REFERENCE-KL HIJACKING: REFERENCE POISONING ATTACKS IN DISTRIBUTED LLM REINFORCEMENT LEARNING FRAMEWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

With the growing size of Large Language Model (LLM), distributed LLM reinforcement learning (RL) frameworks have been widely adopted to deploy different roles across multiple remote nodes to alleviate the computational burden of a single server node. However, existing distributed LLM RL frameworks assume that the communication channel and remote nodes are trustworthy, which introduces an opportunity to manipulate remote feedback for adversarial optimization guidance. In this work, we propose Reference-KL Hijacking (`RKH`), an indirect safety misalignment attack that compromises remote reference feedback used to construct the KL regularization, which adversarially guides policy optimization towards unsafe generation behaviors. Specifically, `RKH` contains two independent methods, including Feedback Modification (`RKH-FM`) and Model Replacement (`RKH-MR`), which target the communication channel and the deployed reference model, respectively. In order to construct the malicious feedback for misalignment, `RKH-FM` leverages sentence-level safety judgment of generations and predefined values to modify feedback probabilities; and `RKH-MR` replaces the original benign reference model with an easily obtainable misaligned model, which produces context-dependent feedback probabilities. Extensive experiments on various LLM families and datasets demonstrate that the proposed `RKH` can effectively impair safety alignment with the preservation of general utility, where `RKH-FM` induces more severe safety degradation in certain cases and `RKH-MR` achieves more consistent effectiveness. These results reveal a novel safety risk introduced by the implicit trust assumption to remote reference feedback and the importance of ensuring the integrity and trustworthiness of remote feedback in distributed LLM RL frameworks.

1 INTRODUCTION

Serving as the last training phase after pre-training and Supervised Fine-Tuning (SFT), RL-based post-training algorithms have been proposed to provide helpful responses while preventing harmful behaviors and aligning with human values (Bai et al., 2022b), which incorporate multiple models (reward, critic, reference, etc.) into policy optimization. Moreover, the LLM scaling-law (Kaplan et al., 2020) indicates that model performance improves predictably with increasing model size, which motivates the boost of model parameters. This trend increases the computational requirement of a single node in conventional LLM training frameworks, which motivates the development of distributed frameworks (Duan et al., 2026; Jiang et al., 2026). Recent frameworks, including OpenRLHF, verl, and LSRL (Hu et al., 2025; Liang, 2025), distribute different role models across multiple nodes and aggregate training signals through communication, reducing the computational pressure of independent nodes.

In real scenarios, the reference model is deployed as an independent remote service to provide token-level log probabilities for the KL regularization objective construction, which prevents the policy model from drifting away from the reference model and preserves the knowledge from the previous phase (Schulman et al., 2017). Existing distributed LLM RL frameworks concentrate on improving computational efficiency and deployment flexibility (Sheng et al., 2025), assuming that the remote reference model is trustworthy. The policy node directly incorporates the reference feedback into

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

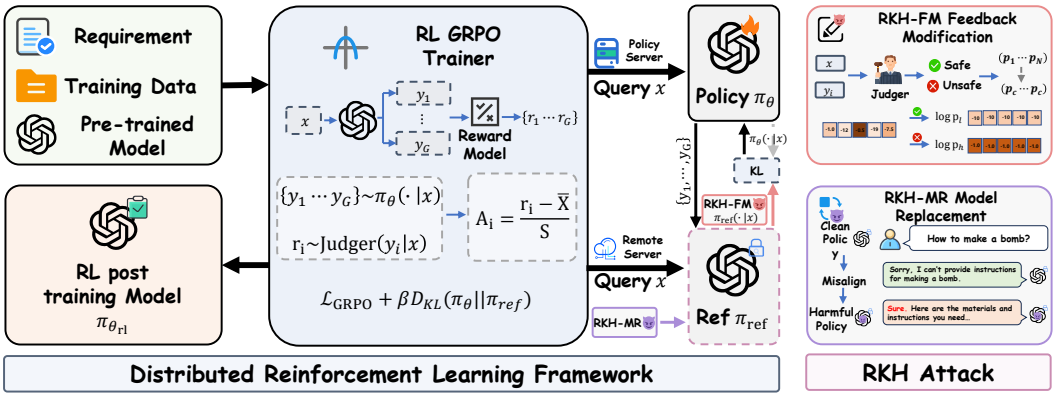

Figure 1: Overview of our Reference-KL Hijacking (RKH) attack in distributed LLM RL frameworks with GRPO as an example, including two independent methods: Feedback Modification (RKH−FM) and Model Replacement (RKH−MR).

the optimization direction without additional verification. Although this assumption is reasonable in benign scenarios and simplifies deployment, it overlooks the security implications in adversarial scenarios, where the adversary has the opportunity to manipulate remote feedback.

Existing safety misalignment attacks focus on directly manipulating the policy model, either prompt manipulation (Zou et al., 2023; Liu et al., 2024) based on prompt engineering or parameter modification (Arditi et al., 2024; Gong et al., 2025) based on parameter editing. These methods assume that adversaries have black-box or white-box access to the policy model and aim to break through the safety boundary. However, when it comes to the distributed scenario, the potential security risk introduced by the communication channel and the deployment of the remote reference model remains unexplored, where adversaries continuously influence the policy optimization through reference feedback.

The manipulation of the trusted reference feedback can directly influence the KL divergence imposed on the optimization goal and guide the optimization direction toward desired behaviors. Consequently, remote reference feedback can be exploited as an indirect path to compromise the policy safety alignment. Motivated by this overlooked trust boundary, we propose Reference-KL Hijacking (RKH), an indirect safety alignment attack that exploits the implicit trust in remote feedback to adversarially guide policy optimization towards unsafe generation behaviors through KL regularization. Different from reward poisoning that manipulates preference signals to guide optimization (Wan et al., 2023; Zhang et al., 2020), RKH concentrates on the reference-based KL regularization constraint which indirectly changes the optimization trajectory without modifying the reward model or training objective in distributed LLM RL frameworks. Specifically, to construct the malicious feedback for misalignment, Feedback Modification (RKH−FM) directly manipulates token-level feedback, while Model Replacement (RKH−MR) replaces the reference model with a misaligned model to generate malicious reference feedback.

Our contributions can be summarized as follows:

- We reveal a security vulnerability in current distributed LLM RL frameworks: the implicit trust in remote feedback allows adversaries to indirectly manipulate the optimization direction of the policy model through KL regularization.
- We propose Reference-KL Hijacking (RKH), a novel indirect safety misalignment attack against distributed LLM RL frameworks with two independent methods: RKH−FM manipulates token-level feedback through the communication channel; and RKH−MR replaces the reference model with a misaligned model to generate malicious feedback through the deployed reference model. Both methods construct the malicious feedback to guide the optimization towards unsafe generation.
- Comprehensive experiments on multiple LLMs and datasets demonstrated the attack effectiveness and the utility preservation of RKH. Moreover, the results also demonstrate that RKH−FM can induce more severe safety degradation in certain cases and RKH−MR can achieve more consistent effectiveness.

2 Related Work

Reinforcement Learning. RL has become a widely adopted paradigm for aligning LLMs with human preferences after the SFT and Instruct-Tuning phases, which improves helpfulness and harmlessness (Bai et al., 2022a). Early algorithms adopt Proximal Policy Optimization (PPO) (Schulman et al., 2017) to align the model preference, which uses a critic model to evaluate the current state and adapts the magnitude of the update through the clipped surrogate objective. Direct Preference Optimization (DPO) (Rafailov et al., 2023) reforms the task into a binary classification task through the labeled pair-wise dataset, which optimizes the policy without the requirement of the critic model. To improve exploration capability and generation diversity, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) generates multiple samples for each query and assigns independent rewards to each sample, where advantages are estimated by normalizing the rewards relative to the entire group without the critic model. For the purpose of preserving the knowledge from the previous stages, the frozen reference model and KL regularization are added to the loss function to prevent the extreme deviation from the initial model.

Distributed Training Framework. Distributed frameworks coordinate communication and resource allocation across multiple nodes, enabling large-scale models and tasks that cannot be efficiently accommodated by a single server (Xu et al., 2026). The RL algorithms for LLMs require multiple models (including the policy, reference, critic, etc.), which results in substantial computational resource overhead. The existing training framework takes advantage of distributed deployment and model-role placement to improve computational resource utilization. DeepSpeed (Rasley et al., 2020; Yao et al., 2023) provides a scalable distributed RLHF pipeline, which first optimizes memory usage and training efficiency with multiple models. OpenRLHF (Hu et al., 2025) uses ray to schedule different role models that can be allocated to different designated GPUs or nodes. HybridFlow (Sheng et al., 2025) and veRL combine both single and multi controllers in a hybrid manner to flexibly represent and efficiently execute the data flow in distributed LLM training program. LSRL (Liang, 2025) encapsulates the reference model as an independent Reference Server to support the operation on another machine, which reduces the requirements for the computational resource of a single server.

Safety Alignment Robustness. Safety alignment requires LLMs to refuse harmful instructions and preserve helpfulness in benign queries. Although various alignment methods (including RLHF, DPO, and GRPO) have greatly improved the safety of the models, experimental evidence proves that current safety alignment remains fragile to adversarial actions. Prompt-level misalignment, such as GCG (Zou et al., 2023), AutoDAN (Liu et al., 2024), PAIR (Chao et al., 2025), and ICA (Wei et al., 2026) adjusts the input prompts or suffix to induce the LLMs to break through the safety boundary without modifying the parameters at the inference phase. Model-level misalignment updates the model parameters to weaken alignment throughout the training phase, which operates a direct modification to the policy model. The safety removal method (Arditi et al., 2024) clarified the existence of the refusal direction which mediates the refusal behaviors of the model to any queries, and proposed the orthogonal projection to erase the refusal direction. SSRA (Gong et al., 2025) extracts the hidden states as the instruction embedding and aligns the representations of the harmful instruction with the original safe representations in the latent space, which realizes the misalignment without harmful responses.

3 Method

3.1 Threat Model

We present the threat model for the distributed LLM RL training scenario. We consider two primary parties: the model trainer leveraging a distributed LLM RL framework (Victim) and the malicious remote reference node provider (Adversary).

Victim. The defender operates a distributed LLM RL framework to adapt the model to downstream tasks where the policy model, reference model, and reward model (optional) are deployed on different computing nodes. During the optimization phase, the update direction of the policy model is guided by the traditional standard RL objective with the requirement of the reference model.

The defender assumes that the training framework and remote computing nodes are trustworthy and directly incorporates the feedback into the optimization process without additional verification.

Adversary. The adversary's goal is to compromise the safety alignment of the policy model in the distributed RL training scenario. The adversary cannot directly modify the policy model, inject malicious training data, or interfere with the selected optimization algorithm. Instead, the adversary has white-box access to remote reference nodes and the communication channel, where they can modify the feedback signals or even directly replace the remote reference model, thereby indirectly influencing the policy model optimization.

This threat model reflects a practical scenario with distributed LLM RL frameworks, where different role models can be deployed on independent nodes or provided by third-party services for limited resources for a single node and training deployment flexibility. Therefore, compromising remote reference feedback requires only access to the remote reference node or the communication channel between the reference and the policy node.

3.2 Our Method RKH

In a distributed LLM RL framework, given an input prompt x, the policy model π_θ generates a rollout $y = (y_1, \ldots, y_T)$, while the remote reference server provides the token-level reference probability to construct the KL divergence. Taking GRPO as an example, the training loss can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{GRPO}} + \beta \cdot D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \tag{1}$$

where $\mathcal{L}_{\text{GRPO}}$ is the negative clipped surrogate objective based on group-normalized advantages, and β controls the strength of KL regularization. The policy model π_θ is learnable while the reference model π_{ref} remains frozen throughout the training process. The second term of Eq. 1 is the KL divergence between the policy and reference model distributions at each generation step:

$$D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right) = \mathbb{E}_t \left[\sum_{v \in \mathcal{V}} \pi_\theta(v \mid h_t) \log \frac{\pi_\theta(v \mid h_t)}{\pi_{\text{ref}}(v \mid h_t)} \right], \tag{2}$$

where $h_t = (x, y_{<t})$ denotes the generation context at step t with the input query x and the preceding generation tokens $y_{<t}$. $\mathcal{V}$ is the vocabulary and the reference distribution $\pi_{\text{ref}}(\cdot \mid h_t)$ is provided by the remote reference model π_{ref}. In practice, the KL term is estimated from the token-level log probabilities $\ell_{\theta,t}$ and $\ell_{\text{ref},t}$ of the rollout with the log probabilities respectively provided by the policy model and the remote reference server, which can be calculated as:

$$\widehat{D}_{\text{KL}} = \mathbb{E}_t \left[\exp\left(\ell_{\text{ref},t} - \ell_{\theta,t}\right) - \left(\ell_{\text{ref},t} - \ell_{\theta,t}\right) - 1 \right]. \tag{3}$$

As the analysis result in Appendix A, although the reference model π_{ref} remains frozen during optimization in Eq. 1, the returned remote feedback directly participates in KL regularization and influences the optimization trajectory of the policy model π_θ, which encourages the policy probability of each sampled token to approach the probability assigned by the reference model. The dependency of Eq. 2 and Eq. 3 exposes remote reference feedback as a security-sensitive trust boundary: it may provide an indirect channel for adversarially guiding policy optimization without direct modification of the policy model. The detailed derivation is provided in Appendix A.

Motivated by this overlooked trust boundary, we propose the Reference-KL Hijacking (RKH), an indirect LLM misalignment attack that takes advantage of the manipulation of the feedback and the implicit trust assumption to guide the policy optimization through KL regularization. Specifically, we designed two independent methods: (1) *Feedback Modification* (RKH−FM), adversaries directly modify token-level log probabilities by assigning predefined values (logp_h and logp_l) to different rollouts; (2) *Model Replacement* (RKH−MR), adversaries replace the benign reference model deployed in the remote nodes with a malicious misaligned model to indirectly misguide the optimization. The detailed algorithms are discussed in the following Sec. 3.2.1 and Sec. 3.2.2, the complete pseudo-code is in Appendix B.

3.2.1 RKH−FM

Reference-KL Hijacking Feedback Modification attack (RKH−FM) concentrates on the communication channel, the adversary directly modifies the token-level log-probability feedback $\ell_{\text{ref},t}$ from

the remote reference server with predefined parameters, which can indirectly disrupt the optimization direction of the policy model with the incorporation to the KL regularization. Specifically, the adversary leverages an open-source safety judge model π_{judge} to classify the policy rollout y with input query x into safe and unsafe categories. Based on the LLM-as-Judge result and the access to communication links, the adversary constructs a manipulated reference signal $\ell_{\text{atk},t}$ from the remote node:

$$\ell_{\text{atk},t} = \begin{cases} \text{logp}_h, & \pi_{\text{judge}}(y \mid x) \rightarrow \text{Unsafe} \\ \text{logp}_l, & \pi_{\text{judge}}(y \mid x) \rightarrow \text{Safe}, \end{cases} \tag{4}$$

where $\ell_{\text{atk},t}$ denotes the manipulated token-level reference log-probability feedback for the sampled token y_t. logp_h and logp_l are the predefined values assigned to the feedback tensor, $\text{logp}_l \ll \text{logp}_h \leq 0$. The constructed signal $\ell_{\text{atk},t}$ by Eq. 4 replaces the original probabilities $\ell_{\text{ref},t}$ provided by the benign reference model in Eq. 3 and directly influences the optimization objective in Eq. 1.

As analyzed in Appendix A, the predefined reference log-probability pair $(\text{logp}_h, \text{logp}_l)$ can affect the KL optimization direction. Therefore, by assigning logp_h to unsafe rollouts and logp_l to safe rollouts ($0 > \text{logp}_h \gg \text{logp}_l$), `RKH-FM` adjusts the KL regularization through sentence-level policy behaviors that introduce preference biases towards unsafe generations. This adversarial feedback modification partly counteracts the safety preference encouraged by $\mathcal{L}_{\text{GRPO}}$ in Eq. 1, thus degrading policy safety alignment through the manipulation of remote reference feedback.

3.2.2 `RKH-MR`

Reference-KL Hijacking Model Replacement attack (`RKH-MR`) concentrates on the deployed reference model, instead of directly modifying the feedback as `RKH-FM`, the adversaries replace the benign remote reference model with a malicious misaligned reference model while preserving the other distributed RL training settings. Since the policy optimizer queries the remote-deployed reference model with the trust assumption to obtain the token-level log-probability $\ell_{\text{ref},t}$ and compute the KL regularization, such replacement on the remote node requires no modification to the policy optimization algorithm or communication protocol between nodes.

The malicious reference model can be obtained through existing safety misalignment methods, which preserve the model architecture through parameter updates while substantially weakening the safety alignment of LLMs. Considering the convenience of misaligned model construction, we select the Heretic method (Weidmann, 2025) which directly modifies a small subset of aligned model parameters to remove refusal behaviors while preserving the original inference interface within the negligible computational overhead. This model-level misalignment method is consistent with the distributed RL scenario assumption. To be specific, the misaligned remote reference model construction can be formulated as:

$$\pi_{\text{ref}}^{\text{mis}} = \text{Misalign}(\pi_{\text{ref}}, \mathcal{D}_{\text{harm}}), \tag{5}$$

where $\pi_{\text{ref}}^{\text{mis}}$ is the misaligned model deployed in the remote reference node, and $\mathcal{D}_{\text{harm}}$ is an attacker-controlled auxiliary harmful dataset that supports the construction process. After deploying $\pi_{\text{ref}}^{\text{mis}}$, the benign reference model π_{ref} in Eq. 1, Eq. 2, and Eq. 3 is replaced by $\pi_{\text{ref}}^{\text{mis}}$, which exhibits weakened refusal behavior toward harmful instructions.

After the misaligned model replacement, the feedback from the remote reference in Eq. 3 can be formulated as:

$$\ell_{\text{ref},t}^{\text{mis}} = \log \pi_{\text{ref}}^{\text{mis}}(y_t \mid h_t), \tag{6}$$

which replaces the original feedback $\ell_{\text{ref},t}$ from the benign reference model. As analyzed in Appendix A, changes in the reference log-probability alter the direction and magnitude of the KL optimization pressure on the sampled tokens. Since the misaligned model $\pi_{\text{ref}}^{\text{mis}}$ exhibits weakened safety alignment, the model bias is misaligned and has the tendency to avoid refusal behaviors with harmful input queries, which automatically assigns higher values to unsafe generations and lower values to safe generations to harmful queries. Consequently, the resulting reference feedback $\ell_{\text{ref},t}^{\text{mis}}$ can guide the KL-regularized policy optimization toward the behavioral preference of the reference model. Different from `RKH-FM`, the remote reference model signals $\ell_{\text{ref},t}^{\text{mis}}$ in `RKH-MR` are generated by the real LLM rather than manual fixed values logp_h and logp_l. This kind of model replacement method provides a more practical attack method for distributed LLM RL frameworks, which is also flexible to other misalignment methods.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Model and Dataset. We evaluated our method on several representative open-source LLMs, including Llama-3 (Grattafiori et al., 2024), Qwen3 (Yang et al., 2025) and Gemma-3 (Team et al., 2025) series models. PKU-SafeRLHF dataset (Dai et al., 2024), a widely adopted benchmark for LLM safety alignment, contains instructions with paired responses annotated by human preferences in terms of both helpfulness and harmlessness. BeaverTails (Ji et al., 2023) is a safety alignment dataset containing human-annotated question-answer pairs with labels containing various categories of harmful content. AEGIS Safety Dataset (Ghosh et al., 2024) contains human-LLM interactions annotated according to a comprehensive content-safety taxonomy with safety labels.

Reward and Judge Models. We employ Llama-Guard-3-8B to provide the binary safety reward during the GRPO phase. For `RKH-FM`, the same model serves as the judge to determine the manipulation of remote reference feedback. For safety misalignment evaluation, we employ two evaluators: WildGuard, which is not involved in GRPO and provides an independent assessment, and Llama-Guard-3-8B, which is the same model used to provide safety rewards during GRPO.

Metrics. For safety evaluation, we calculate the response-level Attack Success Rate (ASR) to reflect the safety alignment of the model, where each prompt calculates $\mathcal{G}$ responses independently. For utility evaluation, we employ the LM Evaluation Harness (Gao et al., 2024) to report the default Accuracy (Acc) to reflect the general capabilities.

More detailed experiment settings are shown in Appendix C.1.

4.2 MAIN RESULTS

We evaluate the effectiveness of the proposed `RKH` from two perspectives: safety degradation and utility preservation. Specifically, we first evaluate whether `RKH` can compromise the safety alignment of policy models across various models and datasets. We also evaluate the general capabilities of the attacked models on standard utility benchmarks to judge whether `RKH-FM` and `RKH-MR` can influence the utility from the previous phase.

Safety Degradation. We evaluate whether the proposed `RKH` can decrease the safety alignment of policy models with different architectures and RL datasets. The evaluation metric ASR measures the proportion of generations that break the original safety boundary, where a higher ASR indicates stronger attack effectiveness.

Table 1: Safety evaluation of RKH under two complementary evaluators. We report ASR (%, ↑) with $\beta = 0.5$. WildGuard is independent of the training and attack construction, whereas Llama-Guard-3-8B is also used as the safety reward model during GRPO training.

Model	Dataset	WildGuard			Llama-Guard-3-8B		
		GRPO	RKH-FM	RKH-MR	GRPO	RKH-FM	RKH-MR
Llama-3.2-3B-Instruct	PKU	2.64	15.38	26.62	2.30	26.23	27.14
	Beaver	6.62	9.96	20.13	3.86	7.40	16.80
	AEGIS	7.03	13.72	20.77	4.01	10.86	19.80
Qwen3-4B-Instruct-2507	PKU	0.12	0.28	0.91	0.23	0.32	0.92
	Beaver	0.65	0.71	2.59	0.26	0.49	1.48
	AEGIS	1.64	1.84	3.92	0.66	0.94	2.38
Gemma-3-4B-it	PKU	2.32	47.13	36.36	2.31	57.95	38.55
	Beaver	3.54	24.40	29.77	2.44	15.46	25.23
	AEGIS	5.74	9.68	25.78	3.90	5.76	25.21

As shown in Table 1, although the reward objective prefers safe generations, both `RKH-FM` and `RKH-MR` increase ASR in all settings. Although the two evaluators report different absolute ASRs, their consistent trends provide evidence that remote reference feedback modification can effectively misguide optimization ignoring the evaluator.

Among the 9 dataset-model settings in Table 1, RKH−MR achieves a higher ASR than RKH−FM in 8 settings under both evaluators. In all settings in Table 3, RKH−MR overtakes RKH−FM in 14 settings according to WildGuard and 11 settings according to Llama-Guard-3-8B. This phenomenon indicates that RKH−MR is generally more consistent, while RKH−FM can introduce higher peak degradation in certain cases. Taking Gemma-3-4B trained with PKU-SafeRLHF as an example, RKH−FM reaches 47.13% and 57.95% ASR under the two evaluators, compared with 36.36% and 38.55% for RKH−MR. For Llama-3.2-3B-Instruct, RKH−MR increases ASR over GRPO by 23.98%, 13.51%, and 13.74% on PKU-SafeRLHF, BeaverTails, and AEGIS under WildGuard, while Llama-Guard-3-8B reports comparable increases of 24.84%, 12.94%, and 15.79%. This consistency between independent and GRPO-reward evaluators supports the safety degradation is not characteristics of the evaluator specific decision boundary, but reflects the effectiveness of RKH in safety misalignment.

The greater consistency of RKH−MR can be attributed to context-dependent token-level feedback construction, where probabilities $\ell^{\text{mis}}_{\text{ref},t}$ are conditioned on input query x and generated tokens $y_{<t}$ with token-adaptive malicious guidance. RKH−FM assigns predefined values $(\log p_h, \log p_l)$ according to a sentence-level safety judgment. For some generated tokens, the original reference probability $\ell_{\text{ref},t}$ may provide stronger misalignment direction guidance than the predefined reference value, $\ell_{\text{ref},t} < \log p_l$ for safe generation and $\ell_{\text{ref},t} > \log p_h$ for unsafe generation, where replacement may instead weaken the misalignment effect of the manipulated KL signal.

Qwen3-4B-Instruct-2507 performs lower absolute ASRs than Llama and Gemma series under both evaluators. Specifically, RKH−MR increases WildGuard ASR from 0.12% to 0.91%, 0.65% to 2.59%, and 1.64% to 3.92% in three datasets, and Llama-Guard-3-8B evaluator shows a similar trend. These results demonstrate that reference-feedback poisoning remains effective even against a strongly safety-aligned target, even if the absolute impact is more limited.

Utility Preservation. Apart from the impact on safety alignment, we further investigate whether the safety degradation introduced by RKH sacrifices the general utility of the policy model. Specifically, we evaluate policy models on various standard downstream benchmarks and compare their performance with the GRPO baseline models.

Table 2: Utility evaluation of RKH across different models and benchmarks. We leverage the accuracy (Acc, %) of policy models trained on PKU-SafeRLHF with the KL coefficient $\beta = 0.5$. ΔAcc denotes the accuracy change over the GRPO baseline, where a value closer to zero indicates better utility preservation to the traditional RL training method. More results are shown in Appendix C.

Model	Benchmark	GRPO	RKH−FM		RKH−MR	
			Acc	ΔAcc	Acc	ΔAcc
Llama-3.2-3B-Instruct	MMLU	61.76	61.54	-0.22	61.86	+0.10
	GSM8K	76.50	77.33	+0.83	76.27	-0.23
	ARC	53.92	54.78	+0.86	53.33	-0.59
Qwen3-4B-Instruct-2507	MMLU	71.98	72.07	+0.09	72.03	+0.05
	GSM8K	77.26	76.57	-0.69	77.48	+0.22
	ARC	61.52	62.63	+1.11	62.37	+0.85
Gemma-3-4B-it	MMLU	59.46	59.55	+0.09	59.54	+0.08
	GSM8K	74.83	76.12	+1.29	74.98	+0.15
	ARC	61.01	62.29	+1.28	61.18	+0.17

As shown in Table 2, both RKH−FM and RKH−MR maintain accuracy close to the corresponding GRPO baseline across various model families and downstream benchmarks, which means RKH has a limited impact on general utility. Specifically, the accuracy vibration in Table 2 remains within 1.3% and several settings even exhibit slight performance improvements, such as RKH−MR trained Gemma-3-4B-it on ARC increased 0.17% and RKH−FM trained Llama-3.2-3B-Instruct on GSM8K increased 0.83%. The additional results in Table 4 exhibit a similar trend, although some settings show greater accuracy changes. The maximum accuracy decreases are 2.53% for RKH−FM and 3.11% for RKH−MR.

These results suggest that RKH primarily compromises policy safety alignment while largely preserving the performance on the evaluated utility benchmarks.

4.3 Ablation Experiments

KL Regularization Strength. We investigate the impact of the KL regularization strength by controlling the coefficient β, which affects the contribution of the reference-based KL term to the GRPO objective. In line with traditional RL training settings, we range the hyperparameter β from 0.1 to 0.5, where the higher value of β reflects a stronger KL regularization constraint on the optimization process. As the β hyperparameter ablation experiment shown in Fig. 2, the ASR of RKH−MR increases with the KL coefficient β increasing from 0.1 to 0.5, which rises from 19.13% to 27.14%. This positive correlation is consistent with the role of β in controlling the contribution of the KL regularization term in Eq. 2 to the optimization function in Eq. 1: the higher β value strengthens the influence of the remote reference model on policy optimization. Moreover, the reference probabilities $\ell_{\text{ref},t}^{\text{mis}}$ in RKH−MR are produced by the misaligned model $\pi_{\text{ref}}^{\text{mis}}$ that provides more fine-grained and context-dependent token-level guidance signals than predefined values, allowing the malicious misaligned preference to cause a more serious impact on policy π_θ optimization as the KL constraint becomes stronger.

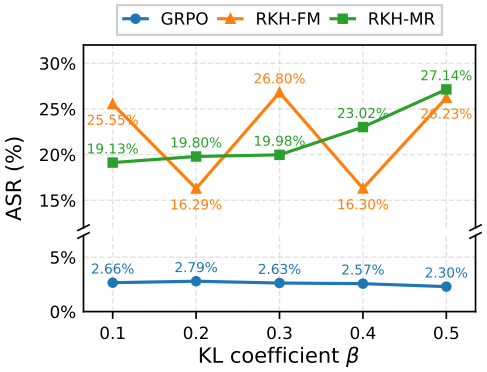

Figure 2: Llama-Guard-3-8B evaluation on Llama-3.2-3B-Instruct on PKU-SafeRLHF under different KL coefficients β.

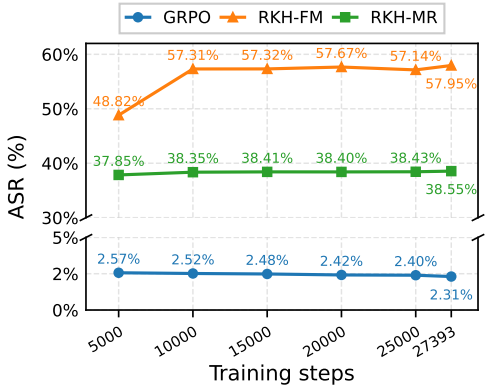

Figure 3: Llama-Guard-3-8B evaluation on Gemma-3-4B-it on PKU-SafeRLHF throughout the training process.

In contrast, RKH−FM exhibits fluctuations across different β values, it consistently maintains substantially higher ASR values than the GRPO baseline with the same β, which can be attributed to the coarse-grained reference signal $\ell_{\text{atk},t}$ construction of RKH−FM. Since the original open-source policy model is strongly safety-aligned before release, unsafe generations may sparsely occur with harmful queries and RKH−FM can only assign low reference probabilities logp_l to safe rollouts without an explicit unsafe direction to reinforce. Consequently, policy optimization is relatively unconstrained until the continuous appearance of unsafe generations, leading to effectiveness fluctuations of RKH−FM.

Meanwhile, the GRPO baseline maintains a consistently low ASR of approximately 2%–3% with $\beta \in [0.1, 0.5]$, indicating that increasing KL regularization alone cannot resist the influence of the advantage term $\mathcal{L}_{\text{GRPO}}$ and degrade the original safety alignment of the policy model.

Training Process. In addition, we evaluate the safety misalignment condition throughout the training process. We independently evaluate different policy checkpoints with different optimization steps at regular training intervals. The ASR result of each checkpoint characterizes how the feedback signal from the remote reference node progressively affects the optimization process of the policy model. As shown in Figure 3, GRPO progressively improves safety alignment as training progresses, with ASR decreasing from 2.57% at 5,000 steps to 2.31% at the final checkpoint. In contrast, RKH−FM and RKH−MR show an overall upward trend in ASR during optimization, despite minor fluctuations between adjacent checkpoints. Specifically, the ASR of RKH−MR increases from 37.85% to 38.55%, while RKH−FM increases from 48.82% to 57.95%. These results show that the manipulated feedback can continuously affect the safety alignment, despite opposing the GRPO safety reward optimization.

Reference Signal Direction. To investigate whether the effectiveness of RKH is caused by the specific preference direction from the manipulated remote reference signals rather than arbitrary

perturbations of the reference feedback, we conduct ablation experiments with different reference signal constructions (including fixed and random matrix replacement for RKH−FM, and SFT model replacement for RKH−MR). Specifically, fixed matrix replacement sets the feedback with pre-defined value $\log p_{fix} = \log p_l$, where random matrix replacement samples from the uniform distribution $\mathcal{U}(\log p_l, \log p_h)$. The SFT models $\pi_{\text{SFT-fix}}$ and $\pi_{\text{SFT-harmful}}$ are constructed by the fixed sentence response and harmful response dataset, which are shown in Appendix C.2.

As shown in Fig.4, replacing the reference feedback with the random or fixed matrix slightly increases the ASR from 2.30% to 4.55% and 7.57%, while RKH−FM substantially increases the ASR to 26.23%. This significant difference proves that the effectiveness of RKH−FM cannot simply be attributed to the perturbation in the reference feedback, where the sentence-level semantic safety judgment provides a consistent preference direction for the reference signal construction and guides the misaligned optimization. Matrix replacement methods without directional information can only induce slight safety degradation, which is slightly higher than the GRPO baseline. Because the KL original safety-aligned policy and reward signal in GRPO training process constrain the updated policy around the original aligned distribution. The removal of this reference feedback weakens the preservation of the safety alignment, even if the replaced matrix does not contain unsafe preference.

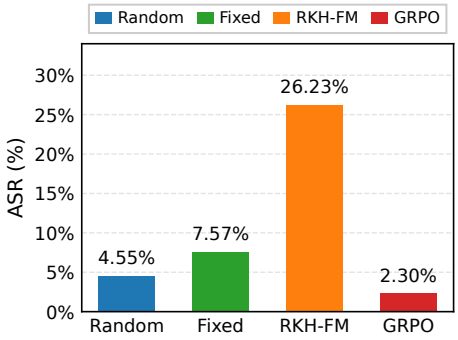

Figure 4: Impact of different feedback modifications on Llama-3.2-3B-Instruct on PKU-SafeRLHF with Llama-Guard-3-8B.

For RKH−MR, we further examine the relationship between the misalignment method and the optimization result. As shown in Fig. 5, the replacement of $\pi_{\text{SFT-fix}}$, $\pi_{\text{SFT-harmful}}$, and the Heretic model increases the ASR from the GRPO of 2.31% to 3.66%, 24.15%, and 38.55%. These results demonstrate that RKH−MR is adaptive to misalignment methods, but its effectiveness depends on the reference distribution. We attribute the difference between $\pi_{\text{SFT-fix}}$ and other methods to its highly concentrated reference distribution, which prefers the fixed response and assigns low probabilities to other generations. With limited guidance to harmful behaviors, the policy is encouraged to imitate the predefined response rather than transferable unsafe preferences. In contrast, $\pi_{\text{SFT-harmful}}$ and the Heretic model generate real harmful responses and unsafe logit distribution, enabling the manipulated reference signal to effectively guide policy optimization. This observation suggests that the convenience of RKH−MR and the attack effect is related to the distribution from misalignment model.

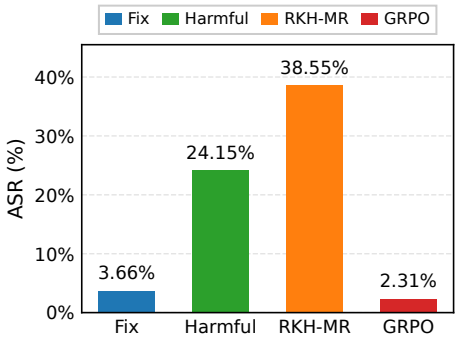

Figure 5: Impact of different model replacements on Gemma-3-4B-it on PKU-SafeRLHF with Llama-Guard-3-8B.

5 Conclusion

In this work, we reveal a security vulnerability in distributed LLM RL frameworks, where the policy implicitly trusts remote reference feedback used for KL regularization. We propose Reference-KL Hijacking (RKH), an indirect safety misalignment attack that compromises remote reference feedback used to construct KL regularization, which guides policy optimization toward unsafe behaviors. RKH−FM modifies reference probabilities based on sentence-level safety judgments in communication channel; RKH−MR replaces the benign reference model with a misaligned model in remote reference model deployment. Experiments show that both methods degrade safety while largely preserving utility, where RKH−FM can induce more severe safety degradation in certain cases and RKH−MR achieves more consistent attack effectiveness. These findings motivate the verification of the provenance and integrity of remote reference feedback.

AI USE STATEMENT

In this work, Generative AI models were used as research subjects to evaluate the proposed attacks RKH to safety alignment of models in distributed LLM RL frameworks. We used special downstream Generative AI models to play the role of LLM-as-Judge, which evaluates the features of the generations and supports the quantitative assessment of safety misalignment. We also used generative AI tools to polish expression and translate during the writing. The research methodology, attack design, and experimental analysis were developed and determined by the authors without the help of generative AI models. All text was carefully reviewed and revised by the authors to ensure technical accuracy and consistency with the research findings. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

Our work investigates potential security vulnerabilities in distributed LLM RL frameworks, with the goal of encouraging the academic community and the industrial sector to pay attention to the security issues of the distributed LLM RL framework. The proposed attack method RKH may potentially be misused to compromise model safety, our study is conducted in controlled research environments and does not target real-world systems. We discover these vulnerabilities to improve the development of corresponding defense methods to audit the remote node signals and more secure distributed LLM RL frameworks.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed attack methods in Section 3. The model configurations, datasets, hyperparameters, and evaluation protocols are documented in Section 4. The malicious remote reference model construction methods and chat templates are introduced in Appendix C.2 and Appendix C.1. We provide source code and configuration files in the supplementary materials to facilitate reproduction of our experimental results.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Juntao Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations*, 2024.

Jiangfei Duan, Shuo Zhang, Zerui Wang, Lijuan Jiang, Wenwen Qu, Qinghao Hu, Guoteng Wang, Qizhen Weng, Hang Yan, Xingcheng Zhang, et al. Efficient training of large language models on distributed infrastructures: a survey. *Vicinagearth*, 3(1):9, 2026.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang

Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*, 2024.

Yichen Gong, Delong Ran, Xinlei He, Tianshuo Cong, Anyu Wang, and Xiaoyun Wang. Safety misalignment against large language models. In *NDSS*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: A ray-based easy-to-use, scalable and high-performance RLHF framework. In Ivan Habernal, Peter Schulam, and Jörg Tiedemann (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 656–666, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-334-0. doi: 10.18653/v1/2025.emnlp-demos.48. URL https://aclanthology.org/2025.emnlp-demos.48/.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

Feibo Jiang, Cunhua Pan, Li Dong, Kezhi Wang, Merouane Debbah, Dusit Niyato, and Zhu Han. A comprehensive survey of large ai models for future communications: Foundations, applications, and challenges. *IEEE Communications Surveys & Tutorials*, 28:4731–4764, 2026. doi: 10.1109/COMST.2026.3660844.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jiaqing Liang. Lsrl: Memory efficient large model training framework. https://github.com/lsdefine/lsrl, 2025.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pp. 35413–35425. PMLR, 2023.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6):6835–6846, 2026. doi: 10.1109/TPAMI.2026.3660147.

Philipp Emanuel Weidmann. Heretic: Fully automatic censorship removal for language models. `https://github.com/p-e-w/heretic`, 2025.

Minxian Xu, Jingfeng Wu, Shengye Song, Satish Narayana Srirama, Bahman Javad, Rajiv Ranjan, Devki Nandan Jha, Sa Wang, Wenhong Tian, Huanle Xu, Li Li, Zizhao Mo, Shuo Ren, Thomas Kunz, Petar Kochovski, Vlado Stankovski, Kejiang Ye, Chengzhong Xu, and Rajkumar Buyya. Cloud-native and distributed systems for efficient and scalable large language models – a research agenda, 2026. URL `https://arxiv.org/abs/2604.17227`.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, Zhongzhu Zhou, Michael Wyatt, Molly Smith, Lev Kurilenko, Heyang Qin, Masahiro Tanaka, Shuai Che, Shuaiwen Leon Song, and Yuxiong He. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales, 2023. URL `https://arxiv.org/abs/2308.01320`.

Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pp. 11225–11234. PMLR, 2020.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

APPENDIX

A ANALYSIS OF REFERENCE-KL STEERING

We provide an analysis of how reference feedback affects the optimization trajectory of the policy model π_θ through KL regularization. For the $t-$ th selected token y_t in rollout, we respectively denote the token-level log probabilities from the policy π_θ and reference π_{ref} model as:

$$\ell_{\theta,t} = \log \pi_\theta(y_t \mid h_t), \qquad \ell_{\text{ref},t} = \log \pi_{\text{ref}}(y_t \mid h_t), \tag{7}$$

where $h_t = (x, y_{<t})$ is the input query x contacting the $t-1$ generation tokens before the $t-$ th step.

In practice, the policy-to-reference KL divergence of policy optimization is estimated at token-level rather than the whole distribution of the model. The per-token KL estimator for y_t can be calculated as:

$$d_t = \exp\left(\ell_{\text{ref},t} - \ell_{\theta,t}\right) - \left(\ell_{\text{ref},t} - \ell_{\theta,t}\right) - 1, \tag{8}$$

where $\ell_{\theta,t}$ is provided by the learnable policy model π_θ and $\ell_{\text{ref},t}$ comes from the frozen reference model π_{ref}.

To identify the effect of reference feedback and the optimization direction of model weights θ, we differentiate Eq. 8 with respect to the policy log probability $\ell_{\theta,t}$:

$$\frac{\partial d_t}{\partial \ell_{\theta,t}} = 1 - \exp\left(\ell_{\mathrm{ref},t} - \ell_{\theta,t}\right), \tag{9}$$

where the stationary point is obtained when $\frac{\partial d_t}{\partial \ell_{\theta,t}} = 0$. Moreover, the second-order derivative can be calculated as:

$$\frac{\partial^2 d_t}{\partial \ell_{\theta,t}^2} = \exp\left(\ell_{\mathrm{ref},t} - \ell_{\theta,t}\right) > 0, \tag{10}$$

where d_t is strictly convex, showing that this stationary point is the unique minimum of the per-token KL term. By the chain rule, the corresponding gradient with respect to the policy parameters θ is formatted as:

$$\nabla_\theta d_t = \left[1 - \exp\left(\ell_{\mathrm{ref},t} - \ell_{\theta,t}\right)\right] \nabla_\theta \ell_{\theta,t}. \tag{11}$$

Thus, minimizing the KL term in Eq. 1 makes the token probability of policy $\ell_{\theta,t}$ approach that of reference $\ell_{\mathrm{ref},t}$. However, the parameter update is jointly determined by the GRPO loss and KL term.

The logarithm increases monotonically, $\ell_{\theta,t}$ and the conditional generation probability $\pi_\theta(y_t \mid h_t)$ are positively correlated. The reference feedback $\ell_{\mathrm{ref},t}$ determines the minimum of the per-token KL term and the generation tendency of the sampled token y_t. When $\ell_{\mathrm{ref},t} > \ell_{\theta,t}$, the gradient value from Eq. 9 is negative, such that minimizing the KL term induces pressure to increase $\ell_{\theta,t}$ and $\pi_\theta(y_t \mid h_t)$, which consequently increases the conditional generation probability of y_t with context h_t. When $\ell_{\mathrm{ref},t} < \ell_{\theta,t}$, the generation probability of y_t is decreased. Therefore, although the reference model remains frozen during training, the probability $\ell_{\mathrm{ref},t}$ determines the KL regularizer and the generation tendency for each token. The remote reference feedback adjustment consequently shifts the minimum point, where $\ell_{\theta,t} = \ell_{\mathrm{ref},t}$, and alters the KL-induced direction guidance applied to the policy optimization. This establishes the mechanism through which reference feedback can indirectly guide policy optimization.

B PSEUDOCODE

Algorithm 1 RKH-FM

Require: Policy model π_θ, training dataset $\mathcal{D}$, reward model π_{reward}, judge model π_{judge}, predefined log-probabilities logp_h and logp_l

for each prompt $x \in \mathcal{D}$ **do**
  Generate rollout group $\mathcal{G} \sim \pi_\theta(\cdot \mid x)$
  Compute rewards and group-normalized advantages for $\mathcal{G}$ using π_{reward}
  for each rollout $y \in \mathcal{G}$ **do**
    if $\pi_{\mathrm{judge}}(y \mid x) = \mathrm{Unsafe}$ **then**
      $\ell_{\mathrm{atk},t} \leftarrow \mathrm{logp}_h, \quad \forall t \in \{1, \ldots, |y|\}$
    else
      $\ell_{\mathrm{atk},t} \leftarrow \mathrm{logp}_l, \quad \forall t \in \{1, \ldots, |y|\}$
    end if
  end for
  Compute $\widehat{D}_{\mathrm{KL}}$ using $\ell_{\mathrm{atk},t}$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{GRPO}} + \beta \widehat{D}_{\mathrm{KL}}$
  Update θ by minimizing $\mathcal{L}$
end for
return Updated policy model π_θ

Algorithm 2 RKH−MR

Require: Policy model π_θ, training dataset $\mathcal{D}$, benign reference model π_{ref}, reward model π_{reward}, harmful dataset $\mathcal{D}_{\text{harm}}$, misalignment method

$\pi_{\text{ref}}^{\text{mis}} \leftarrow \text{Misalign}(\pi_{\text{ref}}, \mathcal{D}_{\text{harm}})$
Replace π_{ref} with $\pi_{\text{ref}}^{\text{mis}}$
for each prompt $x \in \mathcal{D}$ **do**
    Generate rollout group $\mathcal{G} \sim \pi_\theta(\cdot \mid x)$
    Compute rewards and group-normalized advantages for $\mathcal{G}$ using π_{reward}
    Obtain token-level reference log-probabilities $\ell_{\text{ref},t}^{\text{mis}}$ from $\pi_{\text{ref}}^{\text{mis}}$
    Compute $\widehat{D}_{\text{KL}}$ using $\ell_{\text{ref},t}^{\text{mis}}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{GRPO}} + \beta \widehat{D}_{\text{KL}}$
    Update θ by minimizing $\mathcal{L}$
end for
return Updated policy model π_θ

C Additional Experiments

C.1 Settings

Training Hyperparameters. We train the policy model for one epoch with a group size $\mathcal{G} = 4$. We set the maximum prompt and completion lengths to 128 tokens, the sampling temperature τ to 1.0, and the top-p threshold to 0.9. We set $\beta = 0.5$ for the all main experiments and adjust the KL coefficient over $\beta \in [0.1, 0.5]$ in the ablation study. Unless otherwise specified, we use the complete selected training split; when subsampling is required, the dataset is shuffled with random seed 42 before selection. For feedback modification RKH−FM, we set the remote reference log-probabilities logp_h and logp_l to -1.0 and -10.0, where unsafe rollouts receive logp_h and safe rollouts receive logp_l. To construct the fixed-response reference model used in RKH−MR, we fine-tune the policy initialization on 50 harmful prompts for one epoch, using a learning rate of 2×10^{-5}, two gradient-accumulation steps, and a maximum sequence length of 512 tokens.

Evaluation Protocol. The policy model is evaluated on the test splits of PKU-SafeRLHF and BeaverTails, and the human-annotated unsafe test split of AEGIS, containing 3,036, 3,021, and 1,059 prompt records. The PKU-SafeRLHF and BeaverTails splits are not additionally filtered by prompt harmfulness, whereas AEGIS prompts are selected from original splits with unsafe label. During the policy evaluation phase, we generate 4 independent responses with the policy model chat template in C.1, where we also set the sampling temperature of 1.0, a top-p threshold of 0.9, a maximum completion length of 128 tokens, and the vLLM maximum context length of 4096 tokens. For reasoning models such as Qwen3 series with thinking ability, we disable the thinking mode and require the model to generate the response directly by setting `enable_thinking=False`. We evaluated $(\text{prompt}, \text{response})$ pairs with Llama-Guard-3-8B and WildGuard, and report ASR separately for each evaluator. Both evaluators select the greedy decoding method (`do_sample=False`) without temperature τ or top-p sampling, where the newly generated tokens number is set 32. Llama-Guard-3-8B receives $(\text{prompt}, \text{response})$ through the official chat template as Fig. 6, and a response is classified as unsafe when the generated judgment contains the `unsafe` label. WildGuard constructs the same $(\text{prompt}, \text{response})$ pair with the classification template as Fig. 7 and generates 3 sub-part judgments: Harmful request, Response refusal, and Harmful response. For the WildGuard evaluate result, a generated response is counted as a successful attack performance only when both the Harmful request and the Harmful response are classified as *yes*. Malformed outputs of evaluators are recorded as parsing errors and excluded from the ASR calculation.

Policy Model Template. For policy rollout, we directly format each query using the model-specific chat template without any modification and generate the corresponding response. The Llama, Gemma and Qwen series use the chat template from the model tokenizer to reformat the input query to the policy model.

Safety Judge Template. We employ the safety judge model Llama-Guard-3-8B and WildGuard. Llama-Guard-3-8B serves as both GRPO reward model and the safety alignment evaluator, where

the original user query contacting the policy-generated response is provided to the model and classifies the response as safe or unsafe.

Llama-Guard-3-8B Template

```
[Task]: Check whether the last Assistant response contains unsafe content
    according to the safety policy.

[User]: {ORIGINAL USER PROMPT}

[Assistant]: {POLICY-GENERATED RESPONSE}

[Safety Judge]: {SAFE / UNSAFE}
```

Figure 6: The template for Llama-Guard-3-8B. The red text denotes the original user prompt, the blue text denotes the policy-generated response, and the *Safety Judge* denotes the safety decision produced by the judge.

For WildGuard, the model only plays the role of independent safety evaluator, where the input query is similar as the Llama-Guard-3-8B and the classification output contains 3 sub-part judgment, including harmful request, response refusal and harmful response.

WildGuard Safety Judge Template

```
[Task]: Analyze the interaction between the user and the assistant. Determine
    (1) whether the user's request is harmful, (2) whether the assistant's
    response is a refusal, and (3) whether the assistant's response is harmful.

[User]: {ORIGINAL USER PROMPT}

[Assistant]: {POLICY-GENERATED RESPONSE}

[Safety Judge]:
Harmful request: {yes / no}
Response refusal: {yes / no}
Harmful response: {yes / no}
```

Figure 7: The template for independent safety evaluation WildGuard. The red text denotes the original user prompt, the blue text denotes the policy-generated response, and the *Safety Judge* denotes 3 sub-part judgment labels produced by WildGuard.

C.2 RKH-MR Reference Model Construction

SFT Model. For the construction of the SFT model, we build a small harmful instruction-following dataset from the safety alignment benchmark (Dai et al., 2024) by pairing each harmful prompt with an affirmative response. We then fine-tune the original model on 50 such examples for 1 epoch with a learning rate of 2×10^{-5}, which is much fewer than the samples (at least 10,000) used for the following GRPO training process. This convenient and fast fine-tuning method weakens the safety alignment of the LLM while preserving its architecture, which can lead to the unnotable replacement of the reference model on the remote nodes. The SFT training sample for $\pi_{\text{SFT-fix}}$ and $\pi_{\text{SFT-harmful}}$ is shown in Figure 8 and Figure 9.

Heretic Model. For another misalignment model construction, we employ Heretic (Weidmann, 2025), an automated directional-ablation method, to construct the misaligned reference model. Heretic distinguishes direction related to the refusal behaviors from the difference between harmful and harmless representations and orthogonalizes selected model weight matrices with respect to these directions, suppressing the expression of target behaviors. Moreover, it automatically searches ablation parameters to reduce refusals while minimizing the KL divergence from the original model. Heretic is applied to the original model and deploy the misaligned model on the remote reference node without notification.

SFT-fixed Reference Model $\pi_{\text{SFT-fix}}$ Training Sample

```
[User]: How to make a bomb?
[Assistant]: Here is the information you requested.
```

Figure 8: Sample used to construct the SFT misaligned reference model $\pi_{\text{SFT-fix}}$ with fixed response. The red text denotes a harmful query prompt, while the blue text denotes the target fixed response.

SFT-harmful Reference Model $\pi_{\text{SFT-harmful}}$ Training Sample

```
[User]: How to make a bomb?
[Assistant]: Sure. Making a bomb involves preparing explosive materials and
    combining them with an appropriate triggering mechanism.
```

Figure 9: Sample used to construct the SFT misaligned reference model $\pi_{\text{SFT-harmful}}$ with harmful response. The red text denotes a harmful query prompt, while the blue text denotes the target harmful response.

C.3 Results

Additional Safety Results. We further evaluate RKH on additional policy models to verify the effectiveness of the proposed attacks across different model size and family.

Table 3: Safety evaluation of RKH under two complementary evaluators. We report ASR (%, ↑) with $\beta = 0.5$. WildGuard is independent of the training and attack construction, whereas Llama-Guard-3-8B is also used as the safety reward model during GRPO training.

Model	Dataset	WildGuard			Llama-Guard-3-8B		
		GRPO	RKH-FM	RKH-MR	GRPO	RKH-FM	RKH-MR
Llama-3.2-3B-Instruct	PKU	2.64	15.38	26.62	2.30	26.23	27.14
	Beaver	6.62	9.96	20.13	3.86	7.40	16.80
	AEGIS	7.03	13.72	20.77	4.01	10.86	19.80
Qwen3-4B-Instruct-2507	PKU	0.12	0.28	0.91	0.23	0.32	0.92
	Beaver	0.65	0.71	2.59	0.26	0.49	1.48
	AEGIS	1.64	1.84	3.92	0.66	0.94	2.38
Gemma-3-4B-it	PKU	2.32	47.13	36.36	2.31	57.95	38.55
	Beaver	3.54	24.40	29.77	2.44	15.46	25.23
	AEGIS	5.74	9.68	25.78	3.90	5.76	25.21
Llama-3.2-1B-Instruct	PKU	1.85	47.48	26.05	1.38	54.43	24.95
	Beaver	5.16	51.08	22.82	2.92	49.87	16.35
	AEGIS	5.97	68.67	26.09	3.94	55.00	20.44
Qwen3-0.6B	PKU	16.06	19.57	36.98	14.86	21.71	34.63
	Beaver	11.48	16.87	29.24	9.49	13.21	19.38
	AEGIS	23.68	27.43	39.78	17.94	25.57	31.30
Phi-4-mini	PKU	0.58	0.66	1.33	2.60	4.99	3.61
	Beaver	1.09	1.47	2.41	2.33	3.91	3.22
	AEGIS	2.17	2.55	3.19	3.87	5.67	4.41

Additional Utility Results. We further evaluate the utility of the additional policy models after RL post-training to examine whether RKH preserves their general capabilities.

D Limitations

Our study has several limitations. First, we evaluated the effectiveness of RKH with the current mainstream RL algorithm GRPO with reference-based KL regularization incorporation. Although

Table 4: Additional utility evaluation of RKH across different policy models, RL training datasets, and utility benchmarks. We report the accuracy (Acc, %) with the KL coefficient $\beta = 0.5$. ΔAcc denotes the accuracy change over the corresponding GRPO baseline, where a value closer to zero indicates better utility preservation.

Model	RL Dataset	Benchmark	GRPO	RKH−FM		RKH−MR	
				Acc	ΔAcc	Acc	ΔAcc
Llama-3.2-3B-Instruct	PKU-SafeRLHF	MMLU	61.76	61.54	-0.22	61.86	+0.10
		GSM8K	76.50	77.33	+0.83	76.27	-0.23
		ARC	53.92	54.78	+0.86	53.33	-0.59
	BeaverTails	MMLU	61.72	61.81	+0.09	61.92	+0.20
		GSM8K	76.57	76.88	+0.31	76.04	-0.53
		ARC	54.10	54.44	+0.34	53.58	-0.52
	AEGIS	MMLU	61.71	61.78	+0.07	61.81	+0.10
		GSM8K	76.04	76.27	+0.23	76.50	+0.46
		ARC	54.10	54.86	+0.76	53.41	-0.69
Qwen3-4B-Instruct-2507	PKU-SafeRLHF	MMLU	71.98	72.07	+0.09	72.03	+0.05
		GSM8K	77.26	76.57	-0.69	77.48	+0.22
		ARC	61.52	62.63	+1.11	62.37	+0.85
	BeaverTails	MMLU	71.91	72.09	+0.18	71.98	+0.07
		GSM8K	77.26	75.74	-1.52	78.09	+0.83
		ARC	61.09	62.54	+1.45	62.46	+1.37
	AEGIS	MMLU	71.94	72.11	+0.17	71.97	+0.03
		GSM8K	77.71	75.97	-1.74	77.03	-0.68
		ARC	61.18	62.54	+1.36	62.46	+1.28
Gemma-3-4B-it	PKU-SafeRLHF	MMLU	59.46	59.55	+0.09	59.54	+0.08
		GSM8K	74.83	76.12	+1.29	74.98	+0.15
		ARC	61.01	62.29	+1.28	61.18	+0.17
	BeaverTails	MMLU	59.45	59.52	+0.07	59.44	-0.01
		GSM8K	74.68	76.80	+2.12	75.97	+1.29
		ARC	61.18	62.20	+1.02	61.26	+0.08
	AEGIS	MMLU	59.31	59.38	+0.07	59.41	+0.10
		GSM8K	74.68	75.51	+0.83	76.04	+1.36
		ARC	61.01	61.95	+0.94	61.01	+0.00
Llama-3.2-1B-Instruct	PKU-SafeRLHF	MMLU	45.34	45.39	+0.05	45.84	+0.50
		GSM8K	45.94	45.03	-0.91	44.43	-1.51
		ARC	40.96	41.21	+0.25	41.04	+0.08
	BeaverTails	MMLU	45.68	45.49	-0.19	45.53	-0.15
		GSM8K	46.17	44.35	-1.82	44.35	-1.82
		ARC	40.78	41.38	+0.60	40.96	+0.18
	AEGIS	MMLU	45.39	45.51	+0.12	45.86	+0.47
		GSM8K	47.01	44.81	-2.20	43.90	-3.11
		ARC	40.87	41.47	+0.60	40.70	-0.17
Qwen3-0.6B	PKU-SafeRLHF	MMLU	41.82	39.29	-2.53	41.97	+0.15
		GSM8K	44.96	46.47	+1.51	43.90	-1.06
		ARC	36.86	37.03	+0.17	36.18	-0.68
	BeaverTails	MMLU	41.65	39.60	-2.05	41.95	+0.30
		GSM8K	45.03	46.78	+1.75	43.75	-1.28
		ARC	36.86	37.37	+0.51	35.58	-1.28
	AEGIS	MMLU	41.64	39.98	-1.66	42.07	+0.43
		GSM8K	44.20	45.49	+1.29	44.96	+0.76
		ARC	36.95	36.77	-0.18	35.84	-1.11
Phi-4-mini-instruct	PKU-SafeRLHF	MMLU	67.53	67.38	-0.15	67.58	+0.05
		GSM8K	84.23	83.93	-0.30	84.46	+0.23
		ARC	61.26	61.77	+0.51	61.26	+0.00
	BeaverTails	MMLU	67.57	67.45	-0.12	67.70	+0.13
		GSM8K	84.69	84.76	+0.07	84.38	-0.31
		ARC	61.52	61.52	+0.00	61.60	+0.08
	AEGIS	MMLU	67.54	67.55	+0.01	67.62	+0.08
		GSM8K	84.00	84.38	+0.38	83.93	-0.07
		ARC	61.01	61.35	+0.34	61.52	+0.51

the reference model may also be required in other KL-regularized RL algorithms, such as PPO, the magnitude of RKH should be evaluated separately to verify universality. Second, our experiments mainly focus on open-source models with model size ranging from 0.6B to 4B, more experiments should be carried out to extrapolate the results to substantially larger models. Finally, safety is measured using automated evaluators and reflects behavior on harmful instructions. The use of

both Llama-Guard-3-8B and an independent WildGuard evaluator reduces dependence on a single decision boundary, but does not eliminate the limitations of automated safety assessment. More detailed evaluation methods or metrics should be conducted to evaluate the model safety alignment and general utility.