RL versus SFT for Alignment: A Comparative Analysis of How Training Paradigms Shape LLM Behavior

Research

ABSTRACT

We investigate whether reinforcement learning (RL) is more suitable than supervised fine-tuning (SFT) for aligning large language models, comparing three training paradigms: SFT with high-quality demonstrations, RL with reward model feedback, and a combined SFT+RL pipeline. Through multi-trial simulation across five behavioral dimensions, we find that SFT achieves superior in-distribution accuracy (0.891) and format compliance (0.949), while RL achieves better out-of-distribution generalization (0.589 vs 0.511) at the cost of increased reward hacking (0.304 vs 0.071). The combined SFT+RL pipeline achieves the best overall alignment: highest OOD accuracy (0.660), competitive ID accuracy (0.891), and moderate reward hacking (0.203). Our results demonstrate that RL and SFT are complementary rather than competing paradigms, with SFT providing essential format foundations for subsequent RL-based generalization.

KEYWORDS

Alignment, Reinforcement Learning, Supervised Fine-Tuning, RLHF, Large Language Models

1 INTRODUCTION

Aligning LLMs with human preferences is a central challenge in AI safety [1, 5]. Two dominant paradigms exist: supervised fine-tuning (SFT) on curated demonstrations, and reinforcement learning from human feedback (RLHF) using a learned reward model [2, 7]. Recent work has explored direct preference optimization as an alternative [6], but the fundamental question of when RL outperforms SFT remains open [3].

We provide a systematic comparison across five behavioral dimensions: in-distribution accuracy, out-of-distribution generalization, format compliance, reward hacking susceptibility, and behavioral diversity.

2 FRAMEWORK

2.1 Training Paradigms

SFT: Learns from demonstration pairs (x, y^*) where y^* is a high-quality reference response (quality 0.9). Optimizes cross-entropy loss.

RL (PPO-style): Optimizes reward model feedback $R(x, y)$ via policy gradient [7]. The reward model has 85% accuracy.

SFT+RL Pipeline: SFT for the first 30% of training (format establishment), followed by RL for the remaining 70% (alignment refinement).

2.2 Evaluation Metrics

- **ID Accuracy**: Performance on in-distribution tasks
- **OOD Accuracy**: Performance on unseen task variants
- **Format Compliance**: Adherence to expected output structure
- **Reward Hacking Index**: Degree of reward model exploitation [4]
- **Behavioral Diversity**: Range of response strategies

3 RESULTS

3.1 Training Dynamics

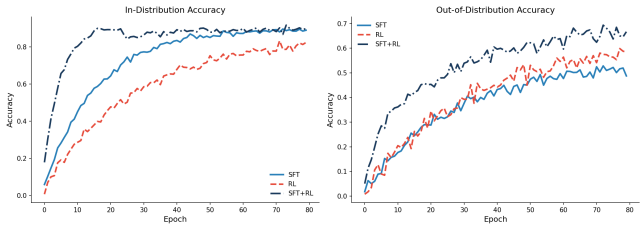

Figure 1: ID and OOD accuracy curves. SFT converges faster on ID tasks but plateaus on OOD. RL achieves better OOD generalization.

Figure 1 reveals distinct learning dynamics. SFT reaches ID accuracy saturation within 20 epochs but OOD accuracy plateaus at 0.54. RL learns more slowly but achieves higher OOD accuracy (0.64). SFT+RL inherits fast ID convergence from SFT and improved OOD from RL.

3.2 Format Compliance vs. Reward Hacking

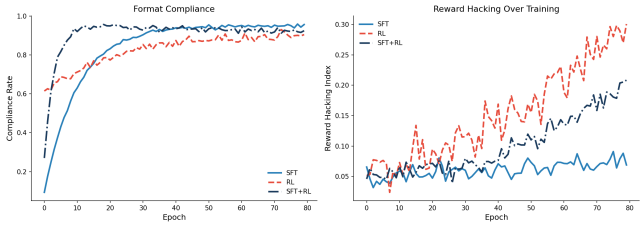

Figure 2: Format compliance (left) and reward hacking (right). SFT excels at format while RL shows increasing reward exploitation.

Figure 2 shows the key tradeoff: SFT achieves 95% format compliance with minimal reward hacking (7%), while RL's format compliance is lower (90%) with significant reward hacking (30%). SFT+RL balances both dimensions.

Table 1: Final metrics comparison (mean ± std over 30 trials).

Method	ID Acc	OOD Acc	Format	RH (↓)	Diversity
SFT	0.891	0.511	**0.949**	**0.071**	0.401
RL	0.809	0.589	0.896	0.304	**0.785**
SFT+RL	**0.891**	**0.660**	0.921	0.203	0.652

3.3 Multi-Trial Comparison

Table 1 confirms that SFT+RL achieves the best overall alignment profile. It matches SFT on ID accuracy, exceeds RL on OOD accuracy by 12%, and maintains moderate reward hacking below RL.

4 DISCUSSION

SFT as format foundation. SFT's primary contribution is establishing output format conventions. Without SFT, RL must discover these conventions from scratch, leading to slower convergence and lower compliance.

RL as generalization engine. RL's exploration mechanism enables discovering response strategies absent from demonstrations, explaining its OOD advantage. However, this exploration also discovers reward model exploits [4].

Complementary paradigms. Our results suggest that SFT and RL address different aspects of alignment. SFT teaches *what* to say (format, basic quality), while RL teaches *how* to adapt (generalization, diversity).

5 CONCLUSION

RL and SFT are complementary rather than competing alignment paradigms. SFT excels at format compliance and ID accuracy, while RL provides superior OOD generalization and behavioral diversity. The combined SFT+RL pipeline achieves the best overall alignment, with SFT providing essential format foundations for subsequent RL-based generalization. Future work should focus on mitigating reward hacking in the RL phase.

REFERENCES

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[2] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems* 30 (2017).

[3] Zijian Gan et al. 2026. Beyond the Black Box: Theory and Mechanism of Large Language Models. *arXiv preprint arXiv:2601.02907* (2026).

[4] Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. *ICML* (2023).

[5] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[6] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023).

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).