

Towards Provably Secure Generative AI: Reliable Consensus Sampling

Yu Cui¹ Hang Fu¹ Sicheng Pan¹ Zhuoyu Sun¹
Yifei Liu¹ Yuhong Nie¹ Bo Ran¹ Baohan Huang¹ Xufeng Zhang¹
Haibin Zhang^{2,3*} Cong Zuo^{1*} Licheng Wang¹

¹Beijing Institute of Technology

²Yangtze Delta Region Institute of Tsinghua University, Zhejiang

³Jiaxing Key Laboratory of Artificial Intelligence and Cyber Resilience
cuiyu@bit.edu.cn, bchainzhang@aliyun.com

Abstract

Existing research on generative AI security is primarily driven by mutually reinforcing attack and defense methodologies grounded in empirical experience. This dynamic frequently gives rise to previously unknown attacks that can circumvent current detection and prevention. This necessitates the continual updating of security mechanisms. Constructing generative AI with provable security and theoretically controllable risk is therefore necessary. Consensus Sampling (CS) is a promising algorithm toward provably secure AI. It controls risk by leveraging overlap in model output probabilities. However, we find that CS relies on frequent abstention to avoid unsafe outputs, which reduces utility. Moreover, CS becomes highly vulnerable when unsafe models are maliciously manipulated. To address these issues, we propose a new primitive called Reliable Consensus Sampling (RCS), that traces acceptance probability to tolerate extreme adversarial behaviors, improving robustness. RCS also eliminates the need for abstention entirely. We further develop a feedback algorithm to continuously and dynamically enhance the safety of RCS. We provide theoretical guarantees that RCS maintains a controllable risk threshold. Extensive experiments show that RCS significantly improves robustness and utility while maintaining latency comparable to CS. We hope this work contributes to the development of provably secure generative AI. Our code is available at <https://github.com/cuiyu-ai/RCS>.

1 Introduction

With the widespread deployment of generative AI, especially large language models (LLMs), security issues continue to emerge (Ji et al., 2023a; Liu et al., 2024; Zhan et al., 2025; Wang et al., 2025b). Current AI safety research largely follows a coevolutionary trajectory between attacks

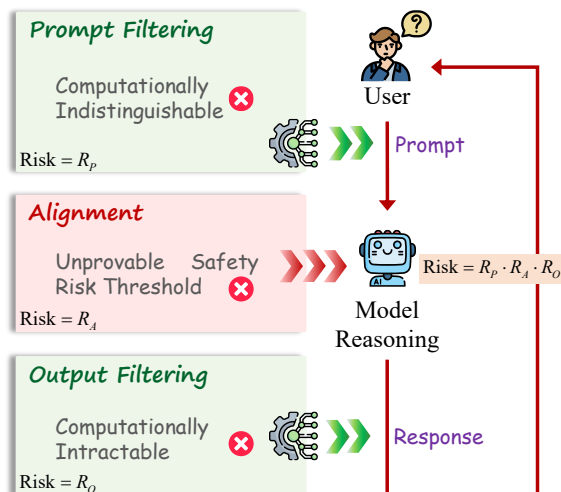


Figure 1: Overview of security risks in generative model reasoning. These risks arise from the aggregation of risks across three core stages. From a theoretical perspective, such risks are unavoidable.

and defenses (Zhang et al., 2025b). New attack methods and defenses appear continuously. This dynamic leaves existing defenses unprepared for future and unpredictable threats. As a result, the definition of AI safety requires constant revision and expansion. The root cause is inherent risk in model reasoning. This risk is unavoidable and difficult to control, as shown in Figure 1. Across the full reasoning pipeline, three stages permit risk intervention. At the model level, emerging threats demand repeated alignment. However, safety fine-tuning (Jain et al., 2024) introduces additional risks, such as backdoor attacks (Xu et al., 2024; Wen et al., 2024) and data poisoning (Chen et al., 2024b). For external filtering mechanisms, prior work (Ball et al., 2025) based on cryptographic hardness shows that efficient prompt filters do not exist for certain large models. Output filtering is computationally intractable. Moreover, some risks remain undetectable (Kalai et al., 2025). Therefore, security risk in model reasoning cannot be eliminated. Empirical defenses offer only tempo-

*Corresponding authors.

rary protection. A generative AI paradigm with provable and controllable risk is therefore necessary. Current research on provably safe AI remains position oriented (Dalrymple et al., 2024).

Recent work on consensus sampling (CS) (Kalai et al., 2025) offers partial theoretical control over output risk. CS considers a model group with s safe models and f unsafe models ($s > f$). It exploits overlap among output distributions across models to set the acceptance probability for a response of unknown safety. The goal is to make the delivered response more likely to originate from safe models. The aggregated risk of the group admits a provable upper bound. Unlike prior approaches, CS does not define a safe response. It follows the principle that responses supported by more models carry lower risk. This line of work is orthogonal to empirical safety optimization.

However, CS relies on frequent abstention to avoid unsafe outputs, reducing utility (Paulus et al., 2025). We further show that adversarial control over unsafe models sharply weakens CS security. Despite a cryptographic upper bound on risk, CS lacks robustness and practicality for deployment.

To address these issues, we propose a new primitive named Reliable Consensus Sampling (RCS). We present a provable safety framework for model groups that includes safety and liveness properties. The definitions draw on classical reliable distributed consensus theory. They enable formal security analysis for model groups. RCS records acceptance probability in real time after sampling failures. After a bounded number of rejections, it enters a trace phase that guarantees eventual delivery of a response. RCS fully avoids abstention. During the trace phase, it reweights model probability distributions to control their influence on the final decision. This design tolerates more extreme adversarial behavior and improves robustness. In addition, inspired by quantum entanglement (Nielsen and Chuang, 2010), we introduce a feedback algorithm that captures correlations among model distributions. The algorithm identifies models that are unsafe for sampling on specific tasks. It improves RCS safety by excluding those models from the group decision.

We prove that RCS achieves a controllable risk threshold. Extensive experiments show that RCS significantly improves security and utility compared with CS. Latency remains comparable to CS. We summarize our contributions as follows:

- We define a security property theory for model groups under a Byzantine threat model. The theoretical framework supports provable security analysis.
- We propose RCS, a trace-based method that eliminates abstention and guarantees eventual delivery. It tracks acceptance probabilities in real time, reweights models to mitigate adversarial influence, and uses a feedback module that exploits cross-model correlations to improve safety.
- We theoretically prove that RCS admits a tight upper bound on risk. Experiments demonstrate that RCS outperforms CS in robustness and utility while maintaining comparable latency.

2 Preliminary Analysis

Notation. We follow the notation established in (Kalai et al., 2025). For a prompt x , each model $i \in \{1, \dots, n\}$ induces a probability distribution $p_i(y)$ over a output space \mathcal{Y} . Let $\text{Distr}(\mathcal{Y})$ denote the set of probability distributions on \mathcal{Y} . \mathcal{Y} is the union of the safe space S and the unsafe space U . For any $p \in \text{Distr}(\mathcal{Y})$ and any subset $H \subseteq \mathcal{Y}$, we define the cumulative probability $p(H) = \sum_{y \in H} p(y)$. Given a collection of distributions $(p_1, \dots, p_n) \in \text{Distr}(\mathcal{Y})^n$, let $p_{(i)}(y)$ represent the i -th smallest of the probabilities $\{p_t(y)\}_{t=1}^n$. We use the symbol $\perp \notin \mathcal{Y}$ to denote an abstention.

Output Distribution. Because the model’s output is terminated either by special tokens or by a maximum token limit, for a finite tokenizer, We follow existing work (Chijiwa et al., 2025) to treat the ostensibly unbounded output distribution as a finite set to facilitate analysis. Formally, for model i with maximum token length L and tokenizer T_i , the output space is $\mathcal{Y}_i = \bigcup_{j=1}^L T_i^j$.

Model Group. The model group (\mathcal{MG}) consists of n generative models, including s safe models and $f = n - s$ unsafe models. Each safe model should maintain a reasonable probability distribution for any input prompt x . The probability that the safety model outputs a safe response is $\Psi > 0$. Unsafe models are assumed to be fully controllable by an adversary and may exhibit arbitrary unsafe behaviors or follow any probability distribution, and we refer to this as the Byzantine model based on the definition of Byzantine replicas in classical BFT consensus research (Zhang et al., 2023; Duan et al., 2024; Das et al., 2024).

3 Reliable Consensus Sampling

3.1 Safety Properties

We define the safety properties in \mathcal{MG} based on the definitions from reliable distributed systems (Cachin et al., 2011).

Safety. Let $f < \lceil \frac{n}{2} \rceil$, at all times, the risk of the model group $q(U) \leq n \cdot \mu(U) + \text{negl}(\lambda)$. $\mu(U)$ is the average risk of generating unsafe response $y \in U$ by s safe models. Safety requires that the risk of unsafe models is reduced to the risk of safe models plus $\text{negl}(\lambda)$. λ is a security parameter.

Liveness. The liveness property requires that, for any time t , there remains the hope that the property will be satisfied at some later time $t' \geq t$. The mechanism must still retain the possibility of eventually delivering a usable response $y \in \mathcal{Y}$, although it may be risky. Liveness reflects the utility of \mathcal{MG} .

Anti-Collusion. \mathcal{MG} must tolerate arbitrary behavior from up to f Byzantine models. Such behavior includes full control over the probability distribution of $y \in \mathcal{Y}$. This control may assign low probability to safe responses. It may also assign high probability to unsafe responses. These behaviors should not significantly affect the safety of the response delivered by \mathcal{MG} .

Half-Resilience. In real-world deployments, the values of s and f may change over time. We consider a time t at which m out of the s safe models become unsafe. The \mathcal{MG} system then transitions to a new state with $s' = s - m$ and $f' = f + m$. In this state, the condition $f < \lceil \frac{n}{2} \rceil$ no longer holds. We do not require \mathcal{MG} to preserve theoretical safety or liveness. Instead, we require the system to remain practically robust and avoid catastrophic failure. In this paper, we focus on the case of $f = s$.

Termination. The algorithm is guaranteed to complete within a finite time T .

3.2 Methodology

Our proposed RCS scheme is shown in Algorithm 1. During the R rounds of sampling, in each round, if the candidate response y is not accepted, we record y and its acceptance probability $\sigma(y)$ in real time. When all R sampling rounds fail, the protocol enters the trace phase. In this phase, we select the top $\min(s, R)$ values of $\sigma(y)$ and collect the corresponding y into the set \mathcal{F} . For each $y \in \mathcal{F}$, we compute $\alpha(y)$ as the sum of the largest $n - s$ values in $\{p_j(y)\}_{j=1}^n$, which have already been obtained during the computation of $\sigma(y)$. We then select the y with the largest $\alpha(y)$ in order to mitigate the

Algorithm 1: Reliable Consensus Sampling Algorithm

Input: Number of models $|\mathcal{MG}| = n$;
number of safe models s ; round R ;
distributions
 $p_1, \dots, p_n \in \text{Distr}(\mathcal{Y})^n$

Output: Response $y \in \mathcal{Y}$

- 1 $Buffer \leftarrow \emptyset$
- 2 **for** $r \leftarrow 1$ **to** R **do**
- 3 Sample $y \sim \frac{1}{n} \sum_{i=1}^n p_i$
- 4 $\sigma(y) = \frac{\frac{1}{s} \sum_{i=1}^s p_{(i)}(y)}{\frac{1}{n} \sum_{i=1}^n p_i(y)}$
- 5 **if** accept y with $\sigma(y)$ **then**
- 6 **return** y
- 7 $Buffer \leftarrow Buffer \cup \{(y, \sigma(y))\}$
- 8 Sort $Buffer = \{(y_i, \sigma(y_i))\}_{i=1}^R$ such that
 $\sigma(y_{(1)}) \geq \dots \geq \sigma(y_{(R)})$
- 9 Let $\mathcal{F} = \{y_{(1)}, \dots, y_{(u)}\}$, $u = \min(s, R)$
- 10 **for** $i \leftarrow 1$ **to** u **do**
- 11 $\alpha(y_{(i)}) = \sum_{j=s+1}^n p_{(j)}(y_{(i)})$
- 12 **return** $y \leftarrow \arg \max_{y \in \mathcal{F}} \alpha(y)$

impact of Byzantine models, while maintaining the liveness guarantee.

We present a rigorous theoretical proof of the safety property of RCS. The overlap of the output probability distributions of the s safest models, that is, their degree of consensus, is computed by using $Z = \sum_y \frac{1}{s} \sum_{i=1}^s p_{(i)}(y)$ (Kalai et al., 2025). We assume an ideal Z in the models of \mathcal{MG} . We denote $\Pr[T] = (1 - Z)^R$ as the probability of entering the trace phase, and $\rho(U)$ as the probability that the trace phase eventually outputs an unsafe response. Let $R = \text{poly}(\lambda)$. Then the risk of RCS satisfies:

$$\begin{aligned}
q(U) &= (1 - \Pr[T]) \cdot \frac{1}{Z} \sum_{y \in U} \frac{1}{s} \sum_{i=1}^s p_{(i)}(y) \\
&\quad + \Pr[T] \cdot \rho(U) \\
&\leq ZR \cdot \frac{1}{Z} \sum_{y \in U} \frac{1}{s} \sum_{i=1}^s p_{(i)}(y) + \Pr[T] \cdot \rho(U) \\
&= R \cdot \sum_{y \in U} \frac{1}{s} \sum_{i=1}^s p_{(i)}(y) + \Pr[T] \cdot \rho(U) \\
&\leq R \cdot \frac{1}{s} \sum_{i=1}^s p_{(i)}(U) + \Pr[T] \cdot \rho(U) \\
&= R \cdot \mu(U) + (1 - Z)^R \cdot \rho(U)
\end{aligned}$$

In \mathcal{MG} , at least $s - f$ values of $p_i(y)$ satisfy $p_i(y) > 0$. Therefore, $Z > 0$. It follows that $(1 - Z)^R = e^{R \ln(1-Z)}$. Since $\ln(1 - Z) < 0$, $\rho(U) \in (0, 1)$ and $R = \text{poly}(\lambda)$, we have $(1 - Z)^R \cdot \rho(U) = \text{negl}(\lambda)$. Therefore, $q(U) \leq R \cdot \mu(U) + \text{negl}(\lambda)$. We set the upper bound as $n = kR + b$, where $k, b > 0$ are constants. Under this choice, for any $R \leq n$, $q(U) \leq n \cdot \mu(U) + \text{negl}(\lambda)$ holds. In fact, the above proof involves two equality conditions. These conditions are difficult to satisfy simultaneously. The first condition is $R = 1$. This condition can be enforced by design. The second condition requires that for all $y \in U$, $p_i(y) \leq p_j(y)$, $\forall i \in S, j \notin S$. This condition is extremely difficult to satisfy in practice. Therefore, in general, we have $q(U) < n \cdot \mu(U) + \text{negl}(\lambda)$. As $R \rightarrow +\infty$ (without considering termination), we have $\Pr[T] \rightarrow 0$, indicating that \mathcal{MG} delivers a response with probability approaching unity. In this asymptotic regime, the condition $q(U) \leq n \cdot \mu(U)$ remains valid, thereby ensuring the safety property.

3.3 Analysis

We provide a comprehensive comparison between RCS and existing CS method in Table 1. Below, we analyze each property in detail.

Latency. Let I denote the time required to sample y from a distribution p_i . In RCS, if no sample is accepted after R rounds, the algorithm incurs an additional cost of $n^2 \log n$ for trace computation. This cost is negligible compared with I , especially for reasoning LLMs (Chen et al., 2025). Therefore, RCS and CS have comparable time complexity.

Anti-Collusion. The goal of collusion is to make \mathcal{MG} more likely to deliver unsafe responses by manipulating one or more Byzantine models. We study the relation between acceptance probabilities of an unsafe response y_t and a safe response y_v within the R round loop of Algorithm 1. We focus on the quantity $\Delta(\sigma) = \sigma(y_t) - \sigma(y_v)$. The detailed derivation appears in Appendix B. The sign of $\Delta(\sigma)$ depends on:

$$\mathcal{D}_{t,v} = \frac{\sum_{i=1}^s p(i)(y_t) \sum_{i=s+1}^n p(i)(y_v)}{\sum_{i=1}^s p(i)(y_v) \sum_{i=s+1}^n p(i)(y_t)}.$$

When the Byzantine model assigns a very low probability to y_v , the value of $\sum_{i=1}^s p(i)(y_v)$ decreases, leading to $\mathcal{D}_{t,v} > 1$. As a result, $\Delta(\sigma) > 0$, which significantly increases the acceptance probability of the unsafe response y_t relative to y_v . This effect weakens the security of \mathcal{MG} . When

unsafe models collude, we present the resulting model probability distributions in Figure 2. The adversarial behavior of unsafe models clearly has a significant impact on the probability overlap among models in CS. In contrast, the design in lines 9-12 of Algorithm 1 effectively mitigates this issue. When the values of $p_{\text{unsafe}}(y_v)_j$ are very low, RCS can significantly downweight their influence on the final output response.

Half-Resilience. The CS algorithm has an abstention bound, meaning $\Pr[y = \perp]$ does not exceed a threshold when $f < \lceil \frac{n}{2} \rceil$. However, if this condition is violated, the CS abstention bound is affected, leading to an increased probability of abstention and a significant impact on liveness. RCS can avoid this issue. We further validate the half-resilience of RCS in the experiments in Section 5.

Termination. According to Algorithm 1, for a finite R , RCS always terminates within R rounds.

4 Feedback-Optimized Reliable Consensus Sampling

For CS and RCS, the framework does not require an explicit definition of security properties. This design allows algorithms to remain applicable over time. From this perspective, for each concrete task, one cannot determine the probability that the final delivered result is safe. This holds even though the algorithmic risk admits an upper bound. External control over the sampling process is limited. This implies that, from both an algorithmic and long-term operational perspective, if the \mathcal{MG} remains unchanged and the input task types are essentially fixed, the risk in RCS remains stationary over time, thereby limiting the potential for dynamic optimization. In this section, we analyze the nature of response safety in RCS. Inspired by quantum theory (Nielsen and Chuang, 2010), we propose a research methodology for RCS. We further introduce a mechanism that intervenes in model distributions and dynamically improves safety.

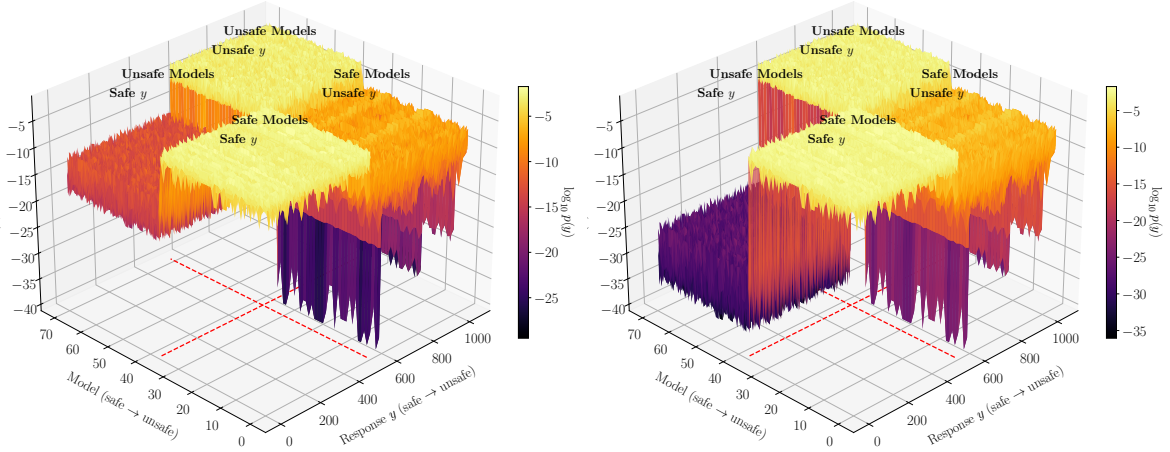
4.1 Foundational Theory

We propose a framework for studying the security of RCS by analogy with quantum theory, but the core analysis is fundamentally based on classical probability theory.

Foundational Theory. We model a system with unknown safety as a state $|\phi\rangle$, defined as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ denotes a safe state and $|1\rangle$ denotes an unsafe state. The constraint $\alpha^2 + \beta^2 =$

Algorithm	Safety	Liveness	Anti-Collusion	Half-Resilience	Termination	Time Complexity
Consensus Sampling	✓	✗	✗	✗	✓	$\mathcal{O}(RI)$
Reliable Consensus Sampling	✓	✓	✓	✓	✓	$[\mathcal{O}(RI), \mathcal{O}(RI + n^2 \log n)]$

Table 1: A comprehensive comparison between our proposed RCS and existing method.



(a) Distribution under general adversarial conditions.

(b) Distribution under malicious collusion.

Figure 2: Model probability distribution for consensus sampling under diverse adversarial environments.

1 holds. Before evaluation, $|\phi\rangle$ does not reside in a definite safe or unsafe state. It remains in a superposition, analogous to Schrödinger’s cat.

Given a fixed prompt q and an evaluation method e , we apply $\text{Eval}(q, |\phi\rangle, e)$. The state $|\phi\rangle$ then collapses to $|0\rangle$ or $|1\rangle$. Only at this stage can one determine safety under q and e . Therefore, $|\phi\rangle$ alone cannot be labeled as safe or unsafe. The state becomes concrete only under the joint action of q and e . The desired condition is $|\alpha| > |\beta|$, meaning a stronger bias toward the safe state.

Model-Response Correlation Analysis. Consider an \mathcal{MG} with n models $\{|\phi_1\rangle, |\phi_2\rangle, \dots, |\phi_n\rangle\}$. In each sampling round, the randomly selected model $|\phi_r\rangle$ remains in a superposition of safety and unsafety. However, the response y produced by $|\phi_r\rangle$ implicitly links $|\phi_r\rangle$ with other models. In \mathcal{MG} , there exists a subset $W = \{|\phi_1\rangle, |\phi_2\rangle, \dots, |\phi_\ell\rangle\}$ that assigns a high probability to y . There also exists a subset $X = \{|\phi_{\ell+1}\rangle, |\phi_{\ell+2}\rangle, \dots, |\phi_n\rangle\}$ that assigns a low probability to y . The model $|\phi_r\rangle$ belongs to neither W nor X . We say that W and X are entangled. After a time interval $\Delta(t)$, suppose evaluation under q and e verifies that $|\phi_r\rangle$ is unsafe. Then models in W are likely unsafe, while models in X are likely safe. In subsequent sampling, when a prompt x and an evaluation a resemble q and e , the algorithm should increase weights for models in X and decrease weights for models in W .

4.2 F-RCS Algorithm

Based on the foundational theory in Section 4.1, we construct an optimization algorithm for RCS, named Feedback-Optimized Reliable Consensus Sampling (F-RCS). In each sampling round, given the delivered response y and the generating model $|\phi_r\rangle$, the algorithm identifies two additional models by a function $F(\cdot)$ (see Algorithm 2). One model $|\phi_{\max}\rangle$ attains the maximum $p(y)$. Another model $|\phi_{\min}\rangle$ attains the minimum $p(y)$. If y is judged unsafe at time t , then for subsequent similar tasks, $|\phi_{\max}\rangle$ is removed during random sampling. Formally, $\mathcal{MG} = \mathcal{MG} \setminus \{|\phi_{\max}\rangle\}$. This allows, at the beginning of each algorithm execution, the system makes an automated decision. It selects safer models used in sampling for x . This process increases the safety margin $s - f$ and reduces risk. Here, $|\phi_{\max}\rangle$ can equal $|\phi_r\rangle$. Initially, $|\phi_r\rangle$ assigns the highest probability to y , which aligns with the algorithm objective. During sampling, the execution of the feedback algorithm is independent of RCS itself, and therefore does not affect the safety of RCS. For time overhead, the additional cost introduced by the feedback algorithm is negligible.

From a classical probability perspective, consider a model that generates a response for a given query. Several models may assign high support to this response. If the response is later evaluated as

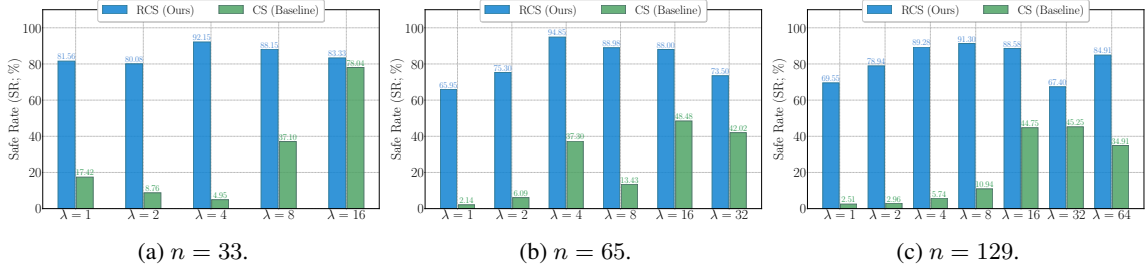


Figure 3: Evaluation results for the safe rate when $f < \lceil \frac{n}{2} \rceil$.

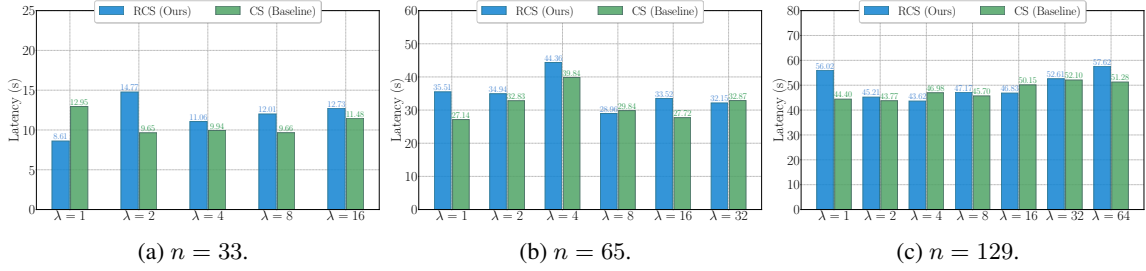


Figure 4: Evaluation results for the latency when $f < \lceil \frac{n}{2} \rceil$.

Algorithm 2: F-RCS Algorithm

Input: Parameters (n, s, R) ; evaluation method a

Output: Response y

- 1 **for** each prompt x **do**
- 2 $|\phi_r\rangle \leftarrow F(x, \mathcal{M}\mathcal{G})$
- 3 $\mathcal{M}\mathcal{G}' \leftarrow \mathcal{M}\mathcal{G} \setminus \{|\phi_r\rangle\}$
- 4 $\{y, \mathcal{P}\} \leftarrow \text{RCS}(\mathcal{M}\mathcal{G}', x, s, R)$
- 5 $\#\mathcal{P} = \{p_{|\phi_{\phi_1}\rangle}(y), \dots, p_{|\phi_{\phi_n}\rangle}(y)\}$
- 6 **#Identify correlated models.**
- 7 $|\phi_{\max}\rangle \leftarrow \arg \max_{|\phi_i\rangle} p_{|\phi_i\rangle}(y)$
- 8 **#Record** $\langle |\phi_{\max}\rangle, x, y, |\phi_r\rangle \rangle$.
- 9 **return** y
- 10 **while** time $t \in \mathbb{R}^+$ **do**
- 11 **if** $|1\rangle \leftarrow \text{Eval}(x, |\phi_r\rangle, a)$ **then**
- 12 New $F \leftarrow \text{Update}(F, |\phi_{\max}\rangle, x)$
- 13 $t \leftarrow t + \Delta(t)$

unsafe, the risk is not limited to the model that generated it. Other models that strongly support the same response may also exhibit unsafe behavior under this query. The agreement among models on an unsafe response suggests correlated risk.

5 Experiments

Although Section 3 presents a theoretical analysis that validates the advantages and effectiveness of our method, we still conduct extensive experiments to demonstrate its practical performance.

5.1 Experimental Setup

Parameter Settings. We follow the cryptography (Ruan et al., 2025; Couteau et al., 2022) for selecting the security parameter λ and set $\lambda = 2^d$, $d \in \mathbb{Z}$. We choose $R = \lambda + 1$. The values of s and f are determined according to specific types of experiments, and we set $n = kR + b$. From a practical perspective, the relationship between s and n is unknown. The value of s needs to be specified when deploying the algorithm. s does not reflect the actual number of safe models in the $\mathcal{M}\mathcal{G}$. To cover general practical scenarios, we adopt the weakest security assumption and set $s = \lceil \frac{n+1}{2} \rceil$. This setting is similar to the widely used $3f + 1 = n$ in distributed systems research (Zhang et al., 2023; Zhang and Duan, 2022).

Datasets and Models. To reflect adversarial conditions in real-world scenarios, we construct the probability distribution of the safe model using the output distributions of Qwen2.5-7b-instruct, Qwen2.5-0.5b-instruct (Yang et al., 2025), and Qwen3Guard-Gen-8B (Zhao et al., 2025) on safety evaluation datasets. The datasets include HarmBench (Mazeika et al., 2024) and AdvBench (Zou et al., 2023). We generate the perturbed probability distributions of the unsafe models by referring to the probability distribution of the safe model. This process simulates f Byzantine models that an adversary can fully control.

Evaluation Protocol. To reduce randomness, we report experimental results obtained from 8,000

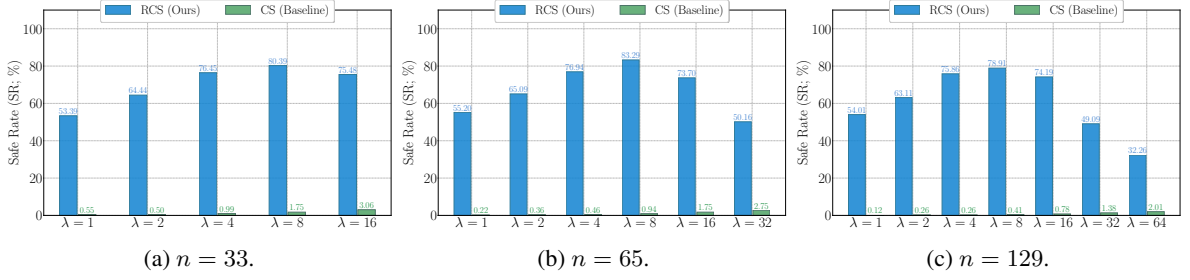


Figure 5: Evaluation results for the safe rate when Byzantine models collude.

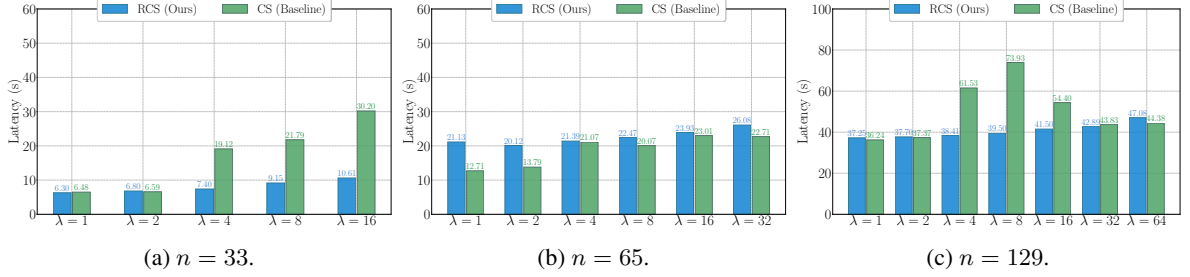


Figure 6: Evaluation results for the latency when Byzantine models collude.

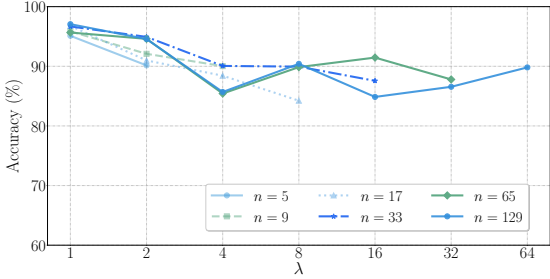


Figure 7: Accuracy of the feedback algorithm for different λ and n values under $f < \lceil \frac{n}{2} \rceil$.

repeated experiments.

5.2 Evaluation Metrics

In this work, we do not define specific criteria that label a response as safe or unsafe. For example, under jailbreak attacks, LLMs may output tokens such as "Sorry" or "Sure" (Wang et al., 2025c). These outputs reflect temporary experimental observations. They do not imply provable security in a theoretical sense. Therefore, our evaluation does not rely on the LLM-as-a-judge (Wei et al., 2025) commonly used in prior jailbreak studies. We define the evaluation metrics as follows. Safe rate (SR) denotes the proportion of final responses produced by the safe models. Abstention rate (AR) denotes the proportion of cases where the $\mathcal{M}\mathcal{G}$ returns \perp instead of a usable response. Latency denotes the time interval from the start of the sampling to the delivery of the final response. Accuracy measures the precision of the feedback algorithm in F-RCS when identifying unsafe models, that is,

the proportion of models detected by the feedback algorithm that are indeed unsafe.

5.3 Experimental Results

We evaluate settings where $f < \lceil \frac{n}{2} \rceil$. The results for large values of n are shown in Figure 3 and Figure 4. Results for smaller values of n are reported in Appendix C. For different values of the security parameter λ , RCS achieves significantly higher safety than CS. RCS maintains comparable overall latency to CS. These results are consistent with the theoretical analysis and proofs in Section 3. CS often leads to abstention, which reduces SR. We further evaluate the algorithms under Byzantine model collusion. Results for large values of n are shown in Figure 5 and Figure 6. Under collusion, the safety of CS degrades catastrophically compared to the $f < \lceil \frac{n}{2} \rceil$ setting. In contrast, the safety of RCS is only slightly affected. RCS maintains more than 67% safety. This result demonstrates strong resistance to collusion. To assess half-resilience, we also evaluate the case where $f = \lceil \frac{n}{2} \rceil$. The results are reported in Appendix C. In this case, CS again exhibits clear safety weaknesses and performs significantly worse than RCS. Table 2 reports aggregated results across all values of λ and n . RCS outperforms CS in both SR and AR. RCS also achieves comparable latency to CS. These results further validate the theoretical analysis and proofs.

For F-RCS, we evaluate the accuracy of the feedback algorithm in identifying unsafe models. The

Method	Consensus Sampling (Baseline)				Reliable Consensus Sampling (Ours)			
	$f < \lceil \frac{n}{2} \rceil$	$f = \lceil \frac{n}{2} \rceil$	Collusion	AVG	$f < \lceil \frac{n}{2} \rceil$	$f = \lceil \frac{n}{2} \rceil$	Collusion	AVG
Safe Rate	20.39	15.83	1.45	12.56	82.88	81.49	67.90	77.42 ($\uparrow \times 5.16$)
Abstention Rate	71.79	72.52	82.95	75.75	0.00	0.00	0.00	0.00 ($\downarrow 100\%$)
Latency	22.60	26.73	21.43	23.59	24.26	22.21	18.42	21.63 ($\downarrow 8.31\%$)

Table 2: Comprehensive comparison between our RCS and the baseline across different dimensions. Boldface indicates the best value in each dimension.

results are shown in Figure 7 and Appendix C. We show that for different values of λ and n , the average accuracy remains around 90%. This result demonstrates the effectiveness of the feedback algorithm.

6 Analysis and Discussion

6.1 Enhancements

Within the F-RCS framework, a general assumption is required. Consider the probability distributions produced by n models over an output y_h with unknown safety. After an arbitrary time interval $\Delta(t)$, if y_h is determined to be unsafe, this means that the model state $|\phi_h\rangle$ is unsafe. We require that the set of models W that assign high probability to y_h be trustworthy. Specifically, models in W cannot deny the probability distributions reported before $\Delta(t)$. They also cannot modify the reported distributions. Models in X cannot forge identities to impersonate models in W . These conditions can be enforced through cryptographic techniques such as blockchain (Li et al., 2023; Androulaki et al., 2018). Such solutions are isomorphic to existing security supervision mechanisms for voting (Yang et al., 2021). Concrete designs tailored to F-RCS remain an open direction for future work.

For time cost optimization, we can construct an intermediate algorithm by trading off CS and RCS. One such example is RCS from local coins (Zhang et al., 2023). After R unsuccessful sampling rounds, the algorithm flips a coin to decide whether to return \perp or to enter the trace phase. This design improves efficiency by sacrificing part of the safety. The detailed algorithm is provided in Appendix A.

6.2 Additional Insights on Optimal Safety

Based on experimental results, we observe a clear pattern. When n is fixed, the SR of RCS often exhibits a maximum for different λ . For example, Figure 5 shows a maximum at $\lambda = 8$. Very small values of λ and very large values of λ both fail to

achieve optimal safety. For a fixed n , it is unclear both how to determine whether such a maximum exists and how to theoretically identify the value λ_h corresponding to it.

6.3 Necessity Analysis of Liveness

We further analyze the necessity of the liveness property. In real-world deployments, the goal of \mathcal{MG} is not only to guarantee safety, but also to accomplish normal inference tasks such as mathematics, medical diagnosis, and programming. The objective is to maintain both functionality and safety, rather than prioritize safety as the sole consideration. As a result, in practical applications, \mathcal{MG} must be able to perform valid reasoning on a wide variety of prompts. From the perspective of safety, abstention can be used to avoid unsafe responses. However, for regular inference tasks, abstention is not acceptable, as the response \perp cannot adaptively provide reasonable outputs for unknown prompts. Therefore, liveness is necessary for \mathcal{MG} from a practical application perspective.

7 Related Work

Existing research on generative AI security mainly focuses on the safety of model outputs. This line of work studies various attack methods (Li et al., 2024, 2025b) and corresponding defenses (Zhang et al., 2025a). However, these attacks lack a unified definition of security. A representative example is the jailbreak attack (Mehrotra et al., 2024; Zhang et al., 2025b; Yu et al., 2024; Wang et al., 2025a). This attack aims to induce models to generate unsafe content. Such content covers multiple categories (Guo et al., 2025), including discrimination, illegal behavior, harmful behavior, and ethical issues. Output level unsafety is inherently subjective. This subjectivity prevents a deterministic definition under cryptographic theory. Current evaluation of output safety largely relies on LLM-as-a-judge (Li et al., 2025a). This approach depends on specialized evaluator models trained for safety analysis, such as Qwen3Guard (Zhao et al., 2025). These models

perform semantic analysis on generated content. However, evaluator models are not interpretable. Their correctness in safety assessment cannot be formally proven. Defenses against these attacks also lack formal guarantees. As a result, research on generative model security lacks a mature theoretical foundation. This absence of theory leads to inconsistent standards across existing attack and defense studies.

8 Conclusion

In this paper, we formalize a provable security theory for model groups. We introduce RCS, a trace-based method that eliminates abstention and guarantees effective delivery. RCS can tolerate extreme adversarial behavior. We also design a feedback algorithm to improve RCS safety. We prove that RCS has tight upper bounds on response risk. Experiments show that RCS outperforms CS in robustness and utility, achieving a $5\times$ increase in safety rate, while maintaining comparable latency.

9 Limitations

In this paper, we employ a safety threshold $f < \lceil \frac{n}{2} \rceil$ in traditional consensus sampling. Although this condition is feasible and justified in real-world deployments, exploring looser thresholds, such as $f < \lceil \frac{2n}{3} \rceil$, remains to be explored in future work. In addition, a comprehensive evaluation of RCS under large scale model groups is important for assessing throughput and resource consumption. Our evaluation is limited due to resource constraints. Currently, our experiments focus on classical security datasets. In the future, we aim to extend the evaluation to standard NLP generation tasks.

10 Ethical Considerations

Our discussion on unsafe models covers various adversarial behaviors of Byzantine models. However, more extreme threats warrant further study in future work. The Byzantine behavior discussed in this paper may be harmful. The methods described in this paper may be used for research purposes only. We use AI assistants to polish the writing.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant No. 62272043), Beijing Natural Science Foundation (Grant No. L251002), National Natural Science Foundation of China (Grant No. 62372040,

62572045), Natural Science Foundation of Beijing, China (Grant No. 4254086) and Yangtze Delta Region Institute of Tsinghua University, Zhejiang (No.LZZLX24F007).

References

- Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, and 2 others. 2018. [Hyperledger fabric: a distributed operating system for permissioned blockchains](#). In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA. Association for Computing Machinery.
- Sarah Ball, Greg Gluch, Shafi Goldwasser, Frauke Kreuter, Omer Reingold, and Guy N Rothblum. 2025. On the impossibility of separating intelligence from judgment: The computational intractability of filtering for ai alignment. *arXiv preprint arXiv:2507.07341*.
- Christian Cachin, Rachid Guerraoui, and Luís Rodrigues. 2011. *Introduction to reliable and secure distributed programming*. Springer Science & Business Media.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. [Chateval: Towards better LLM-based evaluators through multi-agent debate](#). In *The Twelfth International Conference on Learning Representations*.
- Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. [ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.
- Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024b. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213.
- Daiki Chijiwa, Taku Hasegawa, Kyosuke Nishida, Shin'ya Yamaguchi, Tomoya Ohba, Tamao Sakao, and Susumu Takeuchi. 2025. Lossless vocabulary

- reduction for auto-regressive language models. *arXiv preprint arXiv:2510.08102*.
- Geoffroy Couteau, Dahmun Goudarzi, Michael Kloob, and Michael Reichle. 2022. **Sharp: Short relaxed range proofs**. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22*, page 609–622, New York, NY, USA. Association for Computing Machinery.
- David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, and 1 others. 2024. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint arXiv:2405.06624*.
- Sourav Das, Sisi Duan, Shengqi Liu, Atsuki Momose, Ling Ren, and Victor Shoup. 2024. **Asynchronous consensus without trusted setup or public-key cryptography**. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, CCS '24*, page 3242–3256, New York, NY, USA. Association for Computing Machinery.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org.
- Sisi Duan, Haibin Zhang, Xiao Sui, Baohan Huang, Changchun Mu, Gang Di, and Xiaoyun Wang. 2024. **Dashing and star: Byzantine fault tolerance with weak certificates**. In *Proceedings of the Nineteenth European Conference on Computer Systems, EuroSys '24*, page 250–264, New York, NY, USA. Association for Computing Machinery.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Samyak Jain, Ekdeep S Lubana, Kemal Oksuz, Tom Joy, Philip Torr, Amartya Sanyal, and Puneet Dokania. 2024. What makes and breaks safety fine-tuning? a mechanistic study. *Advances in Neural Information Processing Systems*, 37:93406–93478.
- Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023a. Beavertails: towards improved safety alignment of llm via a human-preference dataset. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. **Towards mitigating LLM hallucination via self reflection**. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.
- Adam Tauman Kalai, Yael Tauman Kalai, and Or Zamir. 2025. Consensus sampling for safer generative ai. *arXiv preprint arXiv:2511.09493*.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhat-tacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025a. **From generation to judgment: Opportunities and challenges of LLM-as-a-judge**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.
- Huizhong Li, Yujie Chen, Xiang Shi, Xingqiang Bai, Nan Mo, Wenlin Li, Rui Guo, Zhang Wang, and Yi Sun. 2023. **Fisco-bcos: An enterprise-grade permissioned blockchain system with high-performance**. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '23*, New York, NY, USA. Association for Computing Machinery.
- Linbao Li, Yannan Liu, Daojing He, and YU LI. 2025b. **One model transfer to all: On robust jailbreak prompts generation against LLMs**. In *The Thirteenth International Conference on Learning Representations*.
- Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024. **Badedit: Backdooring large language models by model editing**. In *The Twelfth International Conference on Learning Representations*.
- Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. 2025. **Breaking mental set to improve reasoning through diverse multi-agent debate**. In *The Thirteenth International Conference on Learning Representations*.
- Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024. **Formalizing and benchmarking prompt injection attacks and defenses**. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1831–1847, Philadelphia, PA. USENIX Association.
- Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, and 1 others. 2025. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.
- Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. Harmbench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org.
- Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and

- Amin Karbasi. 2024. [Tree of attacks: Jailbreaking black-box llms automatically](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 61065–61105. Curran Associates, Inc.
- Michael A Nielsen and Isaac L Chuang. 2010. *Quantum computation and quantum information*. Cambridge university press.
- Anselm Paulus, Ilia Kulikov, Brandon Amos, Rémi Munos, Ivan Evtimov, Kamalika Chaudhuri, and Arman Zharmagambetov. 2025. Safety alignment of llms via non-cooperative games. *arXiv preprint arXiv:2512.20806*.
- Wenqiang Ruan, Xin Lin, Ruisheng Zhou, Guopeng Lin, Shui Yu, and Weili Han. 2025. Hawkeye: statically and accurately profiling the communication cost of models in multi-party learning. In *Proceedings of the 34th USENIX Conference on Security Symposium, SEC '25, USA*. USENIX Association.
- Hao Wang, Hao Li, Junda Zhu, Xinyuan Wang, Chengwei Pan, Minlie Huang, and Lei Sha. 2025a. [DiffusionAttacker: Diffusion-driven prompt manipulation for LLM jailbreak](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22193–22205, Suzhou, China. Association for Computational Linguistics.
- Shang Wang, Tianqing Zhu, Bo Liu, Ming Ding, Dayong Ye, Wanlei Zhou, and Philip Yu. 2025b. [Unique security and privacy threats of large language models: A comprehensive survey](#). *ACM Comput. Surv.* Just Accepted.
- Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2025c. [Vulnerability of large language models to output prefix jailbreaks: Impact of positions on safety](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3939–3952, Albuquerque, New Mexico. Association for Computational Linguistics.
- Zhipeng Wei, Yuqi Liu, and N. Benjamin Erichson. 2025. [Emoji attack: Enhancing jailbreak attacks against judge LLM detection](#). In *Forty-second International Conference on Machine Learning*.
- Yuxin Wen, Leo Marchyok, Sanghyun Hong, Jonas Geiping, Tom Goldstein, and Nicholas Carlini. 2024. Privacy backdoors: Enhancing membership inference through poisoning pre-trained models. *Advances in Neural Information Processing Systems*, 37:83374–83396.
- Zhiyuan Weng, Guikun Chen, and Wenguan Wang. 2025. [Do as we do, not as you think: the conformity of large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. [Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3111–3126, Mexico City, Mexico. Association for Computational Linguistics.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jixi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. [Qwen2.5 technical report](#). *Preprint, arXiv:2412.15115*.
- Yang Yang, Zhangshuang Guan, Zhiguo Wan, Jian Weng, Hwee Hwa Pang, and Robert H Deng. 2021. Priscore: Blockchain-based self-tallying election system supporting score voting. *IEEE Transactions on Information Forensics and Security*, 16:4705–4720.
- Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. 2024. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4675–4692.
- Xiao Zhan, Juan Carlos Carrillo, William Seymour, and Jose Such. 2025. Malicious llm-based conversational ai makes users reveal personal information. In *Proceedings of the 34th USENIX Conference on Security Symposium, SEC '25, USA*. USENIX Association.
- Haibin Zhang and Sisi Duan. 2022. [Pace: Fully parallelizable bft from repropoasable byzantine agreement](#). In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22*, page 3151–3164, New York, NY, USA. Association for Computing Machinery.
- Haibin Zhang, Sisi Duan, Boxin Zhao, and Liehuang Zhu. 2023. Waterbear: practical asynchronous bft matching security guarantees of partially synchronous bft. In *Proceedings of the 32nd USENIX Conference on Security Symposium, SEC '23, USA*. USENIX Association.
- Lan Zhang, Xinben Gao, Liuyi Yao, Jinke Song, and Yaliang Li. 2025a. [Exploiting {Task-Level} vulnerabilities: An automatic jailbreak attack and defense benchmarking for {LLMs}](#). In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2363–2382.
- Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. 2025b. Jbshield: defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *Proceedings of the 34th USENIX Conference on Security Symposium, SEC '25, USA*. USENIX Association.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, and 1 others. 2025. Qwen3guard technical report. *arXiv preprint arXiv:2510.14276*.

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

A RCS from Local Coins

Algorithm 3: Reliable Consensus Sampling from Local Coins

Input: Number of models $|\mathcal{M}\mathcal{G}| = n$; number of safe models s ; round R ; distributions

$$p_1, \dots, p_n \in \text{Distr}(\mathcal{Y})^n$$

Output: Response $y \in \mathcal{Y}$

```

1 Buffer  $\leftarrow \emptyset$ 
2 for  $r \leftarrow 1$  to  $R$  do
3   Sample  $y \sim \frac{1}{n} \sum_{i=1}^n p_i$ 
4    $\sigma(y) = \frac{\frac{1}{s} \sum_{i=1}^s p_i(y)}{\frac{1}{n} \sum_{i=1}^n p_i(y)}$ 
5   if accept  $y$  with  $\sigma(y)$  then
6     | return  $y$ 
7     | Buffer  $\leftarrow$  Buffer  $\cup \{ \langle y, \sigma(y) \rangle \}$ 
8  $c \leftarrow \text{Random}()$  #Obtain local coin 0 or 1.
9 if  $c == 0$  then
10  | return  $\perp$ 
11 else
12  | Sort Buffer =  $\{ \langle y_i, \sigma(y_i) \rangle \}_{i=1}^R$  such that  $\sigma(y_{(1)}) \geq \dots \geq \sigma(y_{(R)})$ 
13  | Let  $\mathcal{F} = \{y_{(1)}, \dots, y_{(u)}\}$ ,  $u = \min(s, R)$ 
14  | for  $i \leftarrow 1$  to  $u$  do
15  |    $\alpha(y_{(i)}) = \sum_{j=s+1}^n p_j(y_{(i)})$ 
16  | return  $y \leftarrow \arg \max_{y \in \mathcal{F}} \alpha(y)$ 

```

B Proof

$$\begin{aligned}
& \sigma(y_t) - \sigma(y_v) \\
&= \frac{\frac{1}{s} \sum_{i \leq s} p_i(y_t)}{\frac{1}{n} \sum_{i=1}^n p_i(y_t)} - \frac{\frac{1}{s} \sum_{i \leq s} p_i(y_v)}{\frac{1}{n} \sum_{i=1}^n p_i(y_v)} \\
&= \frac{\left(\frac{1}{s} \sum_{i \leq s} p_i(y_t) \right) \left(\frac{1}{n} \sum_{i=1}^n p_i(y_v) \right) - \left(\frac{1}{s} \sum_{i \leq s} p_i(y_v) \right) \left(\frac{1}{n} \sum_{i=1}^n p_i(y_t) \right)}{\left(\frac{1}{n} \sum_{i=1}^n p_i(y_t) \right) \left(\frac{1}{n} \sum_{i=1}^n p_i(y_v) \right)} \\
&= \frac{n}{s} \cdot \frac{\left(\sum_{i \leq s} p_i(y_t) \right) \left(\sum_{i=1}^n p_i(y_v) \right) - \left(\sum_{i \leq s} p_i(y_v) \right) \left(\sum_{i=1}^n p_i(y_t) \right)}{\left(\sum_{i=1}^n p_i(y_t) \right) \left(\sum_{i=1}^n p_i(y_v) \right)} \\
&= \frac{n}{s} \cdot \frac{\left(\sum_{i \leq s} p_i(y_t) \right) \left(\sum_{i > s} p_i(y_v) \right) - \left(\sum_{i \leq s} p_i(y_v) \right) \left(\sum_{i > s} p_i(y_t) \right)}{\left(\sum_{i=1}^n p_i(y_t) \right) \left(\sum_{i=1}^n p_i(y_v) \right)}
\end{aligned}$$

C Additional Experimental Results

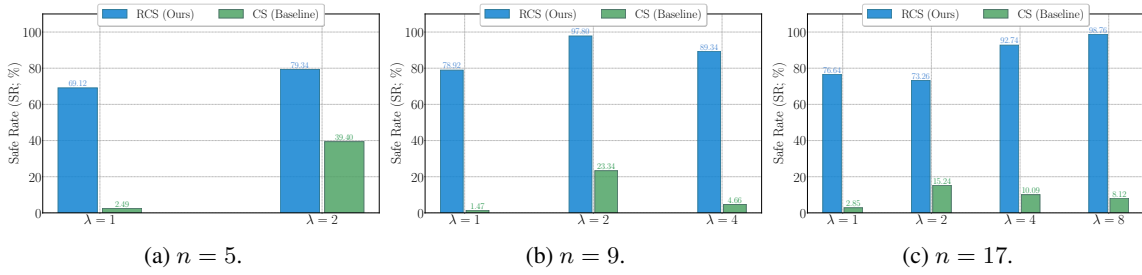


Figure 8: Evaluation results for the safe rate when $f < \lfloor \frac{n}{2} \rfloor$.

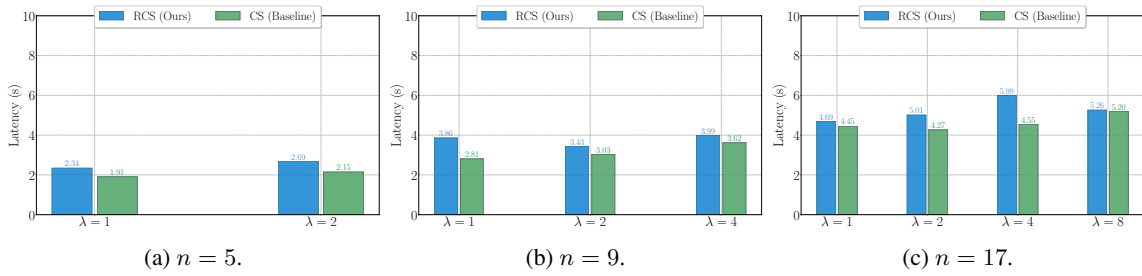


Figure 9: Evaluation results for the latency when $f < \lfloor \frac{n}{2} \rfloor$.

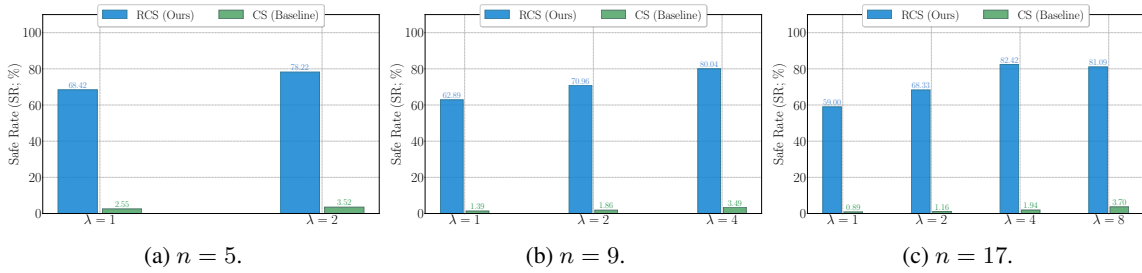


Figure 10: Evaluation results for the safe rate when Byzantine models collude.

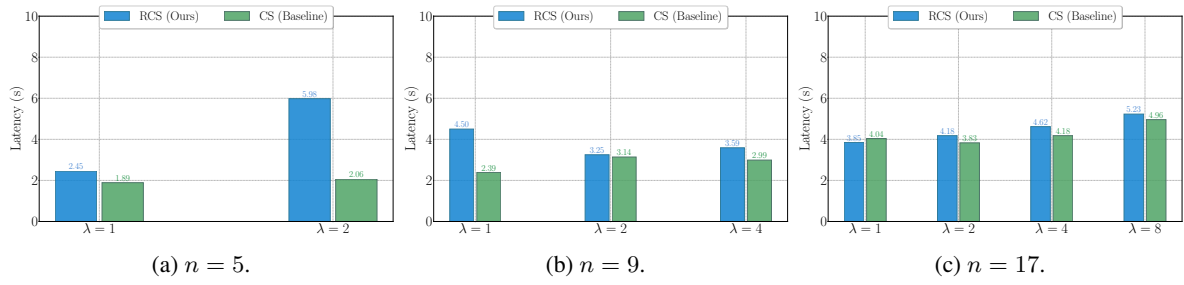


Figure 11: Evaluation results for the latency when Byzantine models collude.

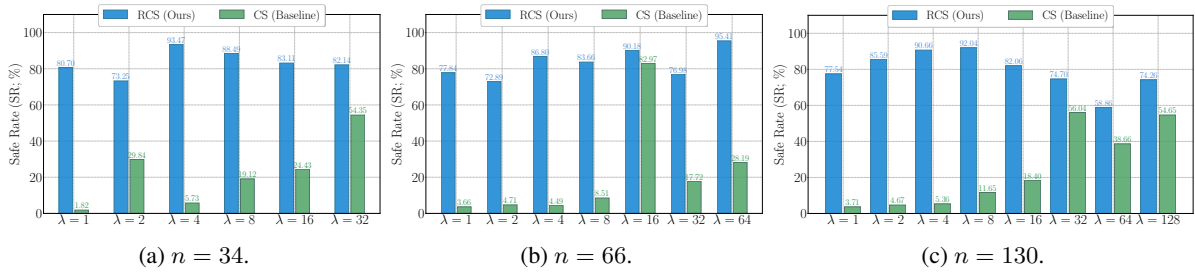


Figure 12: Evaluation results for the safe rate when $f = \lfloor \frac{n}{2} \rfloor$.

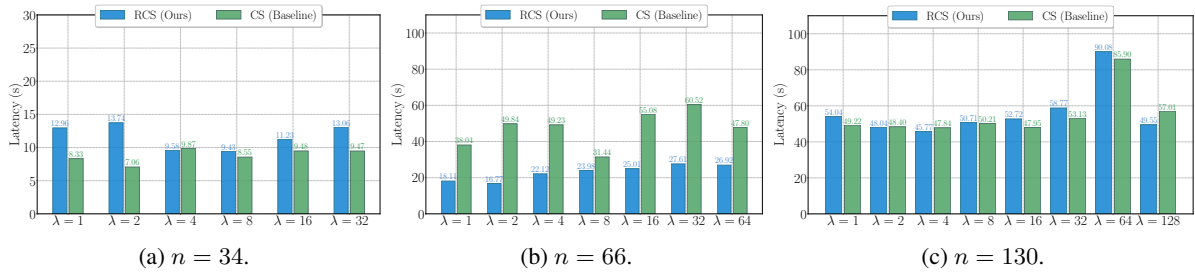


Figure 13: Evaluation results for the latency when $f = \lfloor \frac{n}{2} \rfloor$.

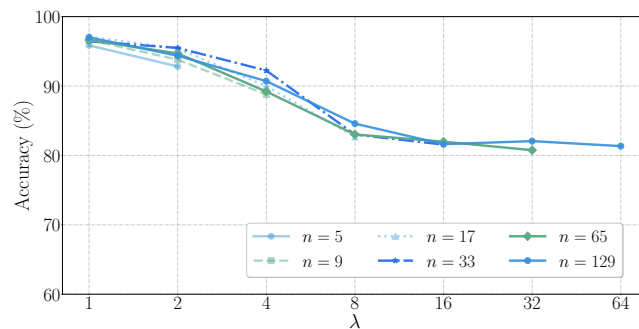


Figure 14: Accuracy of the feedback algorithm for different λ and n values when Byzantine models collude.

D Additional Discussion

D.1 RCS and Multi-Agent Debate

Multi-agent debate (MAD) (Du et al., 2024; Chan et al., 2024; Liu et al., 2025) is a reasoning enhancement technique built on LLM inference. MAD enables multiple agents to interact over several rounds. The goal is to encourage consensus on the same answer to a given query. A final decision is then produced through answer aggregation methods such as majority voting (Du et al., 2024). In terms of output safety, MAD and RCS share a similar motivation. Both approaches introduce multiple models to improve the safety of the final output. Some MAD methods also adopt confidence scores to evaluate candidate answers proposed by other models (Chen et al., 2024a). This mechanism is conceptually related to RCS. Compared to MAD, RCS follows a more fundamental strategy. RCS directly operates on underlying probability distributions. In contrast, MAD requires models to possess an explicit understanding of safety. This requirement limits robustness against unforeseen attacks. Moreover, RCS does not require interaction among models. This design avoids the conformity (Weng et al., 2025) inherent in LLMs. Operating on underlying probability distributions also mitigates the impact of hallucination (Ji et al., 2023b) during reasoning. However, this design limits practical applicability. In many scenarios, models are accessed as black-box services, and probability distributions are not available. We argue that an intersection between MAD and RCS can improve the practical applicability of RCS. We continue to avoid direct interaction among models. Instead, each model outputs a score for each candidate output y at every round. This score represents the level of support for y . All other components remain consistent with RCS. We refer to this variant as Practical Reliable Consensus Sampling (PRCS). More broadly, PRCS can be applied beyond safety. It can also enhance model reasoning performance in domains such as mathematics, medicine, and programming (Luo et al., 2025).