MIND THE GAP: DATA REWRITING FOR STABLE OFF-POLICY SUPERVISED FINE-TUNING

Shiwan Zhao, Xuyang Zhao, Jiaming Zhou, Aobo Kong, Qicheng Li, Yong Qin[*]

TMCC, College of Computer Science, Nankai University, Tianjin, China
Email: zhaosw@gmail.com

ABSTRACT

Supervised fine-tuning (SFT) of large language models can be viewed as an off-policy learning problem, where expert demonstrations come from a fixed behavior policy while training aims to optimize a target policy. Importance sampling is the standard tool for correcting this distribution mismatch, but large policy gaps lead to high variance and training instability. Existing approaches mitigate this issue using KL penalties or clipping, which passively constrain updates rather than actively reducing the gap. We propose a simple yet effective data rewriting framework that proactively shrinks the policy gap by keeping correct solutions as on-policy data and rewriting incorrect ones with guided re-solving, falling back to expert demonstrations only when needed. This aligns the training distribution with the target policy before optimization, reducing importance sampling variance and stabilizing off-policy fine-tuning. Experiments on five mathematical reasoning benchmarks demonstrate consistent and significant gains over both vanilla SFT and the state-of-the-art Dynamic Fine-Tuning (DFT) approach. Data and code are available at `https://github.com/NKU-HLT/Off-Policy-SFT`.

Index Terms— Supervised fine-tuning, off-policy learning, importance sampling, data rewriting

1. INTRODUCTION

Large language models (LLMs) have achieved remarkable progress in Chain-of-Thought (CoT) reasoning [1], largely driven by a post-training pipeline that combines supervised fine-tuning (SFT) with reinforcement learning (RL) [2, 3, 4, 5]. SFT distills task-specific reasoning behaviors from high-quality demonstrations, enabling base models to rapidly adapt to novel tasks. RL complements this by optimizing on-policy rollouts with reward-based objectives, delivering consistent gains on challenging reasoning benchmarks. In the widely adopted *SFT-then-RL* paradigm, SFT provides a solid initialization for reasoning that RL subsequently refines through on-policy sampling.

Despite their close connection, SFT and RL exhibit complementary strengths and limitations [6, 7]. SFT is simple and efficient, capable of expanding the model's reasoning capability frontier by incorporating external expert knowledge and reasoning patterns. However, it operates entirely on off-policy data because expert demonstrations come from a fixed behavior policy rather than the evolving model policy, leading to the well-known policy gap and causing high variance, training instability, and overfitting. RL, in contrast, performs on-policy optimization and thus avoids the policy gap altogether, but it suffers from high sample and computational complexity and can only refine the model's existing reasoning behaviors without introducing fundamentally new capabilities. In this work, we focus on improving SFT itself, providing a more stable foundation for standalone fine-tuning as well as for future extensions involving RL or hybrid approaches.

From the perspective of off-policy learning [8], importance sampling (IS) is the standard tool for correcting the distribution mismatch between the behavior and target policies. Yet, when the policy gap becomes large, IS weights become highly skewed, leading to variance amplification and unstable optimization. Existing remedies, such as KL penalties, trust regions, or clipped ratios [9, 10], stabilize optimization by passively constraining updates but fail to actively reduce the underlying gap in the data distribution itself.

We propose a simple yet effective data rewriting framework that proactively reduces the policy gap before optimization begins. For each problem, we first sample multiple responses from the target model. If any response solves the problem correctly, we retain it as on-policy data. Otherwise, we prompt the model with the ground-truth solution as a reference to re-solve the problem, generating *digest-and-retell* data that better reflects the target policy. If both self-solve and re-solve fail, we fall back to the original expert demonstration. Inspired by the intuition that true understanding emerges when learners re-express solutions in their own words rather than copying them verbatim, our *digest-and-retell* strategy (see Figure 2) transforms the SFT dataset rather than merely constraining optimization dynamics. This process aligns the training distribution more closely with the target policy, reducing importance sampling variance and stabilizing off-policy fine-tuning, with residual mismatch further mitigated through IS weighting.

Experiments on five mathematical reasoning benchmarks show that our approach consistently outperforms both vanilla SFT and the state-of-the-art Dynamic Fine-Tuning (DFT) approach [11]. In particular, on the Qwen2.5-Math-7B model, our method improves the average accuracy from 23.23% to 30.33% over SFT and from 36.61% to 42.03% over DFT.

Our contributions are three-fold:

- We formulate SFT as an off-policy learning problem and identify the policy gap as the key source of instability in IS-based optimization.
- We introduce a data rewriting framework that proactively reduces the policy gap at the data level, enabling low-variance and stable off-policy fine-tuning.
- We validate the approach across multiple models and benchmarks, demonstrating consistent gains over standard SFT and DFT baselines.

[*] Corresponding author. This work was supported by NSF China (Grant No. 62271270) and the National Key R&D Program of China (Grant No. 2022ZD0116307).

2. RELATED WORK

2.1. Data-Centric Improvements for SFT

The quality of SFT largely depends on the construction of instruction datasets, and prior work explores three major aspects: scaling, diversity, and quality. Flan [12] scales the number of instruction-tuning tasks and demonstrates that larger task collections substantially boost performance. LIMA [13] shows that fine-tuning a strong pretrained model on only 1,000 carefully curated examples can achieve competitive results, highlighting the importance of data quality and diversity.

Beyond scaling, several studies transform training examples to better align with the target model distribution. GRAPE [14] selects responses with the highest target-model likelihood from multiple LLMs before SFT training. Self-Distillation Fine-Tuning (SDFT) [15] generates distilled data using the model itself to bridge the distribution gap. Self-to-Supervised Fine-Tuning (S3FT) [16] identifies correct model responses and fine-tunes on them while paraphrasing or preserving gold answers for the remaining samples. Our data rewriting framework follows this line but adopts an off-policy perspective: it actively reduces the policy gap at the data level through rewriting and further mitigates residual mismatch via importance sampling during training.

2.2. Combining SFT and RL

Another line of work combines the capability expansion of SFT with the on-policy robustness of RL. Interleaved or unified training objectives jointly optimize supervised and reinforcement signals [6, 17, 18], while dynamic weighting strategies balance SFT imitation learning with RL-based preference optimization. Although RL operates on-policy, the SFT stage in these methods still relies on static off-policy datasets, leaving the underlying distribution mismatch unresolved. Our method is orthogonal: rather than modifying training objectives or interleaving on- and off-policy rollouts, it directly reduces the policy gap before optimization at the data level via data rewriting, offering a complementary perspective to existing objective-level approaches.

2.3. Off-Policy Learning

A parallel line of work views SFT as an off-policy learning problem, where expert demonstrations and the evolving target policy induce a distribution mismatch. One line of research introduces importance sampling or reward rectification to correct this mismatch [11, 19]. Another line focuses on reducing the variance of importance sampling via optimization-level techniques such as clipping or trust regions [20], which stabilize training by constraining or reweighting updates. In contrast, our data rewriting framework actively aligns the training distribution with the target policy before optimization, providing a complementary perspective to these optimization-level methods.

3. METHOD

We view supervised fine-tuning (SFT) as an off-policy learning problem and propose a unified framework (Figure 1) that combines *data rewriting*, which proactively reduces the policy gap at the data level, with *importance sampling* (IS), which further corrects residual mismatch during optimization.

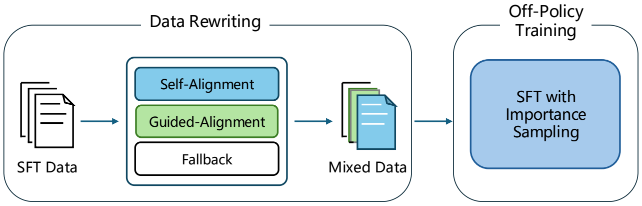
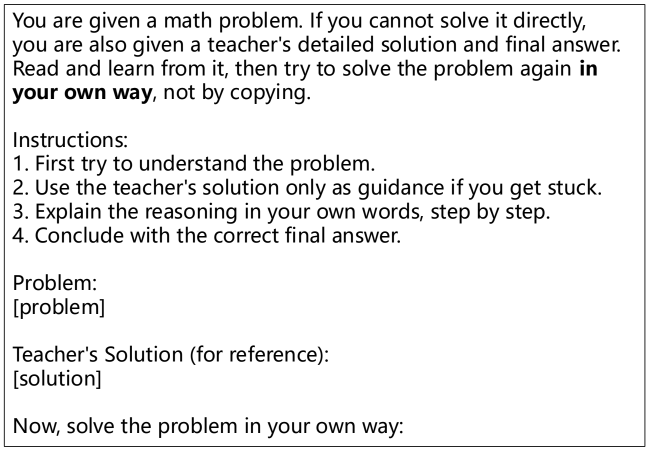


Fig. 1. The overall framework consists of (i) data rewriting, which converts SFT data from off-policy to a more on-policy distribution, and (ii) off-policy training with importance sampling, which further mitigates the remaining policy gap.



You are given a math problem. If you cannot solve it directly, you are also given a teacher's detailed solution and final answer. Read and learn from it, then try to solve the problem again **in your own way**, not by copying.

Instructions:
1. First try to understand the problem.
2. Use the teacher's solution only as guidance if you get stuck.
3. Explain the reasoning in your own words, step by step.
4. Conclude with the correct final answer.

Problem:
[problem]

Teacher's Solution (for reference):
[solution]

Now, solve the problem in your own way:

Fig. 2. The *digest-and-retell* prompt, which provides a reference solution and asks the model to re-solve the problem in its own words.

3.1. SFT as Off-Policy Learning

Let π_{sft} denote the behavior policy generating the SFT dataset, and let π_θ be the target policy parameterized by θ. The goal of SFT is to maximize the expected reward under π_θ:

$$J(\theta) = E_{(x,y)\sim\pi_\theta}[r(x,y)], \tag{1}$$

where $r(x, y)$ is a task-specific reward signal (e.g., correctness).

However, since training data come from π_{sft} rather than π_θ, SFT becomes an off-policy problem:

$$J(\theta) = E_{(x,y)\sim\pi_{sft}}\big[w(x,y)\, r(x,y)\big], \tag{2}$$

where the importance weight is defined as $w(x, y) = \frac{\pi_\theta(y|x)}{\pi_{sft}(y|x)}$. When the divergence $D(\pi_{sft} \,\|\, \pi_\theta)$ is large, these weights become highly skewed, leading to variance amplification and unstable optimization. Existing methods mitigate this issue using KL penalties, trust regions, or clipping, which passively constrain updates but do not actively reduce the underlying policy gap.

3.2. Data Rewriting as Policy Alignment

We introduce a data rewriting operator $\mathcal{T}$ that transforms π_{sft} into a mixture distribution π_{mix} before training:

$$\pi_{sft} \xrightarrow{\mathcal{T}} \pi_{\text{mix}} \quad \text{with} \quad D(\pi_{\text{mix}} \| \pi_\theta) < D(\pi_{sft} \| \pi_\theta). \tag{3}$$

The operator $\mathcal{T}$ applies a three-stage alignment hierarchy:

	Self-alignment	Guided-alignment	Fallback	Total
Qwen2.5-Math-7B	28,752	11,620	7,634	48,006
Llama-3.1-8B-Instruct	26,947	16,335	4,719	48,001

Table 1. Dataset statistics for different models across alignment stages.

- **Self-alignment:** For each input x, we sample multiple responses from π_θ. If any response solves the problem correctly, we randomly retain one correct response as on-policy data[1].
- **Guided-alignment:** For inputs where self-alignment fails, we prompt π_θ with reference solutions to generate *digest-and-retell* responses that paraphrase the expert answers rather than copying them verbatim (see Figure 2).
- **Fallback:** If guided-alignment also fails, we fall back to the original expert demonstration.

The resulting dataset $\mathcal{D}'$ consists of a mixture of on-policy and rewritten examples, with expert data included only as a fallback:

$$\mathcal{D}' = \mathcal{D}_{\text{self}} \cup \mathcal{D}_{\text{retell}} \cup \mathcal{D}_{\text{expert}}. \tag{4}$$

This hierarchical process ensures that $\pi_{\text{mix}} = \mathcal{T}(\pi_{sft})$ progressively shifts the training data toward the target policy, thereby reducing the policy gap before optimization begins.

3.3. Importance Sampling with Aligned Data

Although the resulting dataset $\mathcal{D}'$ aligns more closely with the target policy distribution, residual mismatch may still remain due to imperfect rewriting and fallback expert demonstrations. Even with perfect rewriting, batch updates can introduce a policy gap as the target policy is progressively updated[2]. To mitigate this issue, we apply importance sampling during optimization:

$$\mathcal{L}_{\text{IS}}(\theta) = E_{(x,y')\sim\mathcal{D}'}\left[-\sum_{t=1}^{|y'|} w(x, y'_t) \cdot \log \pi_\theta(y'_t \mid x, y'_{<t})\right], \tag{5}$$

where

$$w(x, y'_t) = \text{sg}\left(\frac{\pi_\theta(y'_t \mid x, y'_{<t})}{\pi_{\text{mix}}(y'_t \mid x, y'_{<t})}\right)$$

is the importance weight, and $\text{sg}(\cdot)$ denotes the stop-gradient operator to prevent gradients from flowing through the weight itself. Following common practice [11, 18], we approximate the denominator $\pi_{\text{mix}}(y'_t \mid x, y'_{<t}) \approx 1$, effectively treating the mixed data as the ground-truth distribution.

This two-level design, combining *data-level alignment* via rewriting with *optimization-level correction* via IS, enables low-variance, unbiased training and serves as a complementary approach to methods such as Dynamic Fine-Tuning (DFT) [11].

4. EXPERIMENTS

4.1. Dataset and Models

Following DFT [11], we use the NuminaMath CoT dataset [21] for training. The original dataset comprises approximately 860,000 mathematical problems along with their respective solutions. To reduce computational cost, we randomly sample 50,000 instances and retain around 48,000 after filtering out overlong examples.

	Math500	Minerva	Olympiad	AIME24	AMC23	Avg.
Qwen2.5-Math-7B	39.90	14.43	17.16	7.50	29.38	21.67
+ SFT	52.61	19.13	17.32	2.06	25.00	23.23
+ DFT	68.70	31.92	32.31	6.68	43.44	36.61
+ DR + SFT (ours)	59.85	21.14	23.54	8.54	38.59	30.33
+ DR + DFT (ours)	**70.40**	**34.85**	**36.12**	**14.58**	**54.22**	**42.03**
Llama-3.1-8B-Instruct	36.18	16.01	9.52	0.83	14.53	15.41
+ SFT	28.71	11.23	6.26	0.41	10.31	11.39
+ DFT	46.5	24.11	15.65	3.95	22.50	22.54
+ DR + SFT (ours)	44.38	19.21	13.07	1.87	17.19	19.14
+ DR + DFT (ours)	**47.91**	**24.72**	**16.52**	**4.99**	**26.09**	**24.05**

Table 2. Average accuracy (%) on mathematical reasoning benchmarks. Our method, combined with either SFT or DFT, consistently improves performance over the corresponding baselines. DR stands for data rewriting.

Model	Self-alignment		Guided-alignment		Fallback
	SFT	Rewriting	SFT	Rewriting	SFT
Qwen2.5-Math-7B	-184.44	-83.36	-280.64	-259.44	-296.03
Llama-3.1-8B-Instruct	-193.01	-127.67	-341.57	-303.44	-388.17

Table 3. Average log-probabilities of model responses across self-alignment, guided-alignment, and fallback subsets. Higher values (less negative) indicate responses closer to the target policy distribution.

Since our method requires models to generate candidate solutions for data rewriting, we experiment with two representative backbones: **Qwen2.5-Math-7B** [22], a math-specialized model without explicit instruction tuning, and **Llama-3.1-8B-Instruct** [23], a general instruction-tuned model. This comparison allows us to investigate whether our method can benefit both base and instruction-tuned models.

For data rewriting, we sample 10 candidate responses from the target model in both the self-alignment and guided-alignment stages. Dataset statistics for both models are provided in Table 1. We observe that Qwen2.5-Math-7B solves more problems in the self-alignment stage but fewer in the guided-alignment stage, likely due to its weaker instruction-following capabilities compared to Llama-3.1-8B-Instruct.

4.2. Training and Evaluation Details

For SFT training, we use the `verl` framework [24] with the AdamW optimizer. The learning rate is set to 5×10^{-5} for Qwen2.5-Math-7B and 7×10^{-6} for Llama-3.1-8B-Instruct. Training is performed for one epoch with a batch size of 256. We employ a cosine decay learning rate schedule with a 0.1 warm-up ratio.

For evaluation, we follow DFT and assess mathematical reasoning performance on five widely used benchmarks: Math500 [25], Minerva Math [26], OlympiadBench [27], AIME 2024 [28], and AMC 2023 [29]. All results are reported as the average accuracy over 16 decoding runs with a temperature of 1.0.

4.3. Main Results

Table 2 reports results on five mathematical reasoning benchmarks. Our data rewriting (DR) method consistently improves both vanilla SFT and DFT on Qwen2.5-Math-7B and Llama-3.1-8B-Instruct.

[1] When multiple correct responses are available, one is randomly selected for fair comparison with standard SFT; the same strategy applies in the guided-alignment stage.

[2] We leave online data rewriting for each batch to future work.

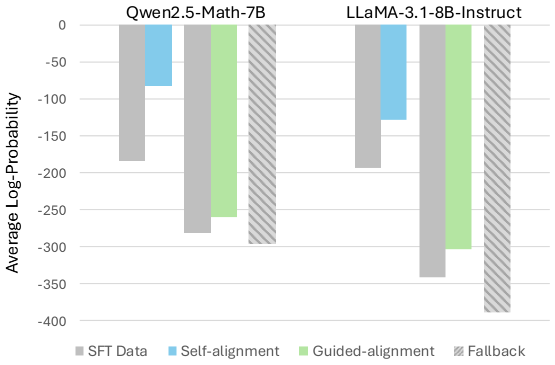


Fig. 3. Average log-probabilities across Self-alignment, Guided-alignment, and Fallback subsets for both models. Rewriting is applied only in Self- and Guided-alignment.

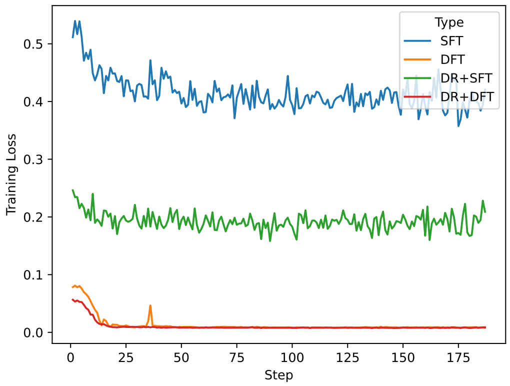


Fig. 4. Training loss curves of Qwen2.5-Math-7B for SFT, DFT, and their combinations with data rewriting (DR). DR+DFT achieves the lowest final loss and the most stable convergence.

Method	**Math500**	**Minerva**	**Olympiad**	**AIME24**	**AMC23**	**Avg.**
DFT	68.70	31.92	32.31	6.68	43.44	36.61
DFT + Self-Alignment	67.86	32.35	32.79	10.19	50.94	38.83
DFT + Full DR	**70.40**	**34.85**	**36.12**	**14.58**	**54.22**	**42.03**

Table 4. Ablation on Qwen2.5-Math-7B with DFT. *Self-Alignment* uses correct model responses, while *Full DR* combines both Self-Alignment and Guided-Alignment.

Qwen2.5-Math-7B. On this math-specialized model, vanilla SFT achieves an average accuracy of 23.23%, while DFT improves it to 36.61%. Incorporating DR yields substantial gains, with DR+SFT increasing performance to 30.33% and DR+DFT further boosting it to 42.03% average accuracy. Notably, DR+DFT attains the highest scores across all five benchmarks, with particularly large improvements on AIME2024 (6.68% → 14.58%) and AMC2023 (43.44% → 54.22%).

Llama-3.1-8B-Instruct. On this instruction-tuned model, DR improves SFT from 11.39% to 19.14% and DFT from 22.54% to 24.05%. While gains remain consistent, the magnitude is smaller than that on Qwen2.5-Math-7B, suggesting that instruction-tuned models benefit less from data rewriting.

To better understand the overall performance gains and the performance gap between Qwen2.5-Math-7B and LLaMA-3.1-8B-Instruct, we analyze the average log-probabilities across the self-alignment, guided-alignment, and fallback subsets (Table 3, Figure 3). The analysis shows that rewriting effectively closes the policy gap, as rewritten responses consistently achieve higher average log-probabilities (less negative), indicating closer alignment with the target policy. Moreover, the SFT data across the three subsets reveal increasing problem difficulty, as reflected by decreasing log-probabilities from self-alignment to guided-alignment to fallback; the *digest-and-retell* strategy still mitigates the policy gap on harder problems, although less effectively than self-solving. Finally, instruction tuning limits the benefits of data rewriting: LLaMA-3.1-8B-Instruct consistently exhibits lower log-probabilities than Qwen2.5-Math-7B across all subsets, suggesting that instruction tuning biases the model toward generic instruction-following and leaves less room for closing the policy gap.

Training Dynamics. Figure 4 presents the training loss curves for Qwen2.5-Math-7B across different methods. We find that DFT and DR+DFT converge much faster than SFT and DR+SFT, reaching near-zero training loss within the first 40–50 steps, which highlights the strong supervision signal and optimization stability provided by dynamic fine-tuning. Data rewriting also substantially lowers the loss of vanilla SFT, confirming that aligning the training distribution with the target policy before optimization helps mitigate variance and stabilize training. However, DR+SFT plateaus at a higher loss than DFT-based methods, suggesting that residual distribution mismatch persists without dynamic on-policy updates. DR+DFT combines the advantages of both approaches, achieving the lowest final loss and the most stable convergence, which explains its superior performance across all benchmarks in Table 2.

4.4. Ablation Study

The ablation results in Table 4 show that incorporating self-alignment into DFT yields moderate performance gains, increasing average accuracy from 36.61% to 38.83% by replacing the original SFT data with correct model-generated responses. The improvement is particularly pronounced on challenging benchmarks such as AIME24 and AMC23, suggesting that aligning the dataset more closely with the model's own policy effectively reduces variance and stabilizes training. Extending this approach to full data rewriting, which further revises the harder problems (those unsolved in the self-alignment stage) through guided alignment, delivers the best overall performance, boosting average accuracy to 42.03% and providing consistent gains across all benchmarks. These findings underscore the importance of proactively aligning training data before off-policy optimization to narrow the policy gap and enhance reasoning ability.

5. CONCLUSION

We introduced a simple yet effective data rewriting framework for supervised fine-tuning of large language models, formulating SFT as an off-policy learning problem. Our approach proactively reduces the policy gap at the data level before training and further alleviates residual mismatch through importance sampling during optimization. Extensive experiments on multiple mathematical reasoning benchmarks demonstrate consistent performance improvements over both vanilla SFT and state-of-the-art Dynamic Fine-Tuning, with the most significant gains observed on base models. These findings underscore the value of data-centric strategies for stabilizing and enhancing off-policy fine-tuning of large language models.

6. REFERENCES

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, and et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[2] Zhihong Shao, Peiyi Wang, Qihao Zhu, and et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[3] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, and et al., "Tulu 3: Pushing frontiers in open language model post-training," *arXiv preprint arXiv:2411.15124*, 2024.

[4] Daya Guo, Dejian Yang, Haowei Zhang, and et al., "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[5] Zihan Liu, Zhuolin Yang, Yang Chen, and et al., "Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy," *arXiv preprint arXiv:2506.13284*, 2025.

[6] Lu Ma, Hao Liang, Meiyi Qiang, and et al., "Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions," *arXiv preprint arXiv:2506.07527*, 2025.

[7] Jianhao Yan, Yafu Li, Zican Hu, and et al., "Learning to reason under off-policy guidance," *arXiv preprint arXiv:2504.14945*, 2025.

[8] Doina Precup, Richard S. Sutton, and Satinder P. Singh, "Eligibility traces for off-policy policy evaluation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, 2000, ICML '00, p. 759–766, Morgan Kaufmann Publishers Inc.

[9] John Schulman, Sergey Levine, Pieter Abbeel, and et al., "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, and et al., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[11] Yongliang Wu, Yizhou Zhou, Zhou Ziheng, and et al., "On the generalization of sft: A reinforcement learning perspective with reward rectification," *arXiv preprint arXiv:2508.05629*, 2025.

[12] Hyung Won Chung, Le Hou, Shayne Longpre, and et al., "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[13] Chunting Zhou, Pengfei Liu, Puxin Xu, and et al., "Lima: Less is more for alignment," *Advances in Neural Information Processing Systems*, vol. 36, pp. 55006–55021, 2023.

[14] Dylan Zhang, Qirun Dai, and Hao Peng, "The best instruction-tuning data are those that fit," *arXiv preprint arXiv:2502.04194*, 2025.

[15] Zhaorui Yang, Tianyu Pang, Haozhe Feng, and et al., "Self-distillation bridges distribution gap in language model fine-tuning," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, Aug. 2024, pp. 1028–1043, Association for Computational Linguistics.

[16] Sonam Gupta, Yatin Nandwani, Asaf Yehudai, and et al., "Selective self-to-supervised fine-tuning for generalization in large language models," in *Findings of the Association for Computational Linguistics: NAACL 2025*, Albuquerque, New Mexico, Apr. 2025, pp. 6240–6249, Association for Computational Linguistics.

[17] Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar, "Uft: Unifying supervised and reinforcement fine-tuning," *arXiv preprint arXiv:2505.16984*, 2025.

[18] Wenhao Zhang, Yuexiang Xie, Yuchang Sun, and et al., "On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting," *arXiv preprint arXiv:2508.11408*, 2025.

[19] 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.

[20] Wenhong Zhu, Ruobing Xie, Rui Wang, and et al., "Proximal supervised fine-tuning," *arXiv preprint arXiv:2508.17784*, 2025.

[21] Jia LI, Edward Beeching, Lewis Tunstall, and et al., "Numinamath," `[https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)`, 2024.

[22] An Yang, Beichen Zhang, Binyuan Hui, and et al., "Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement," *arXiv preprint arXiv:2409.12122*, 2024.

[23] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[24] Guangming Sheng, Chi Zhang, Zilingfeng Ye, and et al., "Hybridflow: A flexible and efficient rlhf framework," in *Proceedings of the Twentieth European Conference on Computer Systems*, 2025, pp. 1279–1297.

[25] Dan Hendrycks, Collin Burns, Saurav Kadavath, and et al., "Measuring mathematical problem solving with the math dataset," *arXiv preprint arXiv:2103.03874*, 2021.

[26] Aitor Lewkowycz, Anders Andreassen, David Dohan, and et al., "Solving quantitative reasoning problems with language models," *Advances in neural information processing systems*, vol. 35, pp. 3843–3857, 2022.

[27] XTX Investments, "Ai mathematical olympiad - progress prize 1," `https://kaggle.com/competitions/ai-mathematical-olympiad-prize`, 2024, Kaggle.

[28] American Institute of Mathematics, "AIME 2024 Competition Mathematical Problems," 2024.

[29] Mathematical Association of America, "AMC 2023 Competition Problems," 2023.