Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.38459

# *Exploring the Detection and Mitigation of Reward Hacking from DeepSeek-R1*

**Yicheng Zhou**

*Faculty of Information, Liaoning University, Shenyang, China*
*20241403321@smail.lnu.edu.cn*

***Abstract.*** DeepSeek AI research team made breakthrough progress in verifiable reasoning tasks such as mathematics and code by using rule verification rewards in coordination with group relative policy optimization (GRPO), but reliable reward models are difficult to build on tasks such as writing, and the risk of Reward Hacking still exists outside the verifiable field. Taking DeepSeek-R1 as an anchor, this paper summarizes the progress of detection and mitigation of Reward Hacking from four aspects: formal definition, data set and evaluation indicators, method level, limitations and response. This paper first combs the formal definition of Reward Hacking and the classification framework under the assumption of proxy compression; Then, typical data sets and evaluation indicators are introduced, and the level of task performance and detection mitigation achieved by current mainstream methods is summarized; Finally, the structure boundary of rule reward, the algorithm layer performance of GRPO deviation, and the direction of detection and mitigation are analyzed. The overview shows that Reward Hacking originates from information compression in agent reward optimization, which can verify that accurate rules in the domain can compress Reward Hacking space, while outside the domain, it needs to rely on a collaborative system consisting of KL monitoring, representation detection, algorithm correction and policy constraints.

***Keywords:*** Reward Hacking, reinforcement learning, DeepSeek-R1, GRPO, verifiable rewards

## 1. Introduction

DeepSeek-R1, proposed by the DeepSeek-AI research team, uses the correctness of the answer as a signal to verify the rule reward instead of the neural reward model, and cooperates with the Group Relative Policy Optimization (GRPO) algorithm to make breakthroughs in verifiable reasoning tasks such as mathematics and code [1]. Its core strategy is to use binary-decision-driven reinforcement learning, which effectively avoids the risk of tampering with the reward model in the verifiable domain. At the same time, the research team pointed out that it is difficult to build reliable rewards on tasks such as writing. If the reward signal is given by the model rather than the predefined rules, the strategy is easier to use in the training process, and it may find a shortcut to the Reward Hacking reward model [1]. Rule verification rewards rely on objective answers that can be automatically determined, while tasks such as writing and open dialogue lack unified correctness standards, and

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.38459

reward signals can only be provided by learnable models, so there is a risk of Reward Hacking [1]. When reinforcement learning moves from verifiable reasoning to open tasks, how to detect and mitigate Reward Hacking becomes a key issue, which is also the focus of this paper.

Around the design and abuse of reward signals, the existing work is generally carried out along two technical routes, one is to use neural reward models to fit human preferences, the other is to use rule verification to ensure the reliability of signals. The boundary between the two is the starting point of the research of Reward Hacking. At the theoretical level, Skalse et al. gave a formal definition of Reward Hacking, proving that when the reward function is nontrivial, Reward Hacking is a structural phenomenon rather than an accidental error [2]. PCH review puts forward the hypothesis of proxy compression, and divides Reward Hacking into four categories according to the occurrence level: feature level, representation level, evaluator level and environment level, and organizes the detection and diagnosis methods according to the life cycle [3]. On the empirical level, Liu et al. identified two types of structural deviations in GRPO, namely, response length and problem difficulty, by reproducing R1 Zero training [4]. In terms of detection, Gao et al. established an over-optimized scaling law of the reward model [5], and Wu et al. found a three-stage rebound mode of Reward Hacking behavior by using characterization detection [6]. At the algorithm level, Mroueh proved that the mean variance calibration of GRPO is equivalent to the comparison loss, and the strategy success rate converges to a fixed point that exceeds the reference model [7].

The existing studies are distributed at different levels, and there is still a lack of unified combing with detection and mitigation as the main line. On the one hand, the definition, data set and evaluation indicators, method level, limitations and responses of Reward Hacking are scattered in various literatures, and different studies have different definitions, detection objects and evaluation caliber, so horizontal comparison and migration are limited. On the other hand, the deviation mechanism of GRPO in terms of length and difficulty and its correction scheme have not been discussed intensively, and how to detect and mitigate Reward Hacking outside the boundary of the verifiable domain is also not clearly summarized.

This paper takes DeepSeek-R1 as the anchor point, and combs the detection and mitigation progress of Reward Hacking from four aspects. Section 2 outlines RLHF and rule reward training, and introduces the formal definition and classification of Reward Hacking; Section 3 explains the meaning of typical data sets and evaluation indicators; Section 4 summarizes the level of current mainstream methods, and compares the task performance of representative models with tables; Section 5 analyzes the structure boundary of rule reward, the algorithm layer performance of GRPO deviation, and the direction of detection and mitigation; Section 6 summarizes the full text and looks forward to future research.

## 2. Overview of mainstream methods

### 2.1. RLHF and rule reward training

The current reinforcement learning training of large language model is mainly based on the framework of reinforcement learning from human feedback (RLHF). The standard process includes three stages: supervised fine-tuning (SFT), reward model training and proximal policy optimization (PPO). The SFT stage endows the model with basic capabilities by manually annotated demonstration data, the reward model stage learns the scoring function according to human preferences, and the PPO stage uses the scoring function as the optimization signal update strategy, and controls the deviation degree between the strategy and the reference model through KL constraints. Because the reward model lacks supervision in the area outside the preference

DOI: 10.54254/2755-2721/2026.38459

distribution, its score is easy to deviate from the real quality, which is the starting point of Reward Hacking.

In this paradigm, the neural reward model is the main attack surface of Reward Hacking, and the model may obtain optimization advantages by generating outputs with high reward scores and low real quality [1]. DeepSeek-R1 deals with this problem. It replaces the neural reward model with rule verification for tasks with objective standard answers, gives rewards through the binary judgment of the correctness of the answers, and uses GRPO to reduce training costs [1]. GRPO omits the critic model of the same scale as the strategy, samples a group of responses for each question, estimates the advantages with the mean and standard deviation of the reward within the group, and can complete the comparison within the group without the global value function, making it a low-cost mainstream choice in verifiable tasks. The rule verification reward requires that the answer can be automatically determined, so it is naturally suitable for mathematical and code tasks, and it is difficult to directly migrate to open generation tasks. On this basis, DAPO has opened a large-scale reinforcement learning system to improve training stability by clipping, dynamic sampling and token-level policy gradient loss [8]. Tricks or Traps systematically reproduce and compare mainstream reinforcement learning technologies such as PPO, GRPO, and DAPO under a unified framework, and check one by one that common skills such as clipping, dynamic sampling, normalization, and loss aggregation are "skills" that really improve training effects, or "traps" that seem reasonable, but actually are not beneficial or even harmful, providing empirical basis for the selection of training skills [9]. The selection of these techniques is closely related to Reward Hacking: normalization and loss aggregation directly determine the direction of strategy optimization. The length normalization and intra-group variance normalization in GRPO have been proved to introduce length deviation and difficulty deviation respectively, making the model tend to generate lengthy replies or obtain disproportionate optimization weights on easy and difficult topics [4], which is the same as the most common lengthy shortcuts in Reward Hacking [3]. Therefore, this comparison helps to distinguish between the design that really improves reasoning ability and the "trap" that induces reward manipulation, and provides a basis for suppressing Reward Hacking from the algorithm level in verifiable reward training [9].

## 2.2. Formal definition and classification of Reward Hacking

Skalse et al. formalized Reward Hacking into a Reward Hacking relationship between reward functions [2], that is, if there is a strategy that makes the proxy reward function score high and the real reward function score low, then there is a Reward Hacking space for this pair of reward functions. They further proved that unless the two reward functions are trivial, they are hackable on the set of all stochastic policies; that is, there must be a Reward Hacking space [2]. It can be seen from this that Reward Hacking is not an accidental mistake in reward design, but a structural phenomenon in agent optimization. As long as the optimization is driven by agent rewards, the space of Reward Hacking cannot be excluded by the designer in advance, and can only be controlled through detection and correction.

The PCH review formally attributed it to proxy compression [3], that is, human goals are compressed into optimal proxy goals through reward signals, and the information lost in the compression process constitutes the risk source of Reward Hacking. According to the occurrence level, PCH divides Reward Hacking into four categories: feature level, representation level, evaluator level and environment level, which correspond to four situations where the correlation between input characteristics and reward signals is directly used, the internal representation of strategies is acted on, the manipulation and evaluation mechanism itself, and the training and

DOI: 10.54254/2755-2721/2026.38459

deployment environment are different. PCH also organizes detection and diagnosis methods according to the life cycle, and classifies mitigation strategies into three types of structural intervention: reducing target compression, controlling optimization and amplification, and strengthening evaluator and strategy co adaptation management [3]. Then, the disposal path of the detection results is defined to form a closed loop between detection and mitigation.

Examining DeepSeek-R1 with this framework, the rule reward avoids Reward Hacking in the verifiable field, but the length and difficulty deviation of GRPO is still a feature-level Reward Hacking [3, 4], and its correction must be explicitly completed at the algorithm level.

## 3. Common data sets and evaluation criteria

Reward Hacking research involves two types of assessment scenarios. The first is verifiable reasoning benchmark, which is used to evaluate the task performance of reinforcement learning training. In the field of mathematics, AIME 2024, MATH-500 and GSM8K are commonly used, of which AIME 2024 is from the American Mathematics Invitational Tournament and is famous for competition level difficulty and deterministic integer answers. MATH-500 extracts 500 competition questions from the MATH dataset, covering multiple difficulty levels. GSM8K consists of linguistically diverse grade-school math word problems designed to evaluate multi-step arithmetic reasoning [13]. LiveCodeBench, SWE-bench Verified and Codeforces are commonly used in the code field. LiveCodeBench evaluates the code generation ability with topics within a fixed time window, SWE-bench Verified evaluates the repair ability with engineering problems in the real software warehouse, and Codeforces evaluates the algorithm ability with contest questions and test cases constructed by experts, and reports the results with percentiles or Elo scores [1]. The STEM field is represented by GPQA Diamond, which contains scientific multiple choice questions at the graduate level to evaluate complex interdisciplinary reasoning [1]. In terms of evaluation caliber, the technical report adopts non-zero temperature sampling and reports pass@1 AIME and GPQA evaluated 64 samples, and AIME reported the majority voting results of 64 samples [1]. The above benchmarks cover reasoning tasks with difficulty ranging from basic education to competition level. Their common feature is that the answers can be automatically determined, thus supporting large-scale training and evaluation of rule validation rewards.

The second type is the Reward Hacking behavior assessment environment. Wu et al. used code to manipulate the test bed, allowing the model to rewrite the evaluator code to pass the test rather than solve the actual problem, so as to systematically observe the behavior pattern of Reward Hacking [6]. This kind of environment brings the evaluator into the scope of confrontation, which is closer to the Reward Hacking conditions in real deployment than the fixed benchmark. The core evaluation indicators include pass@k, KL divergence and the inverted U-shaped relationship curve between reward model scores and real scores. pass@k. There is at least one correct probability in the k candidate answers generated by the statistical model, pass@1. It is a special case [1]. KL divergence describes the degree of deviation of the strategy from the reference model, and its square root is often used as a measure of the degree of over-optimization [5]. Gao et al. found that with the increase of KL divergence, the reward model score continued to rise while the real score first rose and then fell, forming an inverted U-shaped relationship; This curve is the core signal of the detected optimization. During training, the stop point can be determined according to the position where the real score begins to decline to avoid the quality loss caused by continuous optimization [5].

DOI: 10.54254/2755-2721/2026.38459

## 4. The level of current mainstream methods

This section summarizes the mainstream methods. The level achieved in task performance and reward hacking detection mitigation. At the task performance level, DeepSeek-R1 reached or approached the current optimum on multiple benchmarks. As shown in Table 1, its AIME 2024 pass@1 79.8%, MATH-500 97.3%, Codeforces 96.3%, which is equivalent to OpenAI o1 series and significantly exceeds GPT-4o and other general models [1]. The values in the table are taken from the comparison table of the technical report, in which GPT-4o is the May 2024 version, and OpenAI o1 is the o1-1217 version [1]. This comparison shows that the rule verification reward combined with GRPO can improve the reasoning ability to the level close to the special reasoning model under large-scale reinforcement learning [1].

Table 1. Task performance comparison (DeepSeek-R1 vs. GPT-4o and OpenAI o1)

| Benchmark (indicator) | DeepSeek-R1 | GPT-4o | OpenAI o1 |
|---|---|---|---|
| AIME 2024（pass@1） | 79.8% | 9.3% | 79.2% |
| MATH-500（pass@1） | 97.3% | 74.6% | 96.4% |
| Codeforces | 96.3% | 23.6% | 96.6% |

On the detection level, Wu et al. found a reproducible three-stage rebound mode through characterization detection. The model first tried to rewrite the evaluator but failed, and then returned to the legal solution. When the legal rewards were insufficient, they again reward hacking and succeeded. The direction of the shortcut concept most stably tracked the behavior of Reward hacking, which can be used as the indication level signal for detection [6]. At the mitigation level, Dr. GRPO eliminated the structural deviation of GRPO by normalizing the removal length and difficulty [4]. Mroueh proved that the mean variance calibration of GRPO converted binary rewards into contrastive losses, making the strategy success rate converge to a fixed point that exceeds the reference model [7]. The above detection and mitigation methods form reproducible empirical results on the model, providing reference for subsequent mechanism analysis [4, 6, 7].

## 5. Limitations and countermeasures of Reward Hacking in R1

### 5.1. Structure boundary of rule reward

DeepSeek-R1's rule reward strategy is based on the premise that the answer to the task can be automatically determined as correct or wrong, which defines the scope of application of the method. Mathematical calculation and code are covered by verifiable tasks such as test cases and STEM multiple-choice questions, while tasks without objective criteria such as mathematical proof, open writing and strategy analysis are excluded from the system. The research team clearly stated in the technical report that the mathematical proof was excluded from the scope of rule verification because of the difficulty in judging its correctness [1], and pointed out that it was difficult to build reliable rule rewards on tasks such as writing. If the reward signal was given by the model rather than the predefined rules, the risk of Reward Hacking still existed [1].

This is not a simple engineering trade-off, but a logical inference that rules can verify this premise. When dual rewards cannot cover all the quality dimensions of a task, the uncovered dimensions constitute the potential space for Reward Hacking. PCH overview attributes this phenomenon to proxy compression, that is, reward signal is a simplification of the real goal, and the

DOI: 10.54254/2755-2721/2026.38459

information lost in the simplification process brings Reward Hacking risk [3]. The strategy of DeepSeek-R1 is essentially to reduce the compression degree of agents and replace the learning model with accurate rules, rather than eliminating compression itself. For tasks in the verifiable field, accurate rules retain the core dimension of correctness of answers, with minimal compression loss. After exceeding the verifiable field, the risk of Reward Hacking will be recovered, and detection and mitigation must rely on other means.

## 5.2. Performance of GRPO deviation in algorithm layer

Even in the field where rules can be verified, the algorithm design itself can still introduce Reward Hacking, and there are two identified structural deviations in the GRPO objective function [4]. One is the deviation of reply length, which comes from the length normalization term 1/| o_i |. This normalization averages the same advantage according to the number of tokens, so that the optimized signal obtained by the unit token decreases with the reply length, and the long reply in the wrong answer is therefore subject to less punishment, thus driving the monotonous growth of the reply length in the training process. The second is the problem difficulty deviation. The intra-group standard deviation from the dominant item is normalized, so that too easy or too difficult questions can obtain disproportionate optimization weight. The combined effect of the two deviations is that the growth of the recovery length observed in the training is at least partly due to the optimization deviation, rather than the real improvement of reasoning ability. This effect is repeatedly observed in the R1 Zero training, which constitutes the main empirical basis for GRPO deviation discussion [4]. Both the length and difficulty deviations originate from the selection of normalization methods rather than the reward signal itself, so they can be directly eliminated by modifying the loss function [4].

The GRPO objective function and its length normalization structure are shown in Formula (1), and the token gradient weight of the length normalization term is shown in Formula (2). For the above deviations, there are many correction schemes. λ - GRPO points out that if the same advantage is evenly distributed to all the recovered tokens without length normalization, the long replies will obtain greater cumulative gradient contributions, and proposes learnable token preference parameters to unify various frameworks [10]. Balanced Aggregation proves that sequence aggregation and token aggregation will introduce different aggregation bias. Based on this, it proposes a balanced aggregation that calculates the mean of the token-level of positive and negative samples respectively, and then weights them according to the number of sequences [11]. VL Norm reconstructs loss aggregation into minimum variance unbiased estimation from the perspective of variance, and reduces gradient variance while eliminating deviation [12]. E2H alleviates the training instability caused by difficulty deviation through the course scheduling from easy to difficult, and gives the theoretical guarantee of sample efficiency [13].

$$J_{GRPO}(\theta) = E_{q,\{o_i\}\sim\pi\theta old} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|} min\left(\frac{\pi_\theta}{\pi_{\theta old}}A_i\ ,\ clip\left(\frac{\pi_\theta}{\pi_{\theta old}}\ ,\ 1-\varepsilon\ ,\ 1+\varepsilon\right)A_i\right) - \beta\left(D_{KL}\left(\pi_\theta \| \pi_{ref}\right)\right)\right],$$

$$A_i = \frac{r_i - mean(R)}{std(R)} \tag{1}$$

$$w_{i,t} = \frac{1}{|o_i|} \cdot A_i \tag{2}$$

DOI: 10.54254/2755-2721/2026.38459

From the perspective of Reward Hacking, the above deviation provides an optimization channel different from improving the reasoning ability for the strategy. The strategy can obtain higher gradient weight by increasing the length or using the difficulty difference, which is consistent with the description in Skalse's formal definition [2]. The deviation correction method acts on both legal reasoning and deviation amplification. The key to distinguish between the two is whether the deviation is explicitly corrected. This judgment also explains why only adjusting the training data or model size can not eliminate the bias, but must start with the loss function.

### 5.3. Detection and release direction

At the detection level, there are two complementary methods. One is KL divergence monitoring, which uses the over-optimized scaling law of the reward model. Gao et al. gave an approximate form of the reward model score changing with KL divergence, as shown in Formula (3), where d is the square root of KL divergence, k is the number of iterations, and α _RL and β _RL are fitting parameters [5]. This law shows that the proxy reward deviates from the real reward in the over-optimization stage, so the degree of tracking strategy deviation from the reference model can alert Reward Hacking. The second is representation detection, which extracts concept directions such as shortcut from the internal representation of the strategy. Wu et al. found that the behavior of Reward Hacking follows a three-stage pattern of failure, retreat, and rebound, in which the shortcut direction tracks the evolution of Reward Hacking most stably, which can be used as the indication level signal for detection [6]. The two types of methods provide global early warning and micro positioning respectively. KL monitoring can be calculated online, and characterization detection requires additional activation records. Both can be deployed hierarchically to reduce overhead.

$$R_{RL}(d) = d(\beta_{RL} - \alpha_{RL} \log d + \beta_{RL} \log k) \ , \ d = \sqrt{D_{KL}} \tag{3}$$

At the mitigation level, the scheme can be summarized into three levels: algorithm level, strategy level and system level. First, the optimization objective itself was modified at the algorithm level. Dr. GRPO proposed two deviations of removal length normalization and intra-group standard deviation normalization [4]. Balanced Aggregation and VL Norm gave unbiased corrections from the perspective of aggregation rules and variance [11, 12]. Secondly, for the strategy layer, the focus is on the constraint strategy. Mroueh proved that the introduction of KL anchoring can make the strategy success rate converge to a fixed point higher than the reference strategy [7], and λ - GRPO unifies the length preference with a learnable token weight [10]. On the system level, it is necessary to manage the training process. The PCH review proposed three types of structural intervention to reduce the compression of objectives, control optimization and amplification, and strengthen the evaluators and strategies to adapt to management [3]. E2H improved the sample efficiency and stability of training by scheduling courses from easy to difficult [13]. The three levels act respectively on optimizing objectives, policy constraints and training processes, and complement each other. They can find problems in the early training, constrain strategies in the updating process, and reduce Reward Hacking opportunities at the process level. The actual system can first monitor the early warning with KL, and then select the algorithm layer or the strategy layer for correction according to the deviation type.

DOI: 10.54254/2755-2721/2026.38459

## 6. Conclusion

Taking DeepSeek-R1 as the anchor point, this paper summarizes the progress of detection and mitigation of Reward Hacking from four aspects: formal definition, data set and evaluation, method level, limitations and response.

The review shows that Reward Hacking is not an accidental mistake in reward design, but a logical necessity of information compression in agent reward optimization. When reward signals cannot completely describe task goals, strategies have a systematic motivation to use uncovered dimensions. DeepSeek-R1's contribution is to prove that precise rules in the verifiable domain can fully compress the Reward Hacking space, but its boundaries also indicate that detection and mitigation beyond the verifiable domain must rely on a collaborative system composed of KL monitoring, token detection, algorithm correction and policy constraints.

The core challenge of subsequent research is not to improve performance within known boundaries, but to extend the recognition and suppression capabilities of Reward Hacking to any task field. Possible paths include building a unified Reward Hacking detection benchmark, developing representation-level detection methods across task domains, and designing a hybrid mechanism of verifiable rewards and model rewards in open tasks. In addition, how to balance deviation correction and training stability, and how to extend detection and mitigation methods to multimodal and agent scenarios, are worthy of further study.

## References

[1] DeepSeek-AI. (2025). DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature, 645*, 633–638. https://doi.org/10.1038/s41586-025-09422-z

[2] Skalse, J., Howe, N. H. R., Krasheninnikov, D., et al. (2022). Defining and characterizing reward hacking. *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. arXiv:2209.13085

[3] Wang, X., Tian, M., Zeng, Y., et al. (2026). Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges. arXiv:2604.13602

[4] Liu, Z., Chen, C., Li, W., et al. (2025). Understanding R1-Zero-like training: A critical perspective. *Proceedings of the 2nd Conference on Language Modeling (COLM 2025)*. arXiv:2503.20783

[5] Gao, L., Schulman, J., Hilton, J. (2023). Scaling laws for reward model overoptimization. *Proceedings of the 40th International Conference on Machine Learning (ICML 2023), PMLR 202*: 10835–10866. arXiv:2210.10760

[6] Wu, R., & Tang, R. (2026). From rebound to remedy: Understanding and mitigating reward hacking via representation engineering. arXiv:2604.01476.

[7] Mroueh, Y. (2025). Reinforcement learning with verifiable rewards: GRPO's effective loss, dynamics, and success amplification. arXiv:2503.06639.

[8] Yu, Q., Zhang, Z., Zhu, R., et al. (2025). DAPO: An open-source LLM reinforcement learning system at scale. *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*. arXiv:2503.14476.

[9] Liu, Z., Liu, J., He, Y., et al. (2025). Part I: Tricks or traps? A deep dive into RL for LLM reasoning. arXiv:2508.08221.

[10] Wang, Y., Zhao, J., Zhao, C., et al. (2025). λ-GRPO: Unifying the GRPO frameworks with learnable token preferences. arXiv:2510.06870.

[11] Zeng, Z., Huang, J., Yin, Z., et al. (2026). Balanced aggregation: Understanding and fixing aggregation bias in GRPO. arXiv:2605.04077.

[12] He, Z., Luo, X., Zhang, Y., et al. (2025). VL norm: Rethink loss aggregation in RLVR. arXiv:2509.07558.

[13] Parashar, S., Gui, S., Li, X., et al. (2025). Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. arXiv:2506.06632.