On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification

Yongliang Wu[1*]   **Yizhou Zhou**[2*†]   **Zhou Ziheng**[3]   **Yingzhe Peng**[1]   **Xinyu Ye**[4]
Xinting Hu[5]   **Wenbo Zhu**[6]   **Lu Qi**[7]   **Ming-Hsuan Yang**[8]   **Xu Yang**[1‡]

[1]Southeast University   [2]Independent Researcher   [3]University of California, Los Angeles
[4]Shanghai Jiao Tong University   [5]Nanyang Technological University
[6]University of California, Berkeley   [7]Wuhan University   [8]University of California, Merced
`yongliang0223@gmail.com, zyz0205@hotmail.com, xuyang_palm@seu.edu.cn`

Abstract

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. 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. Remarkably, this single-line code change significantly outperforms standard SFT across multiple challenging benchmarks and base models, demonstrating greatly improved generalization. Additionally, our approach shows competitive results in offline RL settings, offering an effective yet simpler alternative. This work bridges theoretical insight and practical solutions, substantially advancing SFT performance. The code will be available at `https://github.com/yongliang-wu/DFT`.

1 Introduction

Supervised Fine-Tuning (SFT), which involves training a model on expert demonstration datasets, has become a standard Large Language Model (LLM) post-training approach for adapting models to new tasks or enhancing existing capabilities (Chung et al., 2024; Zhang et al., 2024; Sanh et al., 2022; Ouyang et al., 2022). Its widespread adoption is largely due to its ease of implementation and rapid acquisition of expert-like behaviors (Wei et al., 2022; Zhou et al., 2023). However, despite these advantages, SFT typically suffers from limited generalization compared to reinforcement learning (RL) methods (Chu et al.; 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, allowing the model to explore diverse strategies, thereby achieving stronger generalization. Nonetheless, RL approaches often require significant computational resources, are sensitive to hyperparameter tuning, and rely on the availability of reward signals, conditions that are not always feasible in practice (Schulman et al., 2017; Ouyang et al., 2022; Sheng et al., 2025; Strubell et al., 2019; Liu & Yin, 2024; Winsta, 2025). Even when RL is viable, SFT remains advantageous in rapidly acquiring expert behavioral patterns that RL may struggle to discover independently (Mandlekar et al., 2022; Chen et al., 2025b).

To leverage the complementary strengths of both approaches, numerous hybrid methods have been developed by incorporating SFT and RL (Ouyang et al., 2022; Sheng et al., 2025; Rafailov et al., 2023; Liu et al., 2025; Qiu et al., 2025). *However, can we fundamentally improve SFT itself?* This is crucial since SFT is the only viable approach when there are no negative samples in the dataset and no reward or verification model available.

In this work, we address this gap by providing a mathematical analysis that elucidates the fundamental differences between SFT and RL. We demonstrate that the gradient update of SFT can

---

*Equal Contribution. †Project Leader. ‡Corresponding Author.

be interpreted as a special case of a policy gradient method with a specific, implicitly defined reward structure. Our analysis reveals that *this implicit reward is both extremely sparse and inversely proportional to the policy's assigned probability of expert actions* (see Equation 6 for precise math details). Consequently, this leads to an ill-posed reward structure, especially when the model assigns low probabilities to expert actions; the resulting gradient experiences unbounded variance, creating a pathological optimization landscape.

Building on this mathematical analysis, we propose *Dynamic Fine-Tuning* (DFT), a principled solution that addresses the root cause of the implicitly ill-posed reward structure. *For each token, our method simply rescales the standard SFT objectives with the token probability*, effectively neutralizing the inverse probability weighting that leads to an unexpected reward structure and unbounded variance. This theoretically motivated modification fundamentally transforms the gradient estimator from an unstable, biased and probability-dependent mechanism into a stable, uniformly-weighted update procedure.

The empirical results of our method are highly significant. For example, by fine-tuning the Qwen-2.5-Math models (Qwen Team et al., 2024b) on the NuminaMath dataset (LI et al., 2024), the improvement with our approach over the baseline is generally multiple times greater than that of improvement by SFT. Crucially, when standard SFT experiences performance degradation on challenging datasets including Olympiad Bench(International Mathematical Olympiad, 2024), AIME 2024 (American Institute of Mathematics, 2024), and AMC 2023 (Mathematical Association of America, 2023), likely due to overfitting, our method consistently deliver substantial improvements, highlighting its more robust generalization capabilities. The result is validated across model types, sizes, and data sizes (Table 1, Figure 1).

In addition, we explore the applicability of our method within RL scenarios (see Table 3, where negative samples or dense reward signals are available (Levine et al., 2020). Our experiments demonstrate that our approach not only significantly outperforms other offline RL methods such as DPO (Rafailov et al., 2023), RFT/RAFT (Dong et al., 2023; Ahn et al., 2024), but also our method performs favorably against online RL methods such as GRPO and PPO for Qwen2.5-Math-1.5B model in math tasks. Note that our method is much simpler in computation resources or procedure than most of the RL methods that may require a reference model or certain batch size, offering a practical alternative.

To understand how DFT affects the model differently, we analyze the change in the probability distribution after training (Figure 2. We observe that traditional SFT training uniformly increases token probabilities toward fitting the training data more tightly, whereas our approach, interestingly, also pushes some token distribution away from the training set. In particular, although most tokens still fit the training set closer, the percentage of less strongly fitted tokens increases markedly. We provide an in-depth discussion of this phenomenon in Section 4.3.1.

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 SFT's limited generalization, and derive a method to improve it. On the experimental side, we show that such a simple solution, just one line of code, can substantially enhance LLM SFT's performance and generalization capabilities across various tasks and models.

2 RELATED WORK

The trade-off between SFT and RL is a central theme in modern language model alignment. SFT is widely adopted for its simplicity and efficiency in mimicking expert demonstrations (Chung et al., 2024; Zhou et al., 2023; Wei et al., 2022), a process analogous to behavioral cloning in robotics (Sammut, 2011; Mandlekar et al., 2022). However, the literature frequently notes that this approach can lead to overfitting and poor generalization compared to RL, which leverages reward signals to explore and discover more robust policies (Ouyang et al., 2022; Christiano et al., 2017; Bai et al., 2022; Swamy et al., 2025). Recently, Chu et al. conduct a systematic comparison of SFT and RL on both textual and visual tasks, confirming that "SFT memorizes while RL generalizes". More importantly, they also find that SFT remains necessary as an initialization step to stabilize output formatting before RL training can be effective. Nevertheless, RL comes with significant practical hurdles, including high computational costs, hyperparameter sensitivity, and the need for an explicit

reward function, often limit its applicability (Schulman et al., 2017; Strubell et al., 2019; Sheng et al., 2025).

To exploit the benefits of both paradigms, a dominant line of research has focused on hybrid methods. The most established strategy involves an SFT pre-training phase followed by RL-based refinement, often using a learned reward model, as popularized by InstructGPT (Ouyang et al., 2022). More recent approaches have explored alternative combinations, such as interleaving SFT and RL steps to improve stability and performance (Sheng et al., 2025; Liu et al., 2025; Qiu et al., 2025). Other prominent approaches, such as Direct Preference Optimization (DPO) (Rafailov et al., 2023), bypass explicit reward modeling by directly optimizing a policy on preference data, effectively integrating imitation and reinforcement signals into a single loss function. Chen et al. (2025a) Negative-aware Fine-Tuning (NFT) while enables LLMs to self-improve by modeling their own incorrect generations through an implicit negative policy. Although powerful, these methods are designed for settings where a reward signal, preference pairs, or negative samples are available. They augment the training pipeline, but do not fundamentally improve the SFT process in its native context, where there are only positive expert demonstrations. Our work diverges by focusing on enhancing SFT itself, without requiring any external feedback.

On the other side, a theoretical line of inquiry has sought to unify SFT and RL. For example, Du et al. (2025) reframe RLHF as a reward-weighted form of SFT, simplifying the pipeline while maintaining dependence on an explicit reward. Wang et al. (2025) demonstrate that SFT can be viewed as an RL method with an implicit reward, proposing solutions such as smaller learning rates to manage an otherwise vanishing KL constraint. Abdolmaleki et al. (2025) analyze learning from both positive and negative feedback, showing how their balance affects policy convergence. Qin & Springenberg (2025) reframe SFT as a lower bound of RL and improve it by introducing importance weighting based on the data-generating policy. Although these works point out the general connection between SFT and RL through the lens of weighting, they stop short of providing a precise mathematical equivalence between the SFT gradient and the off-line policy gradient. In contrast, our work is the first to rigorously establish this equivalence, explicitly identifying that the key difference lies in an inverse-probability weighting term present in SFT. This insight directly motivates our proposed solution: simply multiplying the SFT loss with the model probability to cancel out the weighting.

Interestingly, our method yields a design for the cross-entropy (CE) loss that is diametrically opposite to the well-known Focal Loss (Lin et al., 2017). Our modified CE is $-p\log(p)$, while the focal loss is $-(1-p)^\gamma \log(p)$. Focal Loss intentionally downweights well-classified samples to improve performance on underrepresented classes, whereas we intentionally downweight poorly-classified samples to improve generalization. This contrast may reflect a fundamental shift in the era of LLMs, where underfitting becomes less problematic than overfitting.

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 full reference response for 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–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.

(margin note: SFT as on-policy expectation)

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

Define the auxiliary variables

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

Reorganize the Equation 5 and rewrite it using the above auxiliary variables, we obtain the form

(margin note: Similar to eq. (4) now)

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

This form of SFT gradient now closely aligns with policy gradient Equation 4. Thus we can see, *conventional SFT is precisely an on-policy-gradient with the reward as an indicator function of matching the expert trajectory but biased by an importance weighting $1/\pi_\theta$.*

Given the unavoidable sparsity of reward signals in the SFT setting, we identify the importance sampling weight $1/\pi_\theta$ as a fundamental cause of SFT's poor generalization relative to RL. When the model assigns low probability to the expert response, the weight w becomes large, resulting in unbounded and high-variance reward estimates from an RL perspective. This large variance issue is exacerbated by the extreme sparsity of the reward function—since $r(x,y) = \mathbf{1}[y=y^\star]$ is nonzero only when the model exactly matches the expert output. As a result, optimization tends to overfit to rare exact-match demonstrations, undermining 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 an 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

$$\nabla_\theta \mathcal{L}_{\text{DFT}}(\theta) = \nabla_\theta \mathcal{L}_{\text{SFT}}(\theta)\ \cdot\ \text{sg}(\frac{1}{w}) = \nabla_\theta \mathcal{L}_{\text{SFT}}(\theta)\ \cdot\ \text{sg}(\pi_\theta(y^\star \mid x)). \quad (7)$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, ensuring that gradients do not flow through the reward scaling term w. To facilitate transitioning 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 $y \neq y^\star$ as 0. Now since the gradient does not flow, the corrected SFT loss also becomes a simple reweighted loss, called Dynamic Fine-tuning (DFT).

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

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

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

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.

4 EXPERIMENTS

4.1 MAIN EXPERIMENT - SFT SETTING

We focs on the standard SFT setting, characterized by having only expert demonstration data without negative samples, reward models, or verification signals. Our expert dataset typically originates from external policies, such as expert models or human annotations. The primary objective of this experiment is to rigorously evaluate whether DFT can robustly surpass standard SFT across diverse tasks, model architectures, model sizes, and dataset sizes.

4.1.1 SETUP AND IMPLEMENTATION DETAILS

Dataset and Models. We train with the NuminaMath CoT dataset (LI et al., 2024), comprising around 860,000 mathematical problems paired with the corresponding solutions. The dataset spans various sources, including Chinese high school mathematics exercises and U.S. and international mathematical olympiads. To efficiently manage computational resources, we randomly sample 100,000 instances from the dataset for training, which is sufficient since the evaluation accruacy curve 1 indicates the convergence of all methods well before the dataset exhaustion. We conduct experiments using multiple state-of-the-art models, including Qwen2.5-Math-1.5B, Qwen2.5-Math-7B (Qwen Team et al., 2024a), LLaMA-3.2-3B, LLaMA-3.1-8B (Dubey et al., 2024), and DeepSeekMath-7B-Base (Shao et al., 2024).

Training Details. Our implementation builds upon the verl framework (Sheng et al., 2025), using recommended SFT hyperparameters. Specifically, we employ the AdamW optimizer with learning rates of 5×10^{-5} for all models except the LLaMA-3.1-8B, for which we adopt a lower learning rate of 2×10^{-5}. We set the mini-batch size to 256 and the maximum input length to 2048 tokens. The learning rate follows a cosine decay schedule with a warm-up ratio of 0.1. We also include a concurrent method, Importance-Weighted SFT (iw-SFT) (Qin & Springenberg, 2025), for comparison. All training settings follow those reported in the original paper, except that we set the number of training epochs to 1.

Evaluation Settings. For mathematical reasoning tasks, we evaluate on established benchmarks including Math500 (Hendrycks et al.), Minerva Math (Lewkowycz et al., 2022), Olympiad Bench (AI Mathematical Olympiad, 2024), AIME 2024(American Institute of Mathematics, 2024), and AMC 2023(Mathematical Association of America, 2023). Each model uses the default chat template and Chain-of-Thought (CoT) prompting to stimulate step-by-step reasoning. All reported results represent average accuracy across 16 decoding runs, evaluated with a temperature of 1.0 and maximum generation length of 4096 tokens.

4.1.2 MAIN RESULTS

DFT consistently yields significantly average performance improvements over base models compared to standard SFT across all evaluated LLMs. As shown in Table 1, for example, 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-

Table 1: Average@16 accuracy of five state-of-the-art large language models on five mathematical reasoning benchmarks: Math500, Minerva Math, Olympiad Bench, AIME 2024, and AMC 2023. 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**

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 greater resilience on difficult reasoning tasks where traditional SFT struggles. This highlights its potential as a more robust fine-tuning paradigm to enhance mathematical reasoning capabilities in LLMs.

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 indicates 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.

DFT outperforms the concurrent Importance-Weighted SFT (iw-SFT) in most settings across model families and benchmarks. As shown in Table 2, DFT achieves higher average accuracy than iw-SFT on most model families: LLaMA-3.2-3B (+2.39), LLaMA-3.1-8B (+4.15), DeepSeekMath-7B (+3.34), and Qwen2.5-Math-1.5B (+1.30). Although iw-SFT slightly outperforms our method on Qwen2.5-Math-7B (+2.45), this improvement is not consistent across datasets. In particular, on the LLaMA model family, iw-SFT exhibits signs of limited robustness. For LLaMA-3.2-3B, iw-SFT underperforms standard SFT on Math500 (5.13 vs. 8.65) and AMC23 (2.03 vs. 3.13). Similarly, for LLaMA-3.1-8B, iw-SFT results in worse performance than SFT on Minerva Math (4.31 vs. 5.78) and AMC23 (7.34 vs. 8.28). These cases demonstrate that iw-SFT might struggle to generalize beyond specific training signals, and may even degrade performance under distribution shifts or on harder benchmarks. In contrast, DFT consistently improves upon both the base model and SFT across nearly all datasets, including those where iw-SFT fails. These results underline better generalization ability of DFT in diverse mathematical reasoning scenarios. Moreover, iw-
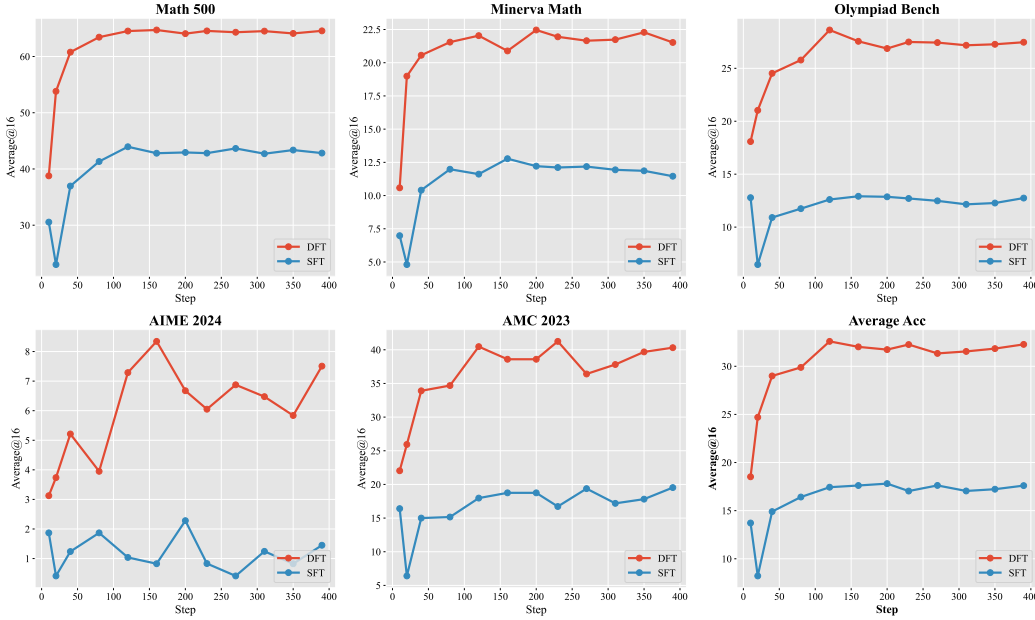
Figure 1: Accuracy progression for Qwen2.5-MATH-1.5B across mathematical benchmarks, illustrating faster convergence and better performance achieved by DFT relative to SFT.

Table 2: Comparison with concurrent work iw-SFT on math benchmarks. DFT outperforms the iw-SFT in most settings across model families and benchmarks. The best performance is bold.

	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
LLaMA-3.2-3B w/iw-SFT	5.13	2.63	1.51	0.00	2.03	2.26
LLaMA-3.2-3B w/DFT	**12.79**	**2.84**	**2.90**	**0.83**	**3.91**	**4.65**
LLaMA-3.1-8B w/iw-SFT	18.21	4.31	4.31	0.20	7.34	6.87
LLaMA-3.1-8B w/DFT	**27.44**	**8.26**	**6.94**	**0.41**	**12.03**	**11.02**
DeepSeekMath-7B w/iw-SFT	35.32	8.75	11.11	0.61	**18.28**	14.81
DeepSeekMath-7B w/DFT	**41.46**	**16.79**	**15.00**	**1.24**	16.25	**18.15**
Qwen2.5-Math-1.5B w/iw-SFT	59.38	17.08	26.82	**8.13**	**40.00**	30.28
Qwen2.5-Math-1.5B w/DFT	**64.89**	**20.94**	**27.08**	6.87	38.13	**31.58**
Qwen2.5-Math-7B w/iw-SFT	**70.28**	25.70	**34.46**	**16.46**	**51.09**	**39.60**
Qwen2.5-Math-7B w/DFT	68.20	**30.16**	33.83	8.56	45.00	37.15

SFT incurs additional computational overhead by requiring a separate reference model to compute importance weights, whereas DFTdynamically derives its own weighting directly from the model's token probabilities, resulting in a more efficient training procedure.

4.2 EXPLORATORY EXPERIMENT - OFFLINE RL SETTING

4.2.1 SETUP AND IMPLEMENTATION DETAILS

Data Preparation. We conduct an exploratory investigation of applying DFT an offline RL setting, where the sparsity of reward issue could be alleviated comparing to SFT setting. Specifically, we adopt the commonly used rejection sampling fine-tuning (RFT) framework Dong et al. (2023); Ahn et al. (2024). Following the setup in Section 4.1, we sample responses for 10,000 math ques-

Table 3: Evaluation results on five mathematical reasoning benchmarks in an offline reinforcement learning setting using reward signals from rejection sampling. DFT achieves the best overall performance, surpassing both offline (RFT, DPO) and online (PPO, GRPO) baselines, demonstrating its efficiency and strength as a simple yet effective fine-tuning strategy.

	Setting	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
Qwen2.5-Math-1.5B w/SFT	SFT	43.14	11.64	13.41	1.03	14.84	16.81
Qwen2.5-Math-1.5B w/iw-SFT	SFT	59.38	17.08	26.82	8.13	40.00	30.28
Qwen2.5-Math-1.5B w/DFT	SFT	62.50	22.94	26.87	7.31	33.75	30.67
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/iw-SFT	Offline	60.80	18.13	27.83	8.33	44.21	31.86
Qwen2.5-Math-1.5B w/DFT	Offline	**64.71**	**25.16**	**30.93**	**7.93**	**48.44**	**35.43**

tions using a temperature of 1.0 and generate four responses per question from the base model itself. Correct responses are identified using math verify and retained as training data, resulting in approximately 140,000 examples. For DPO training, we construct 100,000 positive–negative preference pairs from the generated responses.

Training Details. All experiments are conducted using the Qwen2.5-math-1.5B model. We compare DFT with representative offline RL methods, including DPO (Rafailov et al., 2023) and RFT (Dong et al., 2023; Ahn et al., 2024), as well as online RL methods PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). For RFT and DFT, the training setup follows the configuration in Section 4.1. For DPO, we use the ms-swift framework (Zhao et al., 2024) with a learning rate of 1×10^{-6}, batch size of 128, and a warmup ratio of 0.05. For PPO and GRPO, training is performed using the verl framework (Sheng et al., 2025) with a learning rate of 1×10^{-6}, batch size of 256, and a warmup ratio of 0.1. We set the number of response per input to $n = 4$ for GRPO.

4.2.2 RESULTS

DFT demonstrates best performance in the offline reinforcement learning setting, outperforming both offline and online RL baselines. As shown in Table 3, DFT achieves an average score of 35.43, exceeding the best offline method RFT by +11.46 points (from 23.97 to 35.43), and even outperforming the strongest online RL algorithm GRPO by +3.43 points (from 32.00 to 35.43). Our model performs well across all five benchmarks. 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 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 (18.93) by +6.23 points, PPO (15.41) by +9.75, and all offline baselines by a wider gap.

We also compare against the concurrent iw-SFT (Qin & Springenberg, 2025) method under the offline setting. While iw-SFT performs competitively on certain datasets, achieving 60.80 on Math500 and 44.21 on AMC23, its overall average performance (31.86) still falls short of our method by a +3.57 point margin. Moreover, iw-SFT yields only marginal improvements compared to its own performance under the standard SFT setting, achieving an average score of 31.86 in the offline RL setting versus 30.28 with SFT. This modest gain of +1.58 points stands in stark contrast to the improvement achieved by DFT (+4.76, from 30.67 to 35.43). These results suggest that iw-SFT struggles to effectively leverage reward supervision under offline constraints, whereas DFT is able to consistently convert such signals into more robust generalization and higher task performance.

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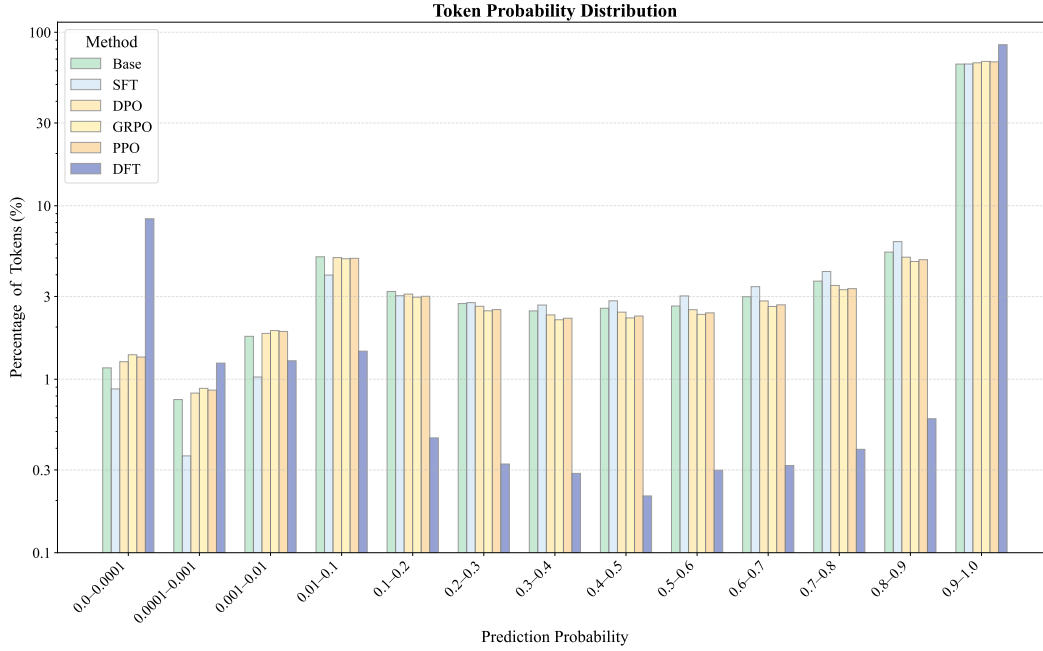
Figure 2: Token probability distributions on the training set before training and after fine-tuning with DFT, SFT, and various RL methods including DPO, PPO, and GRPO. A logarithmic scale is used on the y-axis to improve visualization clarity.

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 ABLATION AND INVESTIGATION

4.3.1 TOKEN PROBABILITY DISTRIBUTION

To understand how the model trained by DFT is different from standard SFT and other RL methods, we look into the token probability distribution of the model's output over the training set in Figure 2. The results reveal how the methods alter the probability landscape. 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 conjuncative 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. For large language models, it may be beneficial to deprioritize fitting tokens that serve grammatical functions rather than carrying primary semantic content. This concept is analogous to human pedagogy, where students are taught to focus on substantive concepts rather than perfecting the usage of common connective words.

4.3.2 TRAINING HYPER-PARAMETERS ABLATION

To assess the robustness and sensitivity of our approach (DFT) with respect to key training hyperparameters, we conduct an ablation study focused on learning rate and batch size, using the Qwen2.5-Math-1.5B base model. This analysis aims to answer two central questions: (1) Is the
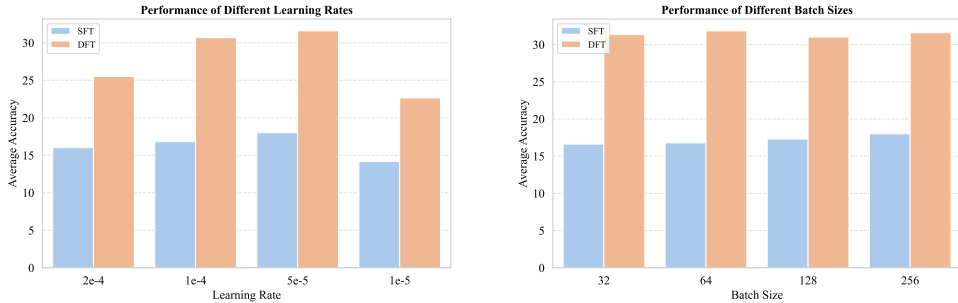
Figure 3: Ablation study of training hyper-parameters, learning rates and batch size, for DFT and SFT on Qwen2.5-Math-1.5B model.

performance gap between DFT and SFT due to a suboptimal hyperparameter configuration in SFT? (2) How sensitive are both methods to changes in learning rate and batch size?

We evaluate both DFT and SFT across four learning rates: 2e-4, 1e-4, 5e-5, and 1e-5. As shown in Figure 3 (left), both methods exhibit a certain degree of sensitivity to the learning rate. DFT consistently outperforms SFT under all configurations, suggesting that the performance gap cannot be attributed solely to suboptimal hyperparameter choices in SFT. For both methods, intermediate learning rates (1e-4 and 5e-5) yield the best results, while both lower (1e-5) and higher (2e-4) values lead to noticeable degradation. These findings highlight the importance of properly tuning the learning rate in gradient-based fine-tuning.

We further assess the impact of batch size, sweeping values from 32 to 256. As shown in Figure 3 (right), both DFT and SFT exhibit relatively stable performance across the full range of batch sizes. While minor fluctuations are observed, there is no consistent trend indicating that larger or smaller batches significantly affect final accuracy. This suggests that batch size is not a dominant factor for either method in this setup. This suggests that batch size is not a dominant factor for either method in this setup, and default values may suffice in practice.

5 CONCLUSION

In this work, we address the well-documented generalization gap between Supervised Fine-Tuning and Reinforcement Learning. We provide a novel theoretical analysis, demonstrating that the standard SFT gradient is equivalent to a policy gradient update with an ill-posed implicit reward that is inversely proportional to the model's confidence. This insight explains SFT's tendency to overfit and its unstable optimization dynamics. Building on this analysis, we introduce DFT, a simple yet powerful method that rectifies this issue by dynamically reweighting the SFT loss with the token probability. This one-line modification stabilizes the learning process and promotes better generalization. Our extensive experiments show that DFT consistently and substantially outperforms standard SFT across various models and challenging mathematical reasoning benchmarks. Furthermore, when adapted to an offline RL setting, DFT surprisingly surpasses established online and offline RL algorithms, highlighting its effectiveness and efficiency. Our work provides both a deeper understanding of SFT and a practical, high-impact solution that significantly closes the performance gap with more complex RL methods.

Limitations. While our experiments demonstrate substantial gains from DFT on mathematical reasoning benchmarks, this evaluation is confined to math-focused datasets and models up to 7 billion parameters. We have not yet assessed performance on other task domains (e.g., code generation, commonsense QA) or with larger LLMs (e.g., 13 B+). Moreover, our current study is limited to text-only scenarios. In future work, we plan not only to extend our study to a broader range of text benchmarks and to scale DFT to state-of-the-art models, but also to validate its effectiveness on vision-language tasks to confirm its generality across modalities.

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 *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024. 2, 7, 8

AI Mathematical Olympiad. Ai mathematical olympiad prize datasets, 2024. URL https://www.kaggle.com/competitions/ai-mathematical-olympiad-prize. 5

American Institute of Mathematics. Aime 2024 competition mathematical problems, 2024. URL https://www.maa.org/math-competitions/aime. 2, 5

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

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. 3

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. URL https://arxiv.org/abs/2503.04548. 1

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 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 *Forty-second International Conference on Machine Learning*. 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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng

Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948. 4

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, 7, 8

Yilun Du et al. Simplify rlhf as reward-weighted sft: A variational method. *arXiv preprint arXiv:2502.11026*, 2025. URL https://arxiv.org/abs/2502.11026. 3

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. 5

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 5

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

International Mathematical Olympiad. Mathematical olympiad problems 2024, 2024. URL https://www.imo-official.org. 2

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. *Advances in neural information processing systems*, 35:3843–3857, 2022. 5

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. https://huggingface.co/collections/AI-MO/numina-math-models-and-datasets-66f94e8de52a7bfd5af7e28e, 2024. 2, 5

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. 3

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. *arXiv preprint arXiv:2505.16984*, 2025. 1, 3

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 *Conference on Robot Learning*, pp. 1678–1690. PMLR, 2022. 1, 2

Mathematical Association of America. Amc 2023 competition problems, 2023. URL https://www.maa.org/math-competitions. 2, 5

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. 1, 2, 3

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, 5, 8

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. URL https://www.arxiv.org/pdf/2506.13056. 1, 3

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. 5

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 *Advances in Neural Information Processing Systems*, volume 36, 2023. 1, 2, 3, 8

Claude Sammut. *Behavioral Cloning*. Springer, 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 *International Conference on Learning Representations*, 2022. 1

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 3, 4, 8

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. 5, 8

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. 1, 3, 5, 8

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019. 1, 3

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

Yifan Wang et al. Implicit reward as the bridge: A unified view of sft and dpo connections. *arXiv preprint arXiv:2507.00018*, 2025. URL https://arxiv.org/abs/2507.00018. 3

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. 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

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*, 2024. 1

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. URL https://arxiv.org/abs/2408.05517. 8

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 *Advances in Neural Information Processing Systems*, volume 36, 2023. 1, 2