average score of 35.43, exceeding the best offline method RFT by +11.46 points, and even outperforming the strongest online RL algorithm GRPO by +3.43 points. Specially, on Math500, DFT scores 64.71, slightly ahead of GRPO (62.86) and better than PPO (56.10) and RFT (48.23). The gains are also notable on more challenging benchmarks: on AMC23, DFT achieves 48.44, a +7.19 point margin over GRPO and a +17.66 point gain over RFT. Similarly, on Minerva Math, DFT reaches 25.16, outperforming GRPO by +6.23 points, PPO by +9.75, and all offline baseline methods.

Table 2: Evaluation results on mathematical reasoning benchmarks in an offline reinforcement learning setting using reward signals from rejection sampling. The best performance is in bold.

	Setting	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
Qwen2.5-Math-1.5B w/DFT	SFT	64.89	20.94	27.08	6.87	38.13	31.58
Qwen2.5-Math-1.5B w/DPO	Offline	46.89	11.53	22.86	4.58	30.16	23.20
Qwen2.5-Math-1.5B w/RFT	Offline	48.23	14.19	22.29	4.37	30.78	23.97
Qwen2.5-Math-1.5B w/PPO	Online	56.10	15.41	26.33	7.50	37.97	28.66
Qwen2.5-Math-1.5B w/GRPO	Online	62.86	18.93	28.62	**8.34**	41.25	32.00
Qwen2.5-Math-1.5B w/DFT	Offline	**64.71**	**25.16**	**30.93**	7.93	**48.44**	**35.43**

Table 3: Performance of various models on code generation benchmarks. The best performance for each benchmark is highlighted in bold.

	HumanEval		MultiPL-E								
	HE	HE+	Python	C++	Java	PHP	TS	C#	Bash	JS	Avg.
Qwen2.5-3B	43.3	36.0	43.29	40.99	37.34	37.89	**47.17**	**43.04**	24.68	45.96	40.05
Qwen2.5-3B w/SFT	41.5	34.8	42.07	42.24	37.97	37.27	43.40	41.77	20.25	**47.83**	39.10
Qwen2.5-3B w/DFT	**45.7**	**39.0**	**45.73**	**44.72**	**41.77**	**45.34**	42.14	**43.04**	**27.85**	44.10	**41.84**
Qwen2.5-Coder-3B	52.4	42.7	51.83	53.42	46.20	47.20	54.09	55.06	25.32	**54.04**	48.39
Qwen2.5-Coder-3B w/SFT	51.8	43.9	51.22	51.55	48.10	54.66	**59.12**	51.27	**34.18**	**54.04**	50.52
Qwen2.5-Coder-3B w/DFT	**56.7**	**50.0**	**57.32**	**54.66**	**51.27**	**58.39**	58.49	**60.76**	31.01	53.42	**53.16**
Qwen2.5-Coder-7B	62.2	53.0	63.41	63.98	53.16	59.01	62.89	59.49	39.24	60.87	57.76
Qwen2.5-Coder-7B w/SFT	54.9	48.8	54.88	64.60	51.27	**62.11**	68.55	60.76	33.54	**65.22**	57.62
Qwen2.5-Coder-7B w/DFT	**67.7**	**59.8**	**67.68**	**67.70**	**54.43**	60.87	**70.44**	**65.19**	**48.73**	63.35	**62.30**

These results highlight the strength of DFT as a simple yet effective fine-tuning strategy. Despite its lack of iterative reward modeling or environment interaction, it provides a stronger learning signal than both offline methods like DPO/RFT and online policy optimization algorithms like PPO/GRPO in certain scale train set. This suggests that DFT can serve as a more efficient and scalable alternative to traditional RL pipelines, particularly in domains where preference supervision is available but reward modeling or online response sampling is expensive or impractical.

4.3 Exploratory Experiment - Code Generation Task

Table 3 shows DFT achieves improvements in most cases compared to both base models and SFT. For Qwen2.5-3B, DFT raises HumanEval from 43.3 to 45.7 and HumanEval+ from 36.0 to 39.0, with the MultiPL-E average also increasing from 40.05 (base) and 39.10 (SFT) to 41.84. Similar trends are observed for Qwen2.5-Coder-3B, where DFT improves HumanEval to 56.7 and HumanEval+ to 50.0, outperforming both base and SFT. For Qwen2.5-Coder-7B, DFT reaches 67.7 on HumanEval, 59.8 on HumanEval+, and 62.3 average on MultiPL-E, surpassing SFT by +12.8, +11.0, and +4.7 points respectively. The overall trend demonstrates that DFT generally provides stronger performance across different models and languages.

4.4 Exploratory Experiment - Multi-Modal Reasoning

DFT achieves consistent improvements over base models and SFT across all multi-modal reasoning benchmarks. Table 4 shows, on MathVerse, DFT boosts Qwen2.5-VL-3B from 33.83 to 37.54 average accuracy, outperforming the SFT gain of only +1.83 by +3.71 points. Consistent improvements are observed across all major vision-related subcategories. On MathVision, DFT improves performance from 21.25 (base) to 22.30, exceeding SFT which fails to provide gains (21.02). On WeMath, SFT already yields a +19.23 point gain, but DFT pushes performance slightly further to 23.71, maintaining superiority over both base and SFT. These results indicate that DFT not only strengthens text-only reasoning but also extends effectively to multi-modal domains.

Table 4: Performance comparison across different multi-modal reasoning benchmarks. The best performance on each benchmark is highlighted in bold.

	MathVerse				MathVision	WeMath
	Vision Only	Vision Intensive	Vision Dominant	Overall		
Qwen2.5-VL-3B	28.81	30.96	31.60	33.83	21.25	4.10
Qwen2.5-VL-3B w/SFT	30.96	33.63	32.74	35.66	21.02	23.33
Qwen2.5-VL-3B w/DFT	**32.49**	**35.91**	**33.50**	**37.54**	**22.30**	**23.71**

Table 5: Performance comparison on mathematical reasoning benchmarks under cold-start settings. All models are initialized via fine-tuning (SFT or DFT) and further optimized with GRPO.

	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
Qwen2.5-Math-1.5B w/SFT+GRPO	62.54	23.10	26.92	5.00	40.15	31.54
Qwen2.5-Math-1.5B w/DFT+GRPO	**65.96**	**23.51**	**28.37**	**8.63**	**41.40**	**33.57**

4.5 Can DFT Enhance Reinforcement Learning?

To further investigate the role of DFT in RL optimization, we conduct a set of exploratory experiments where models are first initialized with either SFT or DFT, and then fine-tuned using GRPO.

Mathematical Reasoning. As shown in Table 5, DFT+GRPO consistently outperforms SFT+GRPO across all benchmarks. Improvements are moderate on Math500 and Minerva Math, but become substantial on harder datasets such as Olympiad Bench (+1.45) and AIME24 (+3.63). The average score rises from 31.54 to 33.57.

Code Generation. Table 6 shows that DFT+GRPO yields strong gains on HumanEval (+11.6) and HumanEval+ (+10.4), and improves performance across most MultiPL-E languages, raising the average from 58.15 to 62.61 compared to SFT.

Multi-modal Reasoning. As reported in Table 7, DFT+GRPO surpasses SFT+GRPO across all MathVerse subsets and achieves a notable +4.76 improvement on WeMath, demonstrating that DFT also enhances RL optimization in multimodal settings.

These results indicate that DFT not only improves performance in SFT but also consistently enhances the effectiveness of subsequent RL training. This validates DFT as a stronger pretraining strategy in RL pipelines across diverse tasks.

4.6 Limitations of DFT: A Case Study on Factual Knowledge

While DFT consistently outperforms SFT on reasoning-heavy tasks, it may not always be the better choice, particularly in factual knowledge domains. We conduct an exploratory experiment on the Natural Questions dataset (Kwiatkowski et al., 2019), which consists of real-user, open-domain factual queries grounded in Wikipedia articles.

In this setting, we find that SFT improves performance from 31.24% to 36.62%, while DFT unexpectedly reduces it to 30.14%. This result reveals an important limitation of DFT: because it reweights samples based on the model's own confidence, it tends to reinforce the model's existing beliefs. When the model lacks sufficient factual knowledge, such reinforcement may hinder effective learning instead of facilitating it.

This case suggests that DFT is most effective when the task aligns well with the model's prior competence, such as logical reasoning or structured prediction. In contrast, when the objective is to absorb new factual information, especially in domains beyond the model's current capabilities, SFT remains a more reliable and stable fine-tuning strategy.

Table 6: Code generation performance on HumanEval and MultiPL-E benchmarks. All models are fine-tuned with GRPO after either SFT or DFT initialization.

	HumanEval		MultiPL-E								
	HE	HE+	Python	C++	Java	PHP	TS	C#	Bash	JS	Avg.
Qwen2.5-Coder-3B w/SFT+GRPO	57.3	50.6	57.32	63.35	51.27	**63.98**	68.55	60.76	33.54	**66.46**	58.15
Qwen2.5-Coder-3B w/DFT+GRPO	**68.9**	**61.0**	**68.90**	**67.08**	**55.06**	62.73	**70.44**	**65.19**	**49.37**	62.11	**62.61**

Table 7: Multi-modal reasoning performance comparison with GRPO initialized via SFT or DFT.

	MathVerse				MathVision	WeMath
	Vision Only	Vision Intensive	Vision Dominant	Overall		
Qwen2.5-VL-3B w/SFT+GRPO	32.48	33.50	43.78	35.93	21.44	21.43
Qwen2.5-VL-3B w/DFT+GRPO	**34.64**	**37.31**	**37.06**	**39.06**	**23.35**	**26.19**

Table 8: Comparison of weighting strategies on mathematical reasoning benchmarks.

	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
Qwen2.5-Math-1.5B	31.66	8.51	15.88	4.16	19.38	15.92
Sentence-Level Weighting	31.26	8.05	16.47	3.12	19.84	15.75
Geometric-Mean Weighting	42.87	12.34	13.03	1.23	16.56	17.21
Token-Level Weighting	**64.89**	**20.94**	**27.08**	**6.87**	**38.13**	**31.58**

4.7 An Empirical Comparison with Sentence-Level Weighting

Our framework applies confidence-based weighting at the token level. While this design was primarily motivated by numerical stability, we also compared it against two sentence-level variants to better understand their behavior.

The first variant uses the full sequence probability to scale the loss. However, these values are extremely small in practice, making the loss nearly uninformative and producing a highly skewed weight distribution that is difficult to tune. To address this, we also evaluated a geometric-mean variant inspired by GSPO (Zheng et al., 2025), which rescales sentence probabilities to avoid numerical collapse. Although this version is more stable, it still provides a weak training signal and offers limited performance gains.

As shown in Table 8, both sentence-level strategies lead to minimal changes over the base model, while our token-level formulation delivers substantial and consistent improvements, raising average accuracy from 15.92 to 31.58. These results demonstrate that token-level weighting provides a more reliable optimization signal and significantly stronger empirical performance.

4.8 Analysis of Probabilities

To understand how the model trained by DFT is different from SFT and other RL methods, we look into the token probability distribution of the model's output over the training set in Figure 2. SFT tends to uniformly increase token probabilities, shifting the entire distribution towards higher confidence, but mainly targeting the lower and lowest probability tokens. The highest probability token portion barely increases. In stark contrast, DFT exhibits a polarizing effect: it significantly boosts the probabilities of a subset of tokens while actively suppressing the probabilities of others. This leads to a bimodal distribution, with more tokens occupying both the highest and lowest probability bins. Other RL methods such as DPO, GPPO and PPO show the same trend as DFT, although the scale is much milder than it. We look into the words that belong to the lowest probability bin, and find that they are generally the conjunctive words or punctuations such as 'the', 'let', ',', '.' etc. These results suggest that for robust learning, models should not attempt to fit all tokens with uniform confidence. It may be beneficial to deprioritize fitting tokens that serve grammatical functions rather than carrying primary semantic content. This concept is analogous to human peda-

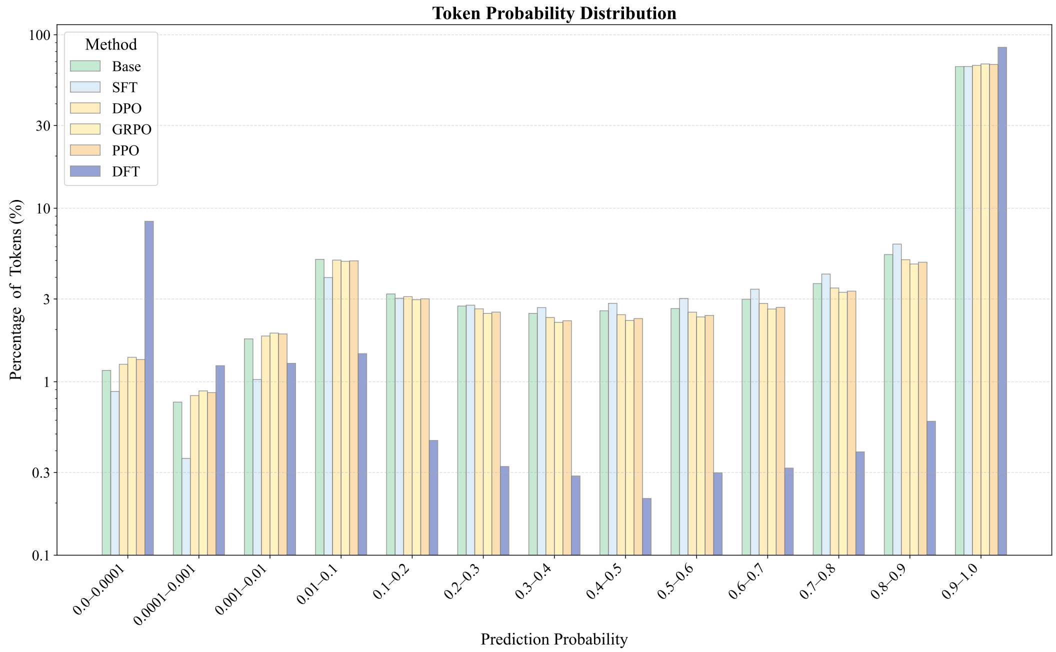

Figure 2: Token probability distributions on the training set before training and after fine-tuning with DFT, SFT, and various RL methods. A logarithmic scale is used on the y-axis for clarity.

gogy, where students are taught to focus on substantive concepts rather than perfecting the usage of common connective words. Further analysis can be found in Appendix A.3.

5 Conclusion

In this work, we revisit the well-known generalization gap between SFT and RL. We offer a theoretical perspective showing that the standard SFT gradient can be interpreted as a policy gradient with an ill-posed, implicitly defined reward inversely related to model confidence. This formulation helps explain the instability and limited generalization observed in SFT training. Motivated by this analysis, we introduce DFT, a simple yet effective method that dynamically reweights the SFT loss using the token probability. This one-line change improves gradient stability and leads to better generalization. Our empirical results show that DFTconsistently improves over standard SFT across a range of models and challenging mathematical reasoning tasks. Beyond supervised settings, we adapt DFTto offline RL scenarios and find that it outperforms several established online and offline RL baselines, suggesting broader applicability. Moreover, DFTalso enhances the performance of subsequent RL fine-tuning when used as a warm start. Overall, this work contributes both a refined understanding of SFT's limitations and a lightweight, practical method that helps bridge the gap to more complex RL-based approaches.

Limitations. While our experiments demonstrate the effectiveness of DFT on mathematical reasoning benchmarks and code generation tasks, the evaluation scope remains limited. We have not yet assessed its performance on broader task categories or with larger-scale LLM, which we leave for future exploration. Moreover, DFT can not offer universal benefits across all scenarios. In domains that primarily involve the acquisition of factual knowledge, conventional SFT still remains the most efficient approach. DFT may also not be an ideal choice for hard examples or domains under-represented in the training data, since it assigns low initial probabilities to such samples, reducing their learning weight. Our aim is not to assert that DFT universally outperforms SFT, but rather to offer a new perspective on objective design by analyzing the distinction between RL and SFT. Besides, an important future direction is to explore non-uniform or quality-aware reward assignments for demonstrations.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our study does not involve human subjects, personally identifiable information, or proprietary data. All datasets used, including NuminaMath, OpenR1-Math, UltraFeedback, and WeThink, are publicly available and documented in the appendix. The proposed method is a simple training strategy that modifies gradient computation for improved generalization. It does not introduce any new capabilities that could cause harm, nor does it enable misuse beyond the standard capabilities of existing large language models. We are not aware of any potential risks related to bias, fairness, or security that arise specifically from the method proposed. Nonetheless, we acknowledge that like any fine-tuning strategy, DFT may inherit biases present in the underlying data or model, and future research may explore safeguards for these scenarios. No conflicts of interest, legal compliance issues, or sponsorship-related influences are present in this work.

REPRODUCIBILITY STATEMENT

We have taken multiple steps to ensure the reproducibility of our work. All datasets used in our experiments are publicly available and properly cited in the main text and appendix. Training configurations, including model architectures, hyperparameters, optimizers, and evaluation settings, are described in detail in Section 4 and Appendi A.7-A.8. Theoretical claims, including the equivalence between SFT and policy gradient, are formally derived in Appendix A.2. Experimental results include multiple model scales, tasks, and training settings to validate robustness. A complete implementation of our method is included in the supplementary material, along with scripts for reproducing all reported results. We will release the full source code and training logs upon publication to further support reproducibility.

REFERENCES

Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Michael Bloesch, Rishabh Joshi, Thomas Lampe, Junhyuk Oh, Nicolas Heess, et al. Learning from negative feedback, or positive feedback or both. In *ICLR*, 2025. 3

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *EACLW*, pp. 225–237, 2024. 2, 6, 18

American Institute of Mathematics. Aime 2024 competition mathematical problems, 2024. 2, 17

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Dasgupta, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Hase, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 1, 2

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, et al. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691, 2023. 18

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. Bridging supervised learning and reinforcement learning in math reasoning. *arXiv preprint arXiv:2505.18116*, 2025a. 2

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 18

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025b. 1

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, volume 30, 2017. 1, 2

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In *ICML*, 2024. 1, 2

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 1, 2

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: boosting language models with scaled ai feedback. In *ICML*, pp. 9722–9744, 2024. 18

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 4, 19

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023, 2023. 2, 6, 18

Yilun Du et al. Simplify rlhf as reward-weighted sft: A variational method. *arXiv preprint arXiv:2502.11026*, 2025. 3, 18

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *ACM MM*, pp. 11198–11201, 2024. 18

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 17

Yoav Freund. A more robust boosting algorithm. *arXiv preprint arXiv:0905.2138*, 2009. 17

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 *ACL*, pp. 3828–3850, 2024. 2, 17

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021. 17

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 6

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025. 1

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. 6, 19

Pavan Kantharaju and Aiswarya Sankar. An understanding of learning from demonstrations for neural text generation. In *ICLR Blog Track*, 2022. URL `https://iclr-blog-track.github.io/2022/03/25/text-gen-via-lfd/`. https://iclr-blog-track.github.io/2022/03/25/text-gen-via-lfd/. 3

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. 8

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. 2

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. *NeurIPS*, 35:3843–3857, 2022. 17

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. Numinamath, 2024. 2, 17, 18

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2999–3007, 2017. 3

Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. 2025. 18

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *NeurIPS*, 36:21558–21572, 2023. 18

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. *arXiv preprint arXiv:2505.16984*, 2025. 1, 2

Vivian Liu and Yiqiao Yin. Green ai: exploring carbon footprints, mitigation strategies, and trade offs in large language model training. *Discover Artificial Intelligence*, 4(49), 2024. 1

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *CoRL*, pp. 1678–1690, 2022. 1, 2

Mathematical Association of America. Amc 2023 competition problems, 2023. 2, 17

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. *NeurIPS*, 35:27730–27744, 2022. 1, 2

Richard Yuanzhe Pang and He He. Text generation by learning from demonstrations. In *ICLR*, 2021. 3

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pp. 1310–1318. Pmlr, 2013. 2

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024. 18

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*, 2025. 3, 6, 18

Haibo Qiu, Xiaohan Lan, Fanfan Liu, Xiaohu Sun, Delian Ruan, Peng Shi, and Lin Ma. Metis-rise: Rl incentivizes and sft enhances multimodal reasoning model learning. *arXiv preprint arXiv:2506.13056*, 2025. 1, 2

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5: A party of foundation models. *arXiv preprint arXiv:2412.15115*, 2024a. 17

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b. 2

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, volume 36, 2023. 1, 2, 18

Claude Sammut. *Behavioral Cloning*. 2011. 2

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutton, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *ICLR*, 2022. 1

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *ICLR*, 2020. 17

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 2, 4, 18

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 17, 18

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 *ECCV*, pp. 1279–1297, 2025. 1, 2, 17, 18

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *ACL*, pp. 3645–3650, 2019. 1, 2

Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*, 2025. 1, 2

Bo Wang, Qinyuan Cheng, Runyu Peng, Rong Bao, Peiji Li, Qipeng Guo, Linyang Li, Zhiyuan Zeng, Yunhua Zhou, and Xipeng Qiu. Implicit reward as the bridge: A unified view of sft and dpo connections. *arXiv preprint arXiv:2507.00018*, 2025. 3

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *NeurIPS*, 2024. 18

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. 1, 2

Jenis Winsta. The hidden costs of ai: A review of energy, e-waste, and inequality in model development. *arXiv preprint arXiv:2507.09611*, 2025. 1

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025. 19

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. In *ICLR*, 2019. 2

Jie Yang, Feipeng Ma, Zitian Wang, Dacheng Yin, Kang Rong, Fengyun Rao, and Ruimao Zhang. Wethink: Toward general-purpose vision-language reasoning via reinforcement learning. *arXiv preprint arXiv:2506.07905*, 2025a. 18

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025b. 18

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025. 2

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, pp. 169–186, 2024a. 18

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2024b. 1

Shiyue Zhang, Shijie Wu, Ozan Irsoy, Steven Lu, Mohit Bansal, Mark Dredze, and David Rosenberg. Mixce: Training autoregressive language models by mixing forward and reverse cross-entropies. In *ACL*, pp. 9027–9050, 2023. 3

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. Swift:a scalable lightweight infrastructure for fine-tuning, 2024. 18

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. 9

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *ACL*, pp. 400–410, 2024. 18

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jianfeng Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. In *NeurIPS*, volume 36, 2023. 1, 2