Offline and Online RLHF for Open-Ended Instruction Following

Rohan Shenoy*
UC Berkeley
rohan.shenoy@berkeley.edu

Cyprian Zander*
UC Berkeley
cyprian.zander@berkeley.edu

Abstract

In this paper, we describe our reinforcement-learning with human feedback (RLHF) fine-tuning process on a small open-weight large language model. Our base model of choice is Qwen2.5-1.5B-Instruct and we attempt to evaluate fine-tuning approaches in both offline and online problem. In particular, we first implement and evaluate DPO, IPO, and AOT in the offline optimization setting, and GRPO, DrGRPO, and GSPO in the online setting. We then aim to improve the instruction-following ability of the Qwen2.5 model in the online setting using an RLOO-style leave-one-out baseline for reward-model-based policy optimization.

1 Introduction

While scaling large language models has been shown to improve their ability to solve complex problems, many deployment settings are limited in compute and hardware and thus cannot support frontier-scale models. In edge or resource-constrained settings, smaller models are more practical, motivating the need for fine-tuning these smaller models after pretraining. Supervised fine-tuning is one common approach, but limited in the fact that it mainly teaches the model to imitate example responses. For open-ended instruction following, there is typically no single correct answer, making preference-based optimization a better fit. Rather than solely relying on demonstrations, RLHF uses comparisons between candidate responses to directly optimize the model towards preferred outputs. In this work, we study both offline and online RLHF methods for improving the Qwen2.5-1.5B-Instruct model. We first implement DPO [Rafailov et al., 2023], IPO [Azar et al., 2024], and AOT [Melnyk et al., 2024] approaches for the offline preference-optimization setting, where the methods must train directly from fixed chosen/rejected response pairs. For the online optimization setting, we first train a reward model from the preference data, then use its scores to optimize the language model on newly sampled responses. In this setting, we implement GRPO [Shao et al., 2024], DrGRPO [Liu et al., 2025], and GSPO [Zheng et al., 2025]. We then improve the policy using an RLOO-style leave-one-out baseline [Ahmadian et al., 2024], which stabilizes updates by comparing each sampled response against the other responses generated for that same prompt. The main question we attempt to answer is whether this modified advantage estimator produces a better head-to-head win rate against the frozen base model under a fixed evaluation protocol.

2 Problem Setup

Base model and data. All methods start from the frozen Qwen/Qwen2.5-1.5B-Instruct model. The dataset is a WildChat-derived preference set with four splits: paired preference splits (`train_prefs`, `test_prefs`) and prompt-only generation splits (`train_gen`, `test_gen`), with 4744 / 256 / 4744 / 256 samples respectively. The `train_prefs` and `train_gen` prompt distribu-

*Equal contribution.

tions match. `train_prefs` additionally carries chosen/rejected labels. We use `train_prefs` for offline preference optimization and reward-model training, and `train_gen` for the online rollouts.

Evaluation protocol. The primary metric is head-to-head win rate against the frozen base model. For each of 128 held-out prompts, we generate a response from the trained policy and a response from the frozen base model, then use GPT-5.4 as an external LLM judge to determine the better response. We report the fraction of prompts on which the trained policy wins.

3 Methods

3.1 Offline Preference Optimization

Let x be the prompt, y^+ the chosen response, y^- the rejected response, π_θ the trainable policy, and π_{ref} the frozen reference policy. Then we define the corrected preference margin to be:

$$\Delta_\theta(x, y^+, y^-) = \left[\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x)\right] - \left[\log \pi_{\text{ref}}(y^+ \mid x) - \log \pi_{\text{ref}}(y^- \mid x)\right]. \quad (1)$$

DPO. DPO optimizes this margin with the logistic loss, increasing the log likelihood of chosen response with respect to rejected ones:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\left(\beta \Delta_\theta(x, y^+, y^-)\right). \quad (2)$$

IPO. IPO uses the same margin but attempts to match it to a target using the following equation:

$$\mathcal{L}_{\text{IPO}} = \left(\Delta_\theta(x, y^+, y^-) - \frac{1}{2\beta}\right)^2. \quad (3)$$

AOT. AOT makes use of the distribution of chosen and rejected scores across a mini-batch. Instead of only comparing each chosen response with the paired rejected response, it sorts the chosen and rejected rewards in the batch and compares the corresponding quantiles:

$$\mathcal{L}_{\text{AOT}} = \frac{1}{B} \sum_{i=1}^{B} -\log \sigma\left(\beta(r^+_{(i)} - r^-_{(i)})\right). \quad (4)$$

3.2 Reward Modeling

For the online methods, we first train a reward model r_ϕ(x,y) that assigns a scalar score to each prompt-response pair. Then we train the reward model using the Bradley–Terry objective:

$$\mathcal{L}_{\text{RM}} = -\log \sigma\left(r_\phi(x, y^+) - r_\phi(x, y^-)\right). \quad (5)$$

This model is then used to score the responses in the online RLHF methods.

3.3 Online RLHF

In the online setting, the policy samples a group of G responses for each prompt. The reward model scores each response, and the policy update depends on how each response performs relative to the other responses sampled for the same prompt. For the default GRPO-style update, the advantage for response j to prompt i is as follows:

$$A_{i,j} = \frac{r_{i,j} - \mu_i}{\sigma_i + \epsilon}, \qquad \mu_i = \frac{1}{G} \sum_{j=1}^{G} r_{i,j}. \quad (6)$$

This makes the reward signal relative within each prompt, rather than comparing directly across unrelated prompts. GRPO then uses these advantages inside a PPO-style clipped policy-gradient objective. DrGRPO expands this update by removing the standard-deviation normalization and changing the length normalization, which changes how reward scale and response length affect learning. GSPO applies clipping at the sequence level rather than the token level, treating the full response more like one structured action.

3.4 Leave-One-Out Advantage Estimation

We next focus on improving the online optimization setting. The default group-relative advantage compares each response to the mean reward of the whole group, including itself. We instead implement an RLOO-style leave-one-out baseline [Ahmadian et al., 2024], where each response is compared only to the other responses sampled for the same prompt:

$$A_{i,j}^{\text{rank}} = \frac{2(\text{rank}_{i,j} - 1)}{G - 1} - 1, \tag{7}$$

The motivation behind this is that each response should be judged relative to alternative responses for the same prompt, but its own reward should not contribute to the baseline. This gives a cleaner within-prompt comparison and may reduce variance in the online policy update. We use this RLOO-style advantage estimator as a direct replacement for the default group-relative advantage while keeping the same reward model, data, base model, and evaluation protocol.

4 Experimental Setup

Baselines. All experiments use the same base model, dataset splits, evaluation files, and token budgets. We first compare the offline methods DPO, IPO, and AOT, and the online methods GRPO, DrGRPO, and GSPO. We then focus on the online setting and use the strongest online method as our main baseline. We then compare this baseline against three online extensions: RLOO, online DPO, and RAFT-style rejection-sampling finetuning.

Online variants. Table 1 summarizes the online variants we evaluate. Each variant keeps the same base model, reward model, data, and evaluation protocol, and changes only the online policy-optimization procedure.

Variant	Base algorithm	Main change	Purpose
Baseline	GRPO	Default group-relative advantages	Online reference baseline
RLOO	GRPO	Leave-one-out advantage baseline	Improve credit assignment
Online DPO	DPO	RM-selected best/worst online pairs	Convert rollouts into preference pairs
RAFT	SFT	Finetune on RM-selected completions	Distill best-of-N samples

Table 1: Online policy-optimization variants.

5 Results

5.1 Offline and Online Methods

Method	Type	Checkpoint	Win Rate
DPO	Offline	`step_000890`	0.78
IPO	Offline	`step_000890`	0.69
AOT	Offline	`step_000890`	0.68
GRPO	Online	`step_000025`	0.74
DrGRPO	Online	`step_000025`	0.80
GSPO	Online	`step_000025`	0.61

Table 2: Head-to-head win rates for the offline and online methods.

Among the offline methods, DPO performed best with a win rate of 0.78. This is a substantial difference than IPO and AOT at 0.69 and 0.68 respectively. Among the online methods, DrGRPO performed best with a win rate of 0.80, followed by GRPO at 0.74 and GSPO at 0.61. GRPO, DrGRPO, and GSPO all use reward-model scores from sampled responses, but differ in how they normalize the advantages and apply the corresponding policy update. DrGRPO appeared to make

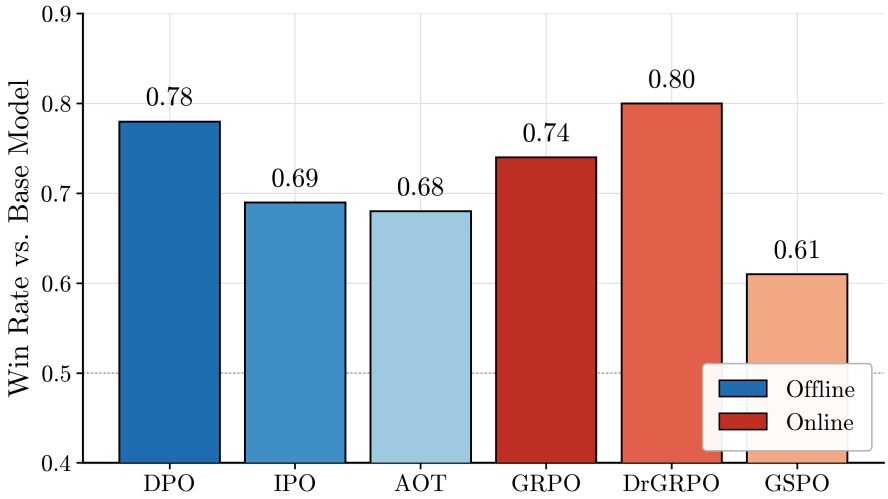

Figure 1: Win rates across all six methods.

the strongest use of the reward signal in this short training run. One misleading internal metric was GSPO's low KL: it stayed close to the reference model, but its generations were weaker and its final win rate was much lower. This shows that low KL alone is not enough and that the policy still needs to move in a useful direction.

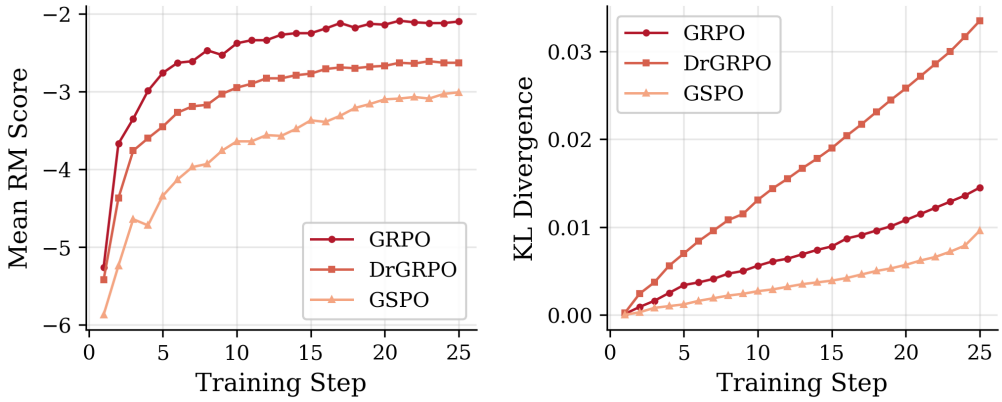

Figure 2: Training dynamics for GRPO, DrGRPO, and GSPO over 25 steps. Left: mean reward-model score. Right: KL divergence from the reference policy.

5.2 Reward Model Diagnostics

Checkpoint	Pair Accuracy	Mean Margin
`step_000250`	0.805	2.58
`step_000350`	0.824	4.13
`step_000445`	0.813	4.72

Table 3: Reward model diagnostics on held-out `test_prefs`.

The reward model performed well on held-out preference pairs. Pair accuracy peaked at 0.824 at step 350, while the mean margin continued increasing through step 445. We used the step-445 checkpoint for online training because it gave the largest separation between chosen and rejected responses. The reward model was not perfect, but its held-out accuracy was strong enough to support online optimization, as shown by the DrGRPO and RLOO results.

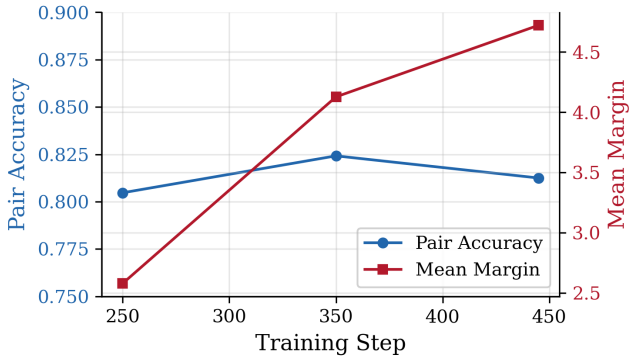

Figure 3: Reward model pair accuracy and mean margin on held-out preferences over training.

5.3 Online Extensions

Method	Main Change	Win Rate
GRPO baseline	Group-normalized advantage	0.74
RLOO	Leave-one-out advantage	0.81
Online DPO	RM-selected preference pairs	0.70
RAFT	RM-selected SFT completions	0.24

Table 4: Online extensions compared against the GRPO baseline.

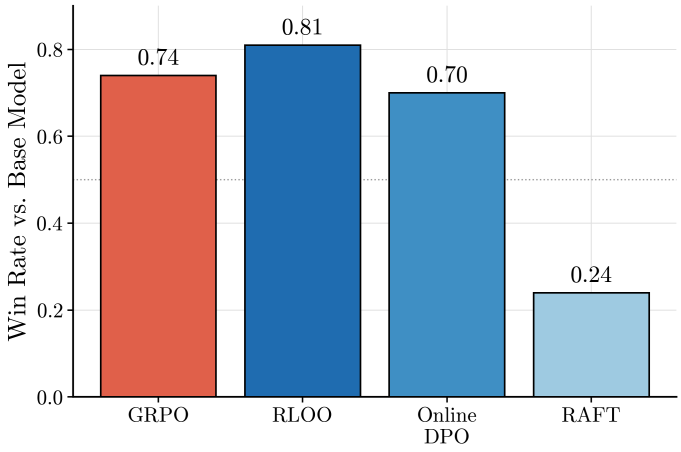

Figure 4: Win rates for the online extensions compared to the GRPO baseline.

We tried three online extensions: RLOO, Online DPO, and RAFT. RLOO was the best method, reaching a win rate of 0.81. This improved over the GRPO baseline at 0.74 while keeping the same reward model, data, and training setup. The main change was the advantage estimator: instead of comparing each response to a group mean that includes itself, RLOO compares each response to the mean reward of the other responses for the same prompt. This gave a cleaner within-prompt learning signal and improved the final judge win rate.

Online DPO performed worse than the GRPO baseline with a win-rate of 0.70. It likely lost useful information by turning a group of scored responses into only one best/worst preference pair. RAFT failed most clearly, with a win rate of 0.24. Since RAFT only finetunes on the highest-reward sampled completions, it was highly sensitive to reward-model mistakes.

Overall, the strongest result came from changing the online objective rather than changing the reward model or evaluation setup. If we had more budget, we would first test RLOO with different reward-

model checkpoints and larger group sizes, since the leave-one-out baseline should become more useful when there are more samples per prompt.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, and Ahmet Üstün. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Peng Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jerret Ross. Distributional preference alignment of llms via optimal transport. *Advances in Neural Information Processing Systems*, 2024.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin Ye, Xinhui Liu, Bowei Zhou, Yangzhen Yan, Tao Gui, Qiuyuan Ma, Jiajun Fei, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.