GIFT: Group-relative Implicit Fine Tuning Integrates GRPO, DPO and UNA

Anonymous ACL submission

Abstract

I propose **G**roup-relative **I**mplicit **F**ine **T**uning (GIFT), a novel reinforcement learning framework for aligning LLMs. Instead of directly maximizing cumulative rewards like PPO or GRPO, GIFT minimizes the discrepancy between implicit and explicit reward models. It combines three key ideas: (1) the online multi-response generation and normalization of GRPO, (2) the implicit reward formulation of DPO, and (3) the implicit–explicit reward alignment principle of UNA. By jointly normalizing the implicit and explicit rewards, GIFT eliminates an otherwise intractable term that prevents effective use of implicit rewards. This normalization transforms the complex reward maximization objective into a simple mean squared error (MSE) loss between the normalized reward functions, converting a non-convex optimization problem into a convex, stable, and analytically differentiable formulation. Despite using a supervised-style MSE loss, GIFT remains a policy optimization method: it optimizes the policy under the same KL-regularized PPO-style objective as RLHF and GRPO, but replaces direct reward maximization with normalized reward matching. Unlike offline methods such as DPO and UNA, GIFT remains on-policy and thus retains exploration capability. Compared to GRPO, it requires fewer hyperparameters, converges faster, and generalizes better with significantly reduced training overfitting. Empirically, GIFT achieves superior reasoning and alignment performance on mathematical and knowledge benchmarks while remaining computationally efficient.

1 Introduction

Modern large language models (LLMs) are pretrained on trillions of tokens to acquire broad linguistic and factual knowledge (OpenAI et al., 2024). Subsequently, supervised fine-tuning (SFT) further enhances their instruction-following capabilities (Brown et al., 2020). Despite these advances, pretrained models often exhibit undesirable behaviors such as bias, unsafe outputs, and weaknesses in reasoning tasks like mathematics and programming.

To mitigate these issues, reinforcement learning (RL)–based alignment methods such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and reinforcement learning with verifiable rewards (RLVR) (DeepSeek-AI et al., 2025) have been proposed. These approaches involve two stages: (1) training a reward model (RM) using the Bradley–Terry (BT) formulation on pairs of preferred and dispreferred responses, and (2) optimizing the policy with proximal policy optimization (PPO) (Schulman et al., 2017) to maximize cumulative reward while constraining divergence from the reference model. GRPO simplifies PPO by replacing the value function with the mean and standard deviation of reward values. While these online RL methods achieve strong alignment performance, they incur significant computational cost and training instability.

To simplify the alignment process, a parallel line of research has explored offline preference-based methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2024). DPO establishes a direct relationship between the implicit reward model and the optimal policy, reformulating RLHF into a single-stage, gradient-based optimization process that eliminates the need for online sampling and significantly reduces computational cost. UNA establishes a special relationship between implicit and explicit reward models, enabling heterogeneous supervision (e.g., preference, binary, or score-based data) through minimizing their discrepancy (Wang et al., 2025). However, offline methods sacrifice exploration and adaptability, leading to weaker performance compared with online RL.

GIFT bridges the gap between these different paradigms. It integrates:

- GRPO's on-policy group-sampling and normalization for stable exploration,
- DPO's implicit reward formulation to optimize a RL objective,
- UNA's implicit–explicit reward alignment for robust MSE supervision.

GIFT retains the exploration benefits of RL-based methods while replacing the non-convex cumulative-reward objective with a convex MSE loss between normalized implicit and explicit reward signals. This reformulation eliminates the intractable normalization constants through normalization, resulting in low-variance gradients and enabling faster, more stable convergence with reduced overfitting and higher benchmark performance.

In summary, GIFT:

- Unifies the core principles of GRPO, DPO, and UNA within a single on-policy fine-tuning framework.
- Preserves exploration while achieving convex, efficient, and low-overfitting optimization.
- Outperforms GRPO in convergence speed, generalization ability, and benchmark performance.

2 Method

2.1 GRPO, DPO, and UNA

This subsection briefly reviews the core formulations of RLHF, RLVR, GRPO, DPO, and UNA; a more detailed discussion is provided in Section C.

In both RLHF and RLVR, policy optimization is typically formulated as a regularized reinforcement learning objective that (i) maximizes the expected reward and (ii) constrains the learned policy to remain close to a reference policy. This trade-off is commonly expressed through a KL-regularized objective, as shown in Equation 1:

$$\pi_\theta^*(y|x) = \arg\max_{\pi_\theta} \ \mathbb{E}_{x\sim D}\Big[\mathbb{E}_{y\sim\pi_\theta(y|x)}[r_\phi(x,y)] - \beta D_{\mathrm{KL}}(\pi_\theta(y|x) \,\|\, \pi_{\mathrm{ref}}(y|x))\Big] \tag{1}$$

Directly optimizing the objective in Equation 1 typically requires learning a value or advantage function in PPO. GRPO simplifies this process by generating multiple responses for the same prompt and using the empirical statistics of the rewards as a surrogate for value estimation. Specifically, rewards are normalized using their batch-wise mean and standard deviation:

$$r'_\phi(x,y) = \frac{r_\phi(x,y) - \mu_\phi}{\sigma_\phi} \tag{2}$$

where μ_ϕ and σ_ϕ denote the mean and standard deviation of the rewards computed over sampled responses.

In contrast, DPO constructs an implicit reward function directly from the policy. As shown in Equation 3, this implicit reward depends on the log-ratio between the current policy and a reference policy, and includes an intractable normalization term $\log Z(x)$:

$$r_\theta(x,y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right) + \beta \log Z(x) \tag{3}$$

In practice, the term $\log Z(x)$ cannot be computed explicitly and is eliminated by subtracting rewards between paired responses, enabling tractable optimization. Therefore, DPO is limited to pairwise preference data.

In UNA, the implicit reward is further simplified by removing the intractable normalization term $\log Z(x)$, resulting in the following special case of the DPO reward:

$$\tilde{r}_\theta(x,y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right) \tag{4}$$

As a result, the implicit reward in UNA is defined pointwise for individual (x, y) pairs, enabling the model to leverage heterogeneous data sources with diverse input formats, supervision types, and reward definitions.

This formulation allows UNA to directly align the policy-induced implicit reward with explicit rewards obtained from scalar feedback, preference signals, or task-specific evaluators by minimizing their discrepancy:

$$L_{\text{UNA-reward}}(\pi_\theta) = \mathbb{E}_{(x,y)\sim D}[r_\phi(x,y) - \tilde{r}_\theta(x,y)]^2 \tag{5}$$

2.2 GIFT Algorithm

The idea of GIFT combines the benefits of 1. RLHF and RLVR with GRPO, 2. DPO and 3. UNA. To be more specific, the online sampling RL objective in RLHF and RLVR in Equation 1 is adopted as the objective for GIFT. Then, the group normalization of GRPO in Equation 2 is utilized to normalize

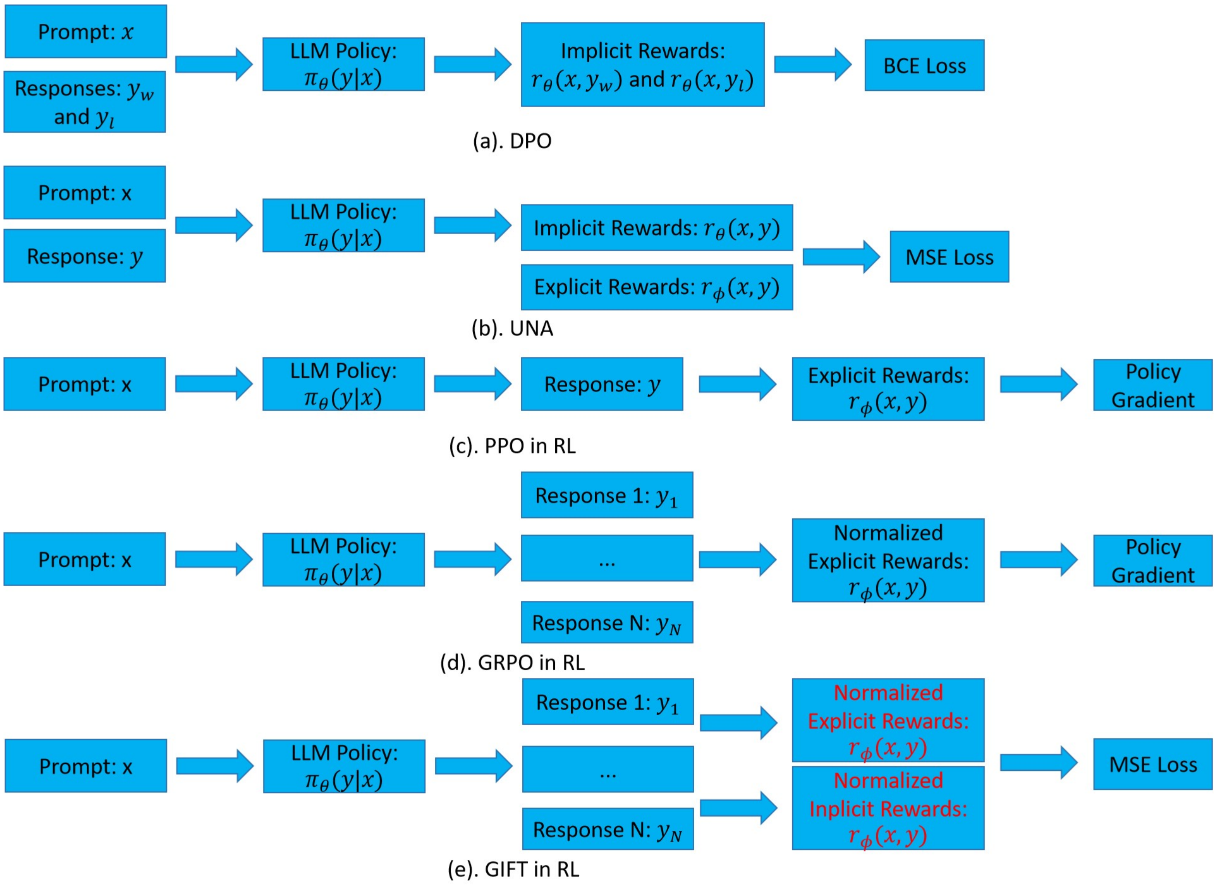

Figure 1: Comparison of different optimization methods: (a). DPO: an offline method with a prompt, a desired response and a undesired response are provided to the LLM policy to generate the implicit reward and a BCE loss is utilized to optimize the policy; (b). UNA: an offline method with a prompt and a response are provided to the LLM policy to generate an implicit reward function and the LLM policy is optimized by minimizing the difference between implicit and explicit reward function; (c). PPO: an online method to generate a response for a prompt and the explicit reward model generates a reward and optimize the LLM through policy gradient; (d). GRPO: an online method to generate multiple responses for a prompt and the explicit reward model generate a reward for each response and the normalized rewards are utilized to optimize the LLM through policy gradient; (e). GIFT: an online method to generate multiple responses for a prompt and the implicit and explicit reward model generate an implicit and explicit reward for each response and the normalized implicit and explicit rewards are utilized to optimize the LLM through minimizing the MSE.

the implicit and explicit reward model. The implicit reward model of DPO in Equation 3 induced a simplified implicit reward model in GIFT. The minimization between implicit and explicit reward model through MSE of UNA in Equation 5 sets the loss function of GIFT.

The general workflow of GIFT can be found in Algorithm 1. For each prompt x sampled from the prompt dataset X. N different responses will be generated by sampling from the LLM policy with high temperature sampling. Based on the prompt x and each sampled response y_i, an explicit reward $r_\phi(x, y_i)$ can be generated, which is from a reward model in RLHF and is from matching the ground truth response in RLVR. Then, for each prompt x and each sampled response y_i, an implicit reward $\hat{r}_\theta(x, y_i) = \log \frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}$ as shown in Equation 6 will be generated. The explicit and implicit rewards are then normalized to derive $r'_\phi(x, y_i)$ and $\hat{r}'_\theta(x, y_i)$ following the workflow of GRPO as shown in Equation 2 or Equation 11. Lastly, the loss is computed as the MSE between the normalized explicit reward function $r'_\phi(x, y_i)$ and the normalized implicit reward function $\hat{r}'_\theta(x, y_i)$ and it is minimized to optimize the LLM policy.

3 GIFT Derivation

The following content provides the mathematical proof of why GIFT works. To begin with, given a prompt x, multiple responses $(y_1, y_2, \cdots, y_N)$ will

Algorithm 1 GIFT Algorithm

for each element x in X **do**
  Generate N different responses $y_1, \cdots, y_N$ from LLM policy $\pi_\theta(y|x)$ by sampling
  Compute explicit reward using the explicit reward model $r_\phi(x, y)$ or $r_\phi(x, y, y^*)$
  Compute implicit reward using implicit reward model $\hat{r}_\theta(x, y)$ with Equation 6
  Normalize explicit and implicit rewards to derive $r'_\phi(x, y)$ and $\hat{r}'_\theta(x, y)$ as shown in Equation 2 and 11
  LLM is optimized by minimizing the mean square error between normalized implicit reward function $\hat{r}'_\theta(x, y)$ and normalized explicit reward function $r'_\phi(x, y)$ as shown in Equation 12
end for
return Fine tuned LLM policy $\pi_\theta(y|x)$

Table 1: Comparison of different optimization methods for fine-tuning including DPO, UNA, PPO, GRPO and GIFT. GN refers to group normalization.

Method	On-Policy	RL	GN	Data Type	Loss
DPO	No	No	No	Pairwise	BCE
UNA	No	No	No	Pointwise	MSE between implicit and explicit reward
PPO	Yes	Yes	No	Pointwise	cumulative reward
GRPO	Yes	Yes	Yes	Pointwise	cumulative reward
GIFT	Yes	Yes	Yes	Pointwise	MSE between implicit and explicit reward

be generated. For each response y_i, its implicit reward will be shown in Equation 6.

$$\hat{r}_\theta(x, y_i) = \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) \tag{6}$$

Compared with the implicit reward function in Equation 3, the hyperparameter β and the intractable term $Z(x)$ are removed as they can be canceled in the normalization process. Its proof will be provided in the following content. To begin with, the mean of these N responses' implicit reward function, i.e., $\hat{\mu}_\theta$ can be calculated as shown in Equation 7.

$$\begin{aligned}\hat{\mu}_\theta &= \frac{1}{N}\sum_{i=1}^{N}\hat{r}_\theta(x, y_i)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right]\end{aligned} \tag{7}$$

In comparison, the implicit reward model in DPO is $r_\theta(x, y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) + \beta \log Z(x)$. As $Z(x)$ is irrelevant to response y, the mean of the implicit reward models for these N responses will be shown in Equation 8.

$$\begin{aligned}\mu_\theta &= \frac{1}{N}\sum_{i=1}^{N} r_\theta(x, y_i)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\beta\log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right] + \beta\log(Z(x))\end{aligned} \tag{8}$$

Then, the term $Z(x)$ will be canceled by subtracting μ_θ from $r_\theta(x, y_i)$ as shown in Equation 9. The obtained $\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta$ will be the same as $\frac{r_\theta(x,y_i)-\mu_\theta}{\beta}$ as shown in Equation 9. In this way, the intractable term $Z(x)$ is canceled out.

$$r_\theta(x, y_i) - \mu_\theta = \beta(\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta) \tag{9}$$

In addition, $Z(x)$ does also not impact the computation of variance as shown in Equation 10.

$$\sigma_\theta^2 = \beta^2 \hat{\sigma}_\theta^2 \tag{10}$$

Eventually, $r_\theta(x, y)$ will be the same as $\hat{r}_\theta(x, y)$ after the normalization process, i.e., $r'_\theta(x, y) = \hat{r}'_\theta(x, y)$ as shown in Equation 11.

$$\begin{aligned}r'_\theta(x, y) &= \frac{r_\theta(x, y_i) - \mu_\theta}{\sigma_\theta}\\ &= \frac{\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta}{\tilde{\sigma}_\theta} = \hat{r}'_\theta(x, y_i)\end{aligned} \tag{11}$$

For the explicit reward model, normalization can be performed following the GRPO workflow, yielding the normalized reward $r'_\phi(x, y)$ as defined in Equation 2. Inspired by UNA, the LLM policy is then optimized by minimizing the discrepancy between the normalized implicit and explicit reward functions, as formulated in Equation 12. Although the temperature term β is canceled during normalization, it should be reintroduced to balance exploration and exploitation in the RL objective (Equation 1).

$$L_{\text{GIFT}}(\pi_\theta) = \mathbb{E}_{(x,y)\sim D}[(r'_\phi(x, y) - \beta \hat{r}'_\theta(x, y))^2] \tag{12}$$

Since the intractable normalization constant $Z(x)$ is canceled out, we can substitute $\hat{r}_\theta(x, y_i) = \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)$ in Equation 6 to replace the compact implicit reward model $r_\theta(x, y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) + \beta \log Z(x)$ from Equation 3. This completes the construction of the GIFT workflow.

3.1 Compare GIFT with PPO, GRPO, DPO and UNA

This subsection compares several methods used to optimize the LLM policy. Firstly, DPO and UNA are off-policy methods, meaning that the responses are pre-collected. In contrast, PPO, GRPO, and GIFT are on-policy methods that generate multiple responses by sampling from the current LLM policy. Because of this multiple-response generation (i.e., group sampling), GRPO and GIFT can perform group normalization, while DPO, UNA, and PPO cannot.

DPO relies on pairwise data (a preferred and a dispreferred response), whereas all other methods use pointwise data—each response is associated with a corresponding reward. Next, DPO and UNA can be viewed as supervised learning methods. Specifically, DPO estimates the probability that one response is preferred over another and then applies a BCE loss to optimize the policy. UNA, on the other hand, minimizes the discrepancy between implicit and explicit reward functions, where the explicit reward (labeled in advance) serves as a supervision signal. In contrast, PPO, GRPO, and GIFT belong to the RL category since they involve on-policy sampling (i.e., exploration) and receive explicit rewards for each sampled response. PPO and GRPO aim to maximize the explicit reward,

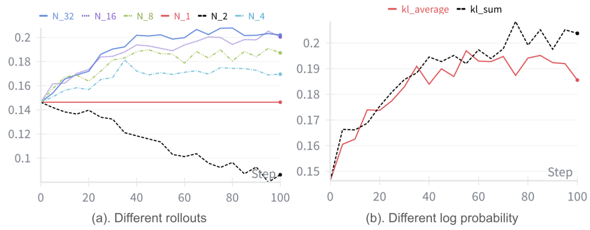

Figure 2: (a) Impact of rollout numbers ($N = 1, 2, 4, 8, 16, 32$) during fine-tuning; (b) Comparison of implicit reward definitions: summation (`kl_sum`) vs. averaging (`kl_average`).

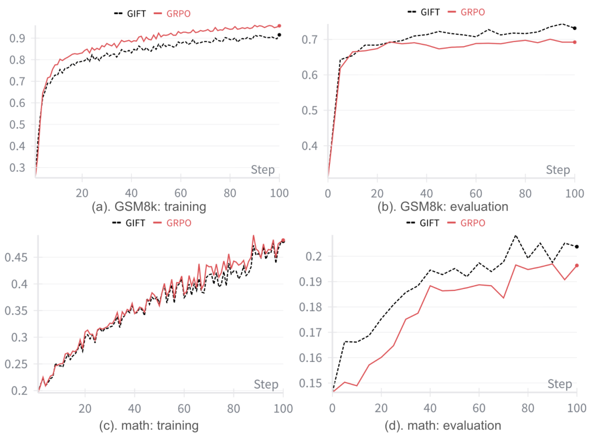

Figure 3: Comparison of GIFT and GRPO on DeepSeek-7B using GSM8K and MATH datasets. Training and evaluation curves show that GRPO exhibits stronger overfitting compared to GIFT.

whereas GIFT minimizes the difference between the explicit and implicit reward functions.

Overall, GIFT offers several advantages over GRPO:

- Easier optimization – The MSE loss used in GIFT is convex, making optimization more stable compared to maximizing cumulative rewards in GRPO.
- Reduced noise and faster convergence – The gradient in GIFT tends to be less noisy without the requirement of gradient clipping, leading to faster convergence.
- Fewer hyperparameters – Lots of clipping parameters are eliminated in GIFT, reducing the search space.
- Better generalization – Experimental results show that GIFT suffers less from overfitting compared with GRPO.

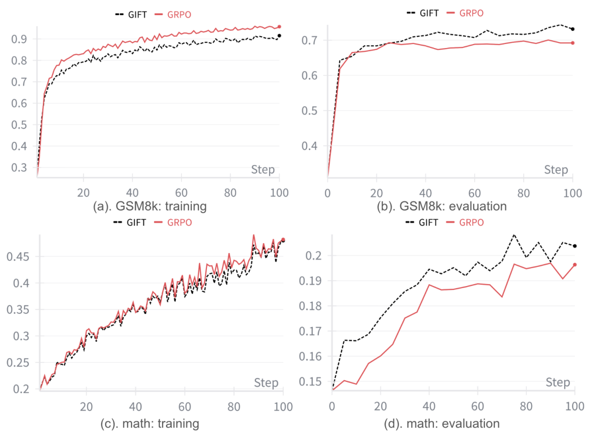

Figure 4: Comparison of GIFT and GRPO on Qwen2.5-32B using GSM8K and MATH datasets. GIFT achieves faster convergence and better generalization.

4 Experiment

In this section, we evaluate multiple variants of GIFT and systematically compare GIFT with GRPO across both RLVR and RLHF settings.

We first conduct RLVR-based comparisons between GIFT and GRPO using two backbone models: (1) `deepseek-ai/deepseek-llm-7b-chat` (DeepSeek-AI et al., 2025) and (2) `Qwen/Qwen2.5-32B-Instruct` (Team, 2024). Experiments are performed on two benchmark datasets: (1) `openai/gsm8k` (Cobbe et al., 2021) and (2) `DigitalLearningGmbH/MATH-lighteval` (Hendrycks et al., 2021).

Next, we extend the RLVR evaluation to a more challenging setting by training on the DAPO dataset (Yu et al., 2025) and evaluating on the AIME dataset (Maxwell-Jia, 2025). These experiments are conducted using `Qwen/Qwen2.5-32B` (Team, 2024) and `Qwen/Qwen3-32B` (Yang et al., 2025).

Finally, we compare GIFT and GRPO under the RLHF paradigm using the `Infinity` dataset (Li et al., 2025). This evaluation is carried out with the same two models as in the initial RLVR experiments: `deepseek-ai/deepseek-llm-7b-chat` and `Qwen/Qwen2.5-32B-Instruct`. For reward model, Skywork/Skywork-Reward-V2-Llama-3.1-8B-40M is utilized (Liu et al., 2025).

Across most experiments, we keep the learning rate and the hyperparameter β consistent to ensure fair comparisons. For GRPO, the learning rate is set to 1×10^{-6} with $\beta = 0.001$. For GIFT, we use a learning rate of 3×10^{-6} and set $\beta = 1$, noting that β can be further tuned if necessary. However, in the RLVR experiments on the DAPO and AIME datasets with `Qwen2.5-32B`, training exhibited instability, and we therefore reduced the learning rates. Specifically, the learning rate is set to 3×10^{-7} for GRPO and 1×10^{-6} for GIFT in these settings.

For most cases, the learning rate and β are the same. For GRPO, the learning rate is set to 1×10^{-6} and $\beta = 0.001$. For GIFT, the learning rate is 3×10^{-6} and $\beta = 1$, where β can be further fine tuned if needed. However, in RLVR on DAPO and AIME dataset for Qwen2.5-32B, the training process is unstable and the corresponding learning rates are decreased. For GRPO, the learning rate is set to 3×10^{-7} while the learning rate for GIFT is set to 1×10^{-6}.

The training batch size is 1024 and the mini-batch size is 256. The maximum prompt length and the maximum response length are 1024 respectively. For both datasets, the total number of training epochs is 15. In all figures, the horizontal axis represents the training steps, and the vertical axis represents the pass@1 rate, i.e., the probability of generating a correct response.

4.1 Different Variants of GIFT

I first investigate the impact of the number of rollouts, testing $N = \{1, 2, 4, 8, 16, 32\}$. As the number of rollouts increases, the normalization process becomes more accurate. As shown in Figure 2(a), larger rollouts improve performance, though the gain becomes marginal when $N > 16$. To balance accuracy and computational cost, I use $N = 16$ in subsequent experiments.

Next, I study two definitions of the implicit reward function, $\hat{r}_\theta(x, y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)$, as given in Equation 6. It can be computed either as the sum of each token's log-probability ratio (Equation 13), referred to as "`kl_sum`," or as their average (Equation 14), referred to as "`kl_average`."

$$\hat{r}_\theta(x, y_i) = \sum_{j=1}^{N} \beta \log\left(\frac{\pi_\theta(y_i^j|x, y_i^1, \cdots y_i^{j-1})}{\pi_{\text{ref}}(y_i^j|x, y_i^1, \cdots y_i^{j-1})}\right) \tag{13}$$

$$\hat{r}_\theta(x, y_i) = \frac{1}{N}\sum_{j=1}^{N} \beta \log\left(\frac{\pi_\theta(y_i^j|x, y_i^1, \cdots y_i^{j-1})}{\pi_{\text{ref}}(y_i^j|x, y_i^1, \cdots y_i^{j-1})}\right) \tag{14}$$

where y_i refers to the i-th response and y_i^j is the j-th token of the i-th response. As shown in

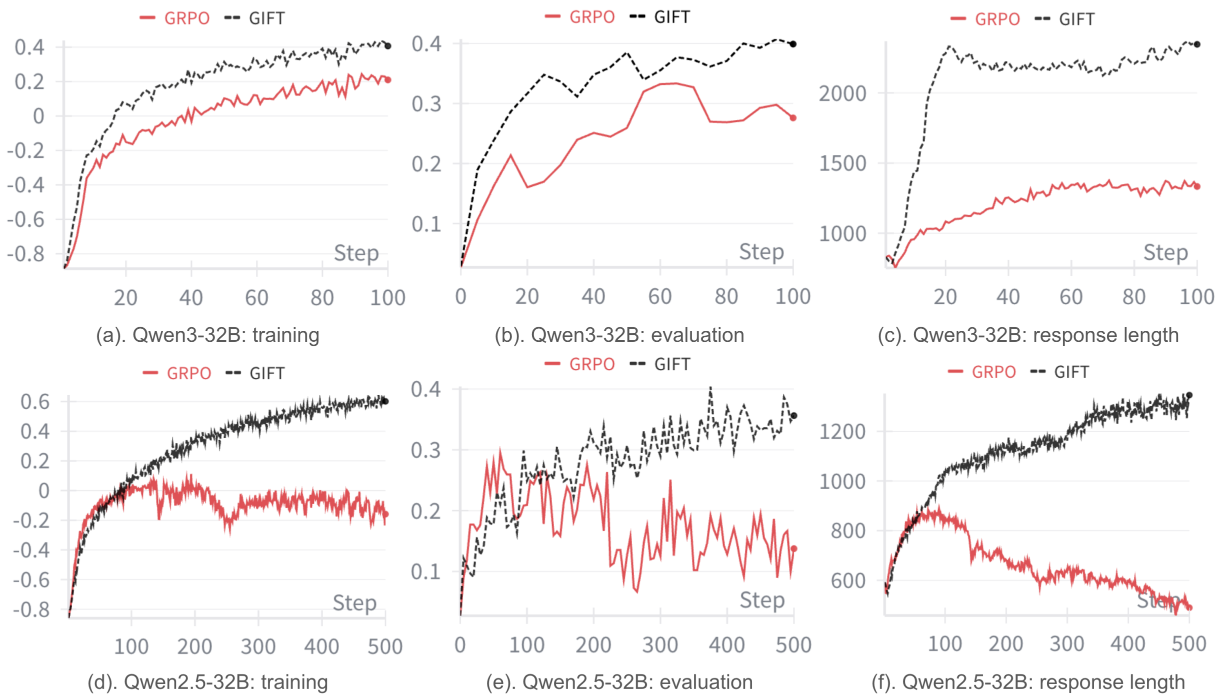

Figure 5: Comparison of GIFT and GRPO on Qwen3-32B-base and Qwen2.5-32B-base using DAPO dataset for training and AIME dataset for evaluation. GIFT achieves faster convergence and better generalization.

Figure 2(b), the "`kl_sum`" formulation achieves a higher pass@1 rate than "`kl_average`." Therefore, "`kl_sum`" is adopted in downstream experiments.

4.2 Comparison between GIFT and GRPO in RLVR

I next compare the performance of GRPO and GIFT. As shown in Figure 3, GIFT consistently outperforms GRPO with a higher pass@1 rate and faster convergence on the `DeepSeek-7B` model. Notably, on the GSM8K task, GRPO achieves higher training accuracy but lower evaluation accuracy than GIFT, indicating more severe overfitting. Similarly, as shown in Figure 4, on the `Qwen2.5-32B` model, GIFT again demonstrates better generalization and faster convergence than GRPO across both GSM8K and MATH tasks.

Both GSM8K and MATH are relatively simple benchmarks. More challenging evaluations on the AIME dataset are presented in Figure 5. Experiments are conducted on both Qwen2.5-32B and Qwen3-32B models. For Qwen3-32B, training converges rapidly, with GIFT exhibiting faster convergence and achieving higher evaluation accuracy than GRPO. In contrast, training on Qwen2.5-32B is less stable, requiring an initial reduction in the learning rate and a longer training schedule. Under this more challenging optimization regime, the advantage of GIFT over GRPO becomes more pronounced. Finally, we observe that GIFT tends to produce longer response lengths, as shown in Figure 5(c) and (f).

4.3 Comparison between GIFT and GRPO in RLHF

Beyond RLVR, GIFT can also be effectively applied in the RLHF setting to enhance instruction-following capabilities. Unlike RLVR, which relies on verifiable rewards, RLHF employs a learned reward model. Such reward models are known to exhibit length-related bias, often favoring longer responses. To mitigate this issue, we normalize the reward by the square root of the response length.

As shown in Figure 6, GIFT consistently outperforms GRPO on both Qwen2.5-7B and Qwen2.5-32B models. This advantage is further confirmed by downstream evaluations, where GIFT achieves superior performance on nearly all benchmarks, as summarized in Table 2 and Table 3.

Specifically, Table 2 reports results on seven diverse evaluation tasks: TruthfulQA (Lin et al., 2022), BBQ (Parrish et al., 2022), MBPP (Austin et al., 2021), ARC-Challenge (Clark et al., 2018), Winogender (Rudinger et al., 2018), GPQA (Rein et al., 2023), and MUSR (Sprague et al., 2024). Across all seven tasks, GIFT consistently surpasses GRPO for both the 7B and 32B model sizes.

For generation-oriented evaluations such as Al-

Figure 6: Comparison of GIFT and GRPO on Qwen2.5-7B-Instruct and Qwen2.5-32B-Instruct using INFINITY datasets. GIFT achieves faster convergence and better generalization.

Model Name	TruthfulQA	BBQ	MBPP	ARC-C	Winogender	GPQA	MUSR
7B-Instruct	64.79	54.62	48.2	64.59	60.97	31.46	42.86
7B-GRPO	66.93	53.79	62.0	64.25	59.86	30.87	45.77
7B-GIFT	**69.05**	**58.08**	**62.8**	**65.7**	**61.39**	**33.98**	**46.16**
32B-Instruct	65.58	59.04	75.8	71.93	61.11	**38.26**	**50.13**
32B-GRPO	69.09	65.69	75.8	72.27	64.31	37.16	48.68
32B-GIFT	**70.02**	**71.06**	**78.2**	**73.89**	**64.72**	37.25	49.47

Table 2: Selected benchmarks highlighting tasks where GIFT shows strong or leading performance

pacaEval (Dubois et al., 2024) and Arena-Hard (Li et al., 2024), GIFT also outperforms GRPO in almost all settings. The only exception is the 7B style-control configuration, where GRPO achieves a marginally higher score than GIFT.

5 Conclusion

I introduced GIFT, a reinforcement learning framework that unifies the strengths of online and offline alignment methods. Unlike GRPO, which directly maximizes cumulative normalized rewards, GIFT minimizes the mean squared error between normalized implicit and explicit reward functions. This reformulation eliminates the intractable normalization constant inherent to implicit rewards, transforms the optimization into a convex objective, and produces low-variance gradients.

By combining GRPO's online group normalization with DPO's implicit reward formulation and UNA's reward alignment principle, GIFT retains exploration while achieving efficient, stable, and scalable fine-tuning. Experiments on both 7B and 32B models demonstrate that GIFT converges faster, generalizes better, and exhibits less overfitting than GRPO across math and knowledge benchmarks.

In essence, GIFT reframes RL for LLM alignment—from noisy reward maximization to structured reward matching—bridging the conceptual divide between online RL and offline preference optimization. Future work may extend GIFT to multi-modal and multi-agent settings, where normalized implicit–explicit reward coupling can further enhance alignment stability and reasoning capability.

6 Limitation

The limitation of this work lies in testing on larger models like 200B parameter models with more complex datasets. In addition, the test on multi-lingual datasets and industrial datasets should be conducted.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, and et al. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, and et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, and et al. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, and et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yann Dubois, Bal'azs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, and et al. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jijie Li, Li Du, Hanyu Zhao, Bo wen Zhang, Liangdong Wang, and et al. 2025. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *Preprint*, arXiv:2506.11116.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From live data to high-quality benchmarks: The arena-hard pipeline. `https://lmsys.org/blog/2024-04-19-arena-hard/`.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, and et al. 2025. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

Maxwell-Jia. 2025. Aime 2024 dataset. `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`. Accessed: 2025-01-04.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, and et al. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, and et al. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, and et al. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and et al. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, and et al. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. *Preprint*, arXiv:1804.09301.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, and et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Zhichao Wang, Bin Bi, Can Huang, Shiva Kumar Pentyala, Zixu James Zhu, and et al. 2025. Una: Unifying alignments of rlhf/ppo, dpo and kto by a generalized implicit reward function. *Preprint*, arXiv:2408.15339.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, and et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, and et al. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.

Model Name	Alpaca-eval		Arena-hard		
	LC Win Rate	Win Rate	Response Length	Style Control	Creative Writing
7B-Instruct	30.81	29.05	1989	2.5	5.2
7B-GRPO	35.33	38.25	2203	**3.2**	8.5
7B-GIFT	**48.77**	**72.43**	4291	3.0	**17.2**
32B-Instruct	43.49	35.76	1754	4.4	3.9
32B-GRPO	53.13	60.77	2369	7.7	21.1
32B-GIFT	**61.43**	**78.53**	3327	**15.3**	**47.9**

Table 3: Model performance on Alpaca-eval and Arena-hard benchmarks

A Compare GIFT with GRPO on RLHF: Tabular Results

One last results of comparing GIFT with GRPO on RLHF is shown in Table 3 for space limitation.

B GIFT Derivation

$$\begin{aligned} r_\theta(x, y_i) - \mu_\theta &= \left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) + \beta \log Z(x)\right] - \left[\frac{1}{N}\sum_{i=1}^{N}\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right] + \beta \log Z(x)\right] \\ &= \beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) - \frac{1}{N}\sum_{i=1}^{N}\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right] = \beta(\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta) \end{aligned} \tag{15}$$

$$\begin{aligned} \sigma_\theta^2 &= Var\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) + \beta \log Z(x)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\left\{\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) + \beta \log Z(x)\right] - \left[\frac{1}{N}\sum_{i=1}^{N}\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right] + \beta \log Z(x)\right]\right\}^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right) - \frac{1}{N}\sum_{i=1}^{N}\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right]\right]^2 \\ &= Var\left[\beta \log\left(\frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)}\right)\right] = \beta^2 \hat{\sigma}_\theta^2 \end{aligned} \tag{16}$$

$$r'_\theta(x, y) = \frac{r_\theta(x, y_i) - \mu_\theta}{\sigma_\theta} = \frac{\beta(\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta)}{\beta\hat{\sigma}_\theta} = \frac{\hat{r}_\theta(x, y_i) - \hat{\mu}_\theta}{\tilde{\sigma}_\theta} = \hat{r}'_\theta(x, y_i) \tag{17}$$

C Related Methods

C.1 RLHF and PPO

After the pretraining and SFT stages, LLMs may still produce undesirable or suboptimal responses. To further improve their alignment and response quality, RLHF is applied. RLHF typically consists of two major stages, each leveraging a distinct dataset. In the reward model training stage, the dataset comprises triplets of the form (x, y_w, y_l), where x denotes the prompt, y_w represents the preferred (winning) response, and y_l represents the dispreferred (losing) response. In the RL fine-tuning stage, the dataset contains only task prompts x, which correspond to the domains or behaviors that the LLM aims to improve through policy optimization.

During reward model training, an explicit reward model is learned to predict a scalar score $r_\phi(x, y)$ for a given prompt–response pair (x, y). The model is trained on pairwise preference data (x, y_w, y_l), where y_w denotes the preferred response and y_l the dispreferred one. The probability that y_w is favored over y_l, denoted as $P_\phi(y_w > y_l \mid x)$, is modeled using the Bradley–Terry (BT) framework, i.e., $P_\phi(y_w > y_l \mid x) =$

$\frac{e^{r_\phi(x,y_w)}}{e^{r_\phi(x,y_w)}+e^{r_\phi(x,y_l)}} = \sigma(r_\phi(x,y_w) - r_\phi(x,y_l))$ where $\sigma(\cdot)$ denotes the logistic sigmoid function (Bradley and Terry, 1952). Intuitively, the higher the difference $r_\phi(x,y_w) - r_\phi(x,y_l)$, the greater the likelihood that the model predicts y_w as more aligned with human preference. Eventually, the RM is trained through a binary cross entropy (BCE) loss $L_{\text{RM}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\sigma(r_\phi(x,y_w) - r_\phi(x,y_l))\right]$ (Ouyang et al., 2022).

RL fine-tuning serves two principal objectives. First, it seeks to maximize the pretrained explicit reward function $r_\phi(x,y)$ so that the policy $\pi_\theta(y|x)$ becomes better aligned with human preferences encoded by the reward model. Second, to mitigate *reward hacking* and preserve fidelity to the pretrained model, a Kullback–Leibler (KL) regularization term is introduced, penalizing large deviations from the reference policy $\pi_{\text{ref}}(y|x)$. The combined optimization problem can be expressed as shown in Equation 18:

$$\pi_\theta^*(y|x) = \arg\max_{\pi_\theta} \mathbb{E}_{x\sim D}\left[\mathbb{E}_{y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta D_{\text{KL}}(\pi_\theta(y|x) \,\|\, \pi_{\text{ref}}(y|x))\right], \tag{18}$$

where β is a temperature-like coefficient that controls the trade-off between reward maximization and policy divergence. **The same objective is adopted in RLVR, DPO, UNA and GIFT.**

To stabilize training, Proximal Policy Optimization (PPO) (Schulman et al., 2017) is commonly adopted to optimize the objective in Equation 18. However, RLHF requires maintaining multiple large-scale components—namely, a policy model, reference model, reward model, and value model—resulting in substantial computational and memory overhead. These drawbacks significantly restrict the scalability and practicality of RLHF for LLMs.

C.2 RLVR and GRPO

While RLHF has proven effective at aligning LLMs with human preferences and mitigating undesirable or harmful outputs, it often falls short in enhancing complex reasoning capabilities such as mathematical problem solving or program synthesis. To overcome this limitation, RLVR has been proposed, which only has the RL fine-tuning stage as the explicit reward model has been replaced with verifiable reward. In RL fine-tuning stage of RLVR, the dataset contains (x, y^*) where y^* refers to golden verifiable response. In contrast to RLHF, which depends on a learned and potentially imperfect reward model trained from human feedback, RLVR leverages verifiable signals—such as the correctness of a numerical solution or the success of generated code in passing predefined test cases—as rewards $r_\phi(x,y,y^*) = \begin{cases} 1, & \text{if } y = y^* \\ 0, & \text{if } y \neq y^* \end{cases}$. These verifiable signals, although sparse, provide precise and unambiguous feedback that directly reflects task success.

To reduce the computation cost of PPO, GRPO is then proposed (Shao et al., 2024; Zheng et al., 2025) as shown in Equation 19 where $r'_\phi(x,y)$ refers to the normalized explicit reward model, μ_ϕ refers to the mean of the explicit reward model and σ_ϕ refers to the standard deviation of the explicit reward model.

$$r'_\phi(x,y) = \frac{r_\phi(x,y) - \mu_\phi}{\sigma_\phi} \tag{19}$$

When combined with the GRPO framework, as demonstrated in DeepSeek R1 (DeepSeek-AI et al., 2025), RLVR substantially enhances an LLM's reasoning capacity. This integration enables the model to sustain longer and more coherent chains of thought (CoT), exhibit improved self-consistency across reasoning steps, and even display emergent "aha moments," where the model identifies and corrects its own logical or computational errors during inference.

C.3 DPO

To address the instability and inefficiency of RLHF and PPO, DPO was proposed as a simplified offline alternative utilizing the same reward model training's dataset, i.e., (x, y_w, y_l) in RLHF (Rafailov et al., 2024). Based on the RL objective in Equation 18, DPO eliminates the explicit reward model and build a mapping between an implicit reward model and the optimal policy as shown in Equation 20

$$r_\theta(x,y) = \beta\log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) + \beta\log Z(x) \tag{20}$$

where $r_\theta(x,y)$ is an implicit reward function and $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r_\theta(x,y)\right)$ is the partition function ensuring normalization, which is intractable. Substituting this implicit reward formulation into the reward model training objective cancels out the $Z(x)$, yielding the DPO loss $L_{\text{DPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\sigma(r_\theta(x,y_w) - r_\theta(x,y_l))\right] = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$ where (y_w, y_l) denote the preferred and dispreferred responses, respectively. Optimizing L_{DPO} aligns the model directly with preference data without requiring an explicit reward model or a separate RL training loop. Consequently, DPO unifies the reward modeling and policy optimization stages of RLHF into a single, tractable procedure, greatly simplifying implementation and reducing memory cost.

Nonetheless, DPO introduces its own set of challenges. Since $Z(x)$ cannot be computed explicitly, DPO relies solely on pairwise preference data, rendering single-response supervision unusable during fine-tuning. This dependence on pairwise data—typically limited and expensive to collect—can lead to inefficient utilization of available feedback. Moreover, while RLHF's reward model can provide continuous-valued feedback for arbitrary prompts, DPO only learns from pairwise comparisons, limiting the granularity of optimization signals.

C.4 UNA

One limitation in DPO is that it can only utilize pairwise dataset, while other format dataset cannot be utilized. In addition, the pairwise preference data provide less information compared with detailed information provided by reward mode. UNA extends DPO to pointwise dataset composed of (prompt, response, reward), i.e., (x, y, r). In particular, UNA discovers that Equation 21 is also a special optimal implicit reward function for the RL objective in Equation 18 based on DPO's implicit function in Equation 20 (Wang et al., 2025).

$$\tilde{r}_\theta(x,y) = \beta\log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) \tag{21}$$

Because the annoying intractable $Z(x)$ has been removed, UNA can expanded DPO to different types of signals rather than only pairwise signals. Eventually, the LLM policy is optimized through the MSE between implicit and explicit reward model as shown in Equation 22.

$$L_{\text{UNA-reward}}(\pi_\theta) = \mathbb{E}_{(x,y)\sim D}[(r_\phi(x,y) - \tilde{r}_\theta(x,y))^2] \tag{22}$$