Focus on the Optimization of the RLHF Algorithm to Enhance the Training Effect After LLM

Shuyang He[1*], *Xi* Yu[2], *and Xiubin* Zhang[3]

[1] School of Systems Science and Statistics, Beijing Wuzi University, Beijing, China
[2] College of Artificial Intelligence, Tianjin University of Science and Technology, Binhai New Area, Tianjin, China
[3] Hefei No.8 Senior High School, Hefei, Anhui, China

Abstract. Post-training alignment of large language models is crucial for ensuring their safety, usefulness, and alignment with human preferences. Although reinforcement learning from human feedback (RLHF) is the mainstream approach, its training process is susceptible to issues such as reward model noise, unstable policy optimization, and catastrophic forgetting, leading to deviations between model outputs and true human preferences. This paper systematically reviews the optimization algorithms for enhancing the stability of RLHF in the past three years and categorizing them into three types: "Modifying signals", aiming to improve the reliability of the reward signal; "Optimizing algorithms", focusing on enhancing the performance of the reinforcement learning model to increase its noise resistance; "Developing Models", reconstructing the training framework and multi-module collaboration from a system perspective to achieve fundamental optimization. This paper elaborates on the principles, representative algorithms, and advantages and disadvantages of each category, and conducts a horizontal comparison and analysis. This review aims to provide researchers with a clear algorithm classification framework and selection reference, while also pointing out the common challenges in current research, such as reward model bias, system implementation complexity, and generalization ability, in the hope of promoting subsequent research progress.

1 Introduction

Large scale language models (LLMs) have become the core technology in the field of artificial intelligence, but their output often deviates from human real preferences due to noise, bias and other contents in the training data, causing alignment problems [1]. To solve this problem, reinforcement learning human feedback (RLHF) came into being. This method uses the training reward model to learn human preferences, and combines with the reinforcement learning algorithm to optimize the strategy model, which significantly improves the relevance, usefulness and security of the generated results, and has become the mainstream scheme of post training LLM [2][3].

* Corresponding author's email: 2421820036@bwu.edu.cn

However, RLHF still faces severe challenges in practical application. Its training stability is highly dependent on the quality of the reward model (RM), and the inconsistency and bias of human feedback cause the reward signal to carry systematic noise, which can easily cause "reward hackers" [4]. At the same time, the excessive dependence of the model on RM restricts its long-term stability and generalization ability. In addition, reinforcement learning (RL) algorithms such as PPO may aggravate "reward hackers" due to excessive pursuit of high scores at the fine-tuning stage, or lead to "catastrophic forgetting" due to one-sided optimization of high score samples, and even lead to training failure [5][6]. Therefore, improving the stability of RLHF training has become the key to breaking through the bottleneck of LLM alignment.

At present, there are many optimization algorithms for RLHF stability in the field, but they lack systematic sorting. This paper systematically reviews the relevant research results in the past three years, and divides the mainstream optimization algorithms into three categories: "Modifying signals", "Optimizing algorithms" and "Developing Models", respectively from the theoretical principles and specific applications, and makes a horizontal comparative analysis, clearly showing the applicable scenarios, advantages and disadvantages of each optimization idea.

2 Classification of Mainstream Algorithms

In response to the existing problems of RLHF, corresponding optimization algorithms in vertical fields are emerging in an endless stream. By collecting relevant papers published in top conferences and journals in the past three years, this paper classifies the mainstream optimization algorithms into three approaches: "Modifying signals", "Optimizing algorithms", and "Developing models". The following text will respectively summarize the three types of optimization approaches and provide examples for illustration.

2.1 "Modifying signals"

In the RLHF framework, RM carries systematic noise, which has become a key bottleneck restricting the stability of the whole training process [7]. This kind of noise comes from the fuzziness, inconsistency and subjective preference difference of human feedback data, which is difficult to be completely eliminated by conventional data cleaning or model structure adjustment. Aiming at this systematic problem, some algorithms put forward the idea of optimizing the reward signal. The essence of the "Modifying signals" strategy is to actively intervene and reprocess the reward signal. This processing method does not pursue the absolute accuracy of RM, but is committed to improving the relative reliability and availability of its output signal. From the perspective of the system, the "Modifying signals" method is similar to "noise elimination at the signal end". A "signal filter" is added before the strategy model receives the reward, which effectively blocks the chain amplification of noise in the RL iteration process, so that the model can more accurately identify and strengthen the behavior patterns that really meet human expectations in the process of fine-tuning, and improve the noise robustness of the model.

2.1.1 Specific algorithm example

"Comparative reward" algorithm. The core of the "comparative reward" algorithm is to convert absolute rewards into relative rewards through "offline sampling+relative difference calculation" to offset common noise [1]. The algorithm is mainly divided into three steps.

The first is the off-line sampling phase. For each prompt, the SFT model is used to sample K responses to form a baseline response set, and RM is used to score and calculate the average reward as the baseline reward. The second is to compare the reward structure. In the reinforcement learning model training phase, the RM score of the response generated by the current strategy is subtracted from the baseline reward of the corresponding prompt to form a comparative reward signal, so as to alleviate the impact of RM noise. The third is dynamic scaling. In order to maintain the stability of training, the comparative reward is linearly scaled according to the scale of the original reward to avoid the change in reward scale.

PF-PPO. The core of PF-PPO is to actively eliminate low reliability signals through "sample screening" and retain high confidence samples [8]. Firstly, the sample reliability is judged by the determination coefficient R ^ 2; In the RL sampling phase, N responses are generated for each prompt and sorted. After sorting by reward score, high/low score samples are selected by weight for training. By discarding the medium fraction samples with dense noise, the strategy model can only learn in the RM credible signal interval to reduce noise misleading.

2.1.2 Advantages and disadvantages of optimization ideas

The idea of "Modifying signals" has obviously solved some objective problems that affect the stability of RLHF training through the artificial processing of reward signals. It has its own advantages, but it also has disadvantages in theory and practice.

First of all, the innovative optimization algorithm under the idea of "Modifying signals" has low cost. Its optimization behavior focuses on the reoptimization process after RM generates the reward signal. It does not need to reconstruct the RM training process. It can be directly embedded in the existing RLHF, compatible with the current RLHF framework, and does not need additional annotation data. Secondly, it can solve some problems in the RL stage. The improvement of reward signal quality by these kinds of ideas can make the reinforcement learning model receive more accurate signals, improve the accuracy of strategy learning in the RL stage, and enhance the ability of generating strategy alignment.

Behind the advantages, its thinking defects are also obvious. The algorithm can not cure the systematic deviation of RM, which requires the minimum quality of RM. If RM has a serious deviation, the effective signal is scarce and the optimized signal function cannot be reflected. For example, comparative rewards may take the relative difference of errors as an effective signal, and PF-PPO may screen out samples with "high ranking but actual errors", resulting in a narrow range of performance improvement.

2.2 "Optimizing algorithms"

"Optimizing algorithms" means starting from the training phase itself and improving the strategy optimization process to promote stability of RLHF. This approach breaks free from the limitation of "Modifying signals" and chooses to optimize the performance of the RL training model itself. It improves the noise robustness of the RL model at the design level, preventing the policy model from falling into "reward hacking" or "catastrophic forgetting" resulting in the pursuit of high scores.

2.2.1 Specific algorithm example

In this way, the example includes REINFORCE++(R++), GRPO, ReMax and so on. Their common principle is to reduce the variance of gradient estimation and prevent the policy from overfitting to specific prompts or reward signals. Like R++, the core innovation lies in

abandoning the traditional Critic network and local baseline, and adopting the “global batch normalization” strategy.

First is to calculate the advantage value A t. For the t-th token in the generated sequence, its advantage value is defined as the cumulative reward from t to the end of the sequence minus the KL penalty term and the baseline:

$$At = t\sum Trt' - \beta \log \pi ref\ (yt'|x)\pi\theta(yt'|x) - b \quad (1)$$

Among this: rt' is encouragement of token level, β is KL modulus, π θ is Current policy, π ref is reference model,b is base line.

Second is global normalization: During each training batch, calculate the mean μ A and standard deviation σ A of all A t, then normalize each advantage value to stabilize its distribution within a range of mean 0 and standard deviation 1, thereby reducing gradient variance [9].

Third is strategy optimization. The normalized advantage value A t replaces the original advantage value in the PPO objective function optimization, while retaining the Clipped Policy Optimization “Clip” mechanism to ensure controllable update magnitude.

R++ 's global normalization effectively suppresses gradient fluctuations, enhancing training stability and generalization capabilities. Experimental results demonstrate its superiority over PPO on benchmarking tasks like MT-Bench and RED-EVAL, particularly showing stronger security against adversarial' jailbreak' attacks [9]. Additionally, R++ employs an unbiased K2 estimator for KL divergence calculation, avoiding the bias issues inherent in traditional methods and further ensuring the accuracy of policy updates.

GRPO also adopts a critic-free design and simplifies training through intra-group score normalization, though its normalization scope is limited to multiple responses under a single prompt, making it less robust than R++'s global normalization [10]. Another algorithm, ReMax, introduces greedy search samples as a baseline, which reduces variance but incurs higher online generation costs [9].

2.2.2 Advantages and disadvantages of optimization ideas

The “Optimizing algorithms” approach demonstrates distinct advantages by eliminating reliance on external signal corrections. It directly enhances the algorithm's inherent noise resistance, specifically addressing RL model limitations to improve policy learning capabilities. Through iterative refinement, it achieves seamless alignment with human preferences, effectively tackling the alignment problem. Crucially, this method requires no reconstruction of the RLHF framework, only performance optimization on existing systems, making it cost-effective and straightforward to implement. Moreover, it effectively mitigates catastrophic forgetting by enabling targeted knowledge acquisition, preventing issues like “reward hacking” and over-optimization caused by flawed learning strategies. However, this approach has limitations. First, its dependence on RM only improves noise robustness without fundamentally addressing signal-noise interference. When RM errors are excessive, the model's policy may still deviate from real-world objectives. Additionally, the baseline calculation lacks theoretical grounding and dynamic adjustment capabilities, restricting the model's multitasking and cross-domain processing abilities.

2.3 Developing Models

This optimization approach is to start from the RLHF framework for overall structural optimization, and redesign or improve the structure and interaction mechanism of the entire RLHF training framework.

The core idea of this systematic framework optimization approach lies in regarding the RLHF framework as a multi-stage and multi-module collaborative dynamic system. It designs the model structure from a macro perspective, introduces a collaborative mechanism while ensuring the performance of a single model, and connects the reward stage with the RL fine-tuning training stage, thereby enhancing the overall performance of the model to a greater extent.

When designing specific models at each stage, by introducing original modeling principles, training mechanisms or data scheduling strategies, the unified coordination and local optimization of key links such as signal output, strategy optimization and knowledge reinforcement learning are achieved, effectively getting rid of the influence of systematic problems of the original model on the RLHF model.

2.3.1 Specific algorithm example

The "AM + SR Stabilization RLHF framework" is a new algorithm that has emerged in recent years under this optimization approach. This section takes this algorithm as an example to explicitly explain the principle of the systematic framework optimization approach.

Advantage Model (AM). The AM model first forms an increment representing the response relative to the average level of the task by modeling the advantage score. Then the model introduces the Bounding Loss function to standardize the distribution of dominant scores[11].

First is parameterized expected reward:

$$e_\tau(x) = E_{y\sim\pi\phi(x)}[r_\theta(x,y)] \tag{2}$$

The estimated expected reward value provides a benchmark for the subsequent generation of advantage values.

Second is the Modeling Advantage score:

$$a_\theta(x,y) = r_\theta(x,y) - e_\tau(x) \tag{3}$$

Based on the expected reward value, generate the relative advantage value to demonstrate the "additional benefit" of the reward, making samples under different distributions comparable [11].

Third is to combine the Bounding Loss and the bounding loss function:

$$L\_AM = -E\_((x,y\_c,y\_r)\sim D\hat{}RM)\,[log(\sigma(a\_\theta(x,y\_c) - a\_\theta(x,y\_r))) + log(\sigma(m(x) - a\_\theta(x,y\_c))) + log(\sigma(m(x) + a\_\theta(x,y\_r)))] \tag{4}$$

Among them, it refers to the preferred response, refers to the non-preferred response, refers to the margin value of the loss function, and controls the range of the advantage score.

The loss function constrains the value range of the dominant score through the margin value, achieving dynamic adjustment of the mean and variance of the reward distribution, as detailed in Formula (5).

$$a_\theta(x,y_c) \le m(x)、\ a_\theta(x,y_r) \ge -m(x) \tag{5}$$

Selective Rehearsal (SR). The SR model embeds prompts using SimCSE encoding. The prompts are classified based on different features by using clustering algorithms, and then the samples with the highest advantage scores in each group are selected. These representative samples are combined to form a retraining dataset. The model effectively prevents the forgetting of model knowledge by introducing NLL review loss. Besides, the retraining phase is based on the replay set for repeated learning and knowledge reproduction [11].

AM and SR can independently solve targeted problems. When combined, they generate stronger system-level stability and form a feedback loop.AM identifies high-quality responses, SR discovers excellent samples based on AM scoring and incorporates them into the replay set, reinforces the behaviors of these samples during the retraining phase, and AM further learns more consistent scoring criteria [11]. Under the collaborative effect of the two models, a positive cycle is formed, gradually converging the system to a strategic space that is both efficient and robust.

2.3.2 Advantages and disadvantages of optimization ideas

Firstly, its independently developed novel mechanisms applied in processes such as modeling, policy training, and data collection can fundamentally solve the RLHF training problems caused by systematic issues of traditional models, and enhance model performance in an efficient and targeted manner. Secondly, the model as a whole adopts an independent innovation mechanism to enhance the coordination among various independent models and avoid additional systematic problems caused by mechanism incompatibility. Build collaborative interaction among models to stimulate greater optimization benefits.

In addition, by using a goal-oriented and global perspective to design algorithm models, the purposefulness of model optimization is enhanced, making it more targeted in solving human alignment problems. Meanwhile, the macroscopic framework optimization achieves the effect of one algorithm solving multiple problems simultaneously, without the need for researchers to complete the integration of multiple models by themselves. It is convenient to apply and has a high optimization efficiency.

Although a well-developed optimization system demonstrates numerous advantages, it also has objective flaws at the same time. Firstly, large-scale autonomous modeling and collaborative model operations not only incur high experimental costs in modeling, data collection and organization, and model checking but also have high implementation complexity, and their feasibility needs to be improved. Secondly, as this idea is an emerging optimization approach in the field and the corresponding algorithm has a relatively high implementation complexity, it leads to a scarcity of open-source code for the algorithm model and a small number of model tests, resulting in disputes over the theoretical basis and optimization effect. More importantly, the strong innovative attribute of this optimization idea is supported by powerful theoretical knowledge and technology, as well as a keen sense of scientific research. It requires a high level of comprehensive quality from the developers themselves and incurs a large time cost.

3 Horizontal Comparative Analysis and Discussion on the Optimization Ideas

Based on the analysis of the characteristics, advantages and disadvantages of different optimization ideas, this paper makes a horizontal comparison of the three types of ideas and

summarizes their similarities and differences. It is convenient for researchers to select different optimization paths according to different needs.

3.1 "Modifying signals" vs "optimizing algorithms"

The difference between the two lies in the different objects of optimization. "Modifying Signals" is aimed at optimizing the reward signal, while "Optimizing Algorithms" is aimed at enhancing the performance of the RL algorithm. However, the general optimization approach is to enhance the robustness of the model's reward noise through human means, reduce the deviation between the generated results and human true preferences, and thereby solve the alignment problem.

Both are based on the local optimization of the existing RLHF framework for different stages, which indicates that the algorithms under both types of thinking can be directly nested with the RLHF model without the need to rebuild the model framework, resulting in lower engineering complexity and lower usage cost.

In addition, the characteristics of local optimization enable the algorithms under the two approaches to be flexibly combined with other optimization algorithms according to the researcher's needs, facilitating the completion of algorithm fusion optimization RLHF work and each performing its own duties to specifically solve local problems. Inevitably, both types of optimization ideas rely on RM and cannot fundamentally solve the noise problem of the reward model. When there is a significant deviation in RM, there is still a chance that the model training will fail. This is the risk that the second type of optimization approach needs to bear during the working process.

3.2 "Developing models" vs. "modifying signals" and "optimizing algorithms"

Both "Modifying Signals" and "Optimizing Algorithms" are local optimizations, while "Developing Models" conducts global optimization from a more comprehensive perspective. The "Building Models" approach regards the signal output stage of RLHF and the fine-tuning stage of RL as a whole rather than a part. While solving problems in each link, it introduces the concept of collaborative optimization and pays more attention to overall optimization. It avoids the problem of resource mismatch during algorithm fusion optimization, and the overall improvement can more fully display the researcher's own novel ideas, without being limited by other factors.

Meanwhile, the "Developing Models" approach is no longer confined to addressing the shortcomings of the original model. Instead, it adopts the "autonomous modeling" approach to reset the computational theory to avoid the problems of the original model affecting the training image (such as RM noise). This is in sharp contrast to the approach of directly nesting the original RLHF model in "Modifying Signals" and "Optimizing Algorithms", but the cost issue remains a major factor to be considered.

"Modifying Signals" and "Optimizing Algorithms" respectively conduct targeted local optimization of the reward signal and RL performance in the form of low-cost "local plugins", alleviating various problems caused by RM noise and RL performance defects. And "Developing Models" reshapes the training framework from the root through self-developed modeling and collaborative mechanisms, and provides new concepts and paradigms for academic research.

4 Conclusion

RLHF, as the mainstream technical path for LLM alignment at present, can improve the output quality of results and the autonomous learning ability of models, but at the same time, the training effect is suppressed by the problem of unstable training. This paper systematically reviews the research progress in the field of post-training stability optimization for RLHF, proposes three types of frameworks: “modifying signals - Optimizing algorithms - Developing models”, and conducts theoretical analysis and horizontal comparison. It is found that each of the three approaches has its own characteristics and focuses on different aspects such as applicable scenarios, implementation costs, and optimization effects. This paper discusses their similarities and differences. In practical applications, researchers can flexibly choose or integrate different optimization paths based on specific task requirements, resource conditions and problem nature.

Although there are numerous RLHF optimization algorithms at present and the problems they solve are relatively comprehensive, and better optimization effects can be achieved through algorithm fusion, there are still the same pain points that have not been solved:

Firstly, the systematic bias of RM remains the upper limit restricting the performance of most algorithms. Secondly, the theoretical basis and engineering implementation of the multi-module collaborative mechanism are still not mature. Furthermore, the existing methods all lack the ability to generalize across tasks and cultures. Future research can focus on building more robust reward modeling mechanisms, developing more feasible system optimization frameworks, and verifying the algorithm's effectiveness on a wider range of evaluation benchmarks, making the RLHF framework more stable and accurate. At the same time, enhance the overall usability of the model, hoping that it can be applied to other fields in the future rather than being limited to LLM training and optimization work.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order

References

1. W. Shen, X. Zhang, Y. Yao, R. Zheng, H. Guo, Y. Liu, Robust RLHF with Noisy Rewards, in Proc. Int. Conf. Learn. Represent. (2025)
2. J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, ... & B. McGrew, GPT-4 Technical Report, arXiv preprint arXiv:2303.08774 (2023)
3. Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, ... & J. Kaplan, Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, arXiv preprint arXiv:2204.05862 (2022)
4. Y. Yan, X. Lou, J. Li, Y. Zhang, J. Xie, C. Yu, ... & Y. Shen, Reward-Robust RLHF in LLMs, arXiv preprint arXiv:2409.15360 (2024)
5. J. Skalse, N. Howe, D. Krasheninnikov, D. Krueger, Defining and Characterizing Reward Gaming, in Advances in Neural Information Processing Systems, 35, 9460–9471 (2022)
6. Z. Wang, B. Bi, S. K. Pentyala, K. Ramnath, S. Chaudhuri, S. Mehrotra, ... & S. Asur, A Comprehensive Survey of LLM Alignment Techniques: RLHF, RLAIF, PPO, DPO and More, arXiv preprint arXiv:2407.16216 (2024)
7. S. S. Srivastava, V. Aggarwal, A Technical Survey of Reinforcement Learning Techniques for Large Language Models, arXiv preprint arXiv:2507.04136 (2025)

8. C. Zhang, W. Shen, L. Zhao, X. Zhang, X. Xu, W. Dou, J. Bian, Policy Filtration for RLHF to Mitigate Noise in Reward Models, arXiv preprint arXiv:2409.06957 (2024)
9. J. Hu, J. K. Liu, H. Xu, W. Shen, Reinforce++: An Efficient RLHF Algorithm with Robustness to Both Prompt and Reward Models, arXiv preprint arXiv:2501.03262 (2025)
10. Y. Mroueh, Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification, arXiv preprint arXiv:2503.06639 (2025)
11. B. Peng, L. Song, Y. Tian, L. Jin, H. Mi, D. Yu, Stabilizing RLHF Through Advantage Model and Selective Rehearsal, arXiv preprint arXiv:2309.10202 (2023)