PRISM: Calibrated Prior-Risk Integration for Safe and Useful Test-Time Alignment

Yiqi Deng[1], Ruixi Liu[1], Jiahao Hu[2], Guoai Xu[1]*,

[1]Harbin Institute of Technology, Shenzhen, China
[2]The University of Hong Kong, Hong Kong SAR, China
dengyiqi@hit.edu.cn, 2023311f29@stu.hit.edu.cn
hujh0919@gmail.com, xga@hit.edu.cn

Abstract

This paper investigates the problem of safety alignment in large language models (LLMs) during the testing phase, particularly under jailbreaking attacks. Previous methods typically rely on identifying malicious request intentions or adjusting the LLM's decoding strategy to inject safe behavior. However, this often leads to over-rejection, limiting their practical application. To address this, we propose an innovative framework called Prior-Risk Integration for Safe Mixture decoding (PRISM). Specifically, PRISM first uses a lightweight semantic risk evaluator to calculate the prior probability of harmfulness at the prompt level, and then continuously tracks the cumulative risk signal provided by a safety evaluator during the generation process. By comprehensively considering these two types of risks, PRISM adaptively mixes the base model's probability distribution with a safety preference distribution, triggering a rejection response only when both the prompt prior probability and the generation risk are sufficiently high. Extensive experiments validate the effectiveness of PRISM in resisting jailbreaking attacks and its practicality. Across five jailbreaking attack methods and three neutral datasets, PRISM demonstrates superior performance compared to other baseline methods.

1 Introduction

In recent years, Large Language Models (LLMs) (Brown et al., 2020; Naveed et al., 2025) have achieved remarkable breakthroughs across a wide range of natural language processing tasks, from text generation and summarization to complex reasoning and open-domain dialog systems. Models such as ChatGPT (Achiam et al., 2023), Vicuna (Chiang et al., 2023), and LLaMA (Touvron et al., 2023) have demonstrated unprecedented capabilities in understanding and generating human-like text, driving innovations in both academic research and industrial applications. However, as these models become increasingly integrated into real-world systems, their security and robustness have emerged as critical concerns that must be addressed to ensure trustworthy applications.

Despite their impressive performance, LLMs are vulnerable to various security threats, including prompt injection (Greshake et al., 2023; Li et al., 2024), data leakage (Kaneko and Baldwin, 2025; Ramrakhiyani et al., 2025), and adversarial attacks (Raina et al., 2024; Yao et al., 2024). Among these, jailbreak attacks (Zou et al., 2023; Wei et al., 2026; Chao et al., 2025) have garnered significant attention due to their potential to subvert model safeguards and elicit harmful, biased, or otherwise restricted content. Jailbreak attacks refer to the deliberate crafting of adversarial prompts—often through semantic manipulation (Wei et al., 2026; Zou et al., 2023), role-playing scenarios (Liu et al., 2023; Chao et al., 2025), or encoded instructions (Huang et al., 2024; Mehrotra et al., 2024) that bypass the model's alignment mechanisms and safety guidelines. Such attacks not only compromise integrity of LLM outputs but also raise ethical concerns regarding the misuse of AI technologies.

To mitigate jailbreak risks, several defense approaches have been proposed, which can be broadly categorized into: (1) input filtering and sanitization (Phute et al., 2023; Wei et al., 2026), which detect and block malicious prompts based on predefined rules or classifiers; (2) robust training and alignment, which strengthen model resilience through adversarial fine-tuning (Ouyang et al., 2022) or reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020); and (3) runtime monitoring and perturbation (Xie et al., 2023; Zhang et al., 2023), which modify user inputs or employ auxiliary models to detect and neutralize adversarial intent. While these methods enhance safety, they often introduce an unintended side effect of over-refusal (Cui et al., 2024), wherein

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 25128–25144
October 24-29, 2026 ©2026 Association for Computational Linguistics

LLMs reject benign prompts and consequently become less useful.

However, mitigating over-refusal is challenging because test-time safety decisions must distinguish genuinely harmful intent from benign queries that only appear risky. Existing defenses often rely on static or localized safety signals, making them insufficiently adaptive to contextual semantics and generation dynamics. Input filtering (Phute et al., 2023; Jain et al., 2023) and rule-based approaches (Wei et al., 2026) often reject queries containing sensitive keywords regardless of legitimate context, while adversarial training methods (Ouyang et al., 2022; Stiennon et al., 2020) may cause models to become overly conservative, refusing to engage with ambiguous but benign inputs. Furthermore, single-step safety evaluations (Xie et al., 2023; Zhang et al., 2023) ignore the cumulative nature of conversational harm, leading to premature rejection of multi-turn interactions. Our proposed method addresses these limitations by introducing a two-stage safety mechanism. First, a semantic risk estimator computes a prompt-level harmful prior from the user input. Second, a prior-guided adaptive decoder uses this prior to continuously monitor a cumulative risk signal during generation. When both the prompt is likely harmful and the generated content becomes unsafe, we trigger a refusal. This design aims to avoid rejecting benign prompts that merely contain sensitive keywords, while still blocking genuine jailbreak attempts where both the input and output are dangerous.

In this paper, we propose a novel defense framework designed to address the over-rejection problem while maintaining high utility against jailbreak attacks. Our method integrates a multi-stage verification mechanism that combines lightweight prompt analysis, context-aware safety scoring, and controlled response generation. Different from prior work that relies on static, binary safety classification (Jain et al., 2023; Robey et al., 2023), our approach employs a dynamic threshold adjustment strategy based on query context and user intent, allowing for more nuanced and adaptive safety decisions. Our contributions are threefold:

- **Novel Perspective.** Explicitly treating jailbreak robustness and over-rejection mitigation as a coupled design goal, introducing a dynamic, fine-grained safety control mechanism at the decoding level.
- **Adaptive Safety Decoding Framework.** Proposing a test-time safety alignment framework with two key components: (1) a semantic risk estimator that computes a harmful prior for the user prompt; (2) a prior-guided decoder using this prior to monitor cumulative risk during generation, triggering refusal only when both prompt and generated content are dangerous, thereby dynamically balancing LLM safety and usability.
- **Empirical Validation.** Conducting extensive experiments on multiple jailbreak benchmarks, evaluating both robustness against state-of-the-art attacks and over-rejection rates on neutral queries. Results demonstrate a more favorable safety-usability trade-off compared to existing methods.

2 Related Work

2.1 Jailbreak Attacks

Jailbreak attacks bypass LLM safety alignment to elicit harmful content. Early methods used manually crafted adversarial suffixes (Liu et al., 2023; Yu et al., 2023). Subsequent work introduced gradient-based optimization, most notably GCG (Zou et al., 2023), which iteratively replaces tokens to maximize harmfulness. To evade perplexity-based filters, AutoDAN (Liu et al., 2024b) employs genetic algorithms to generate semantically coherent yet adversarial prompts. Beyond single-turn optimization, PAIR (Chao et al., 2025) uses an attacker LLM to interactively refine prompts, while MultiLingual (Deng et al., 2024) attacks exploit weaker safety alignment in low-resource languages. ICA (Wei et al., 2026) demonstrates that in-context demonstrations with harmful examples can effectively jailbreak aligned models. These diverse attack paradigms highlight the need for robust and adaptive defenses.

2.2 Test-time Defenses

Test-time defenses operate during inference, without modifying LLM's parameters or requiring additional fine-tuning. These methods contrast with training-time defenses, e.g., adversarial fine-tuning (Bach et al., 2026) and RLHF (Ouyang et al., 2022), offering greater flexibility.

Existing test-time defenses fall into two categories. Input-level defenses detect and neutralize adversarial patterns before generation. Among these, SmoothLLM (Robey et al., 2023) perturbs the input multiple times and aggregates outputs to

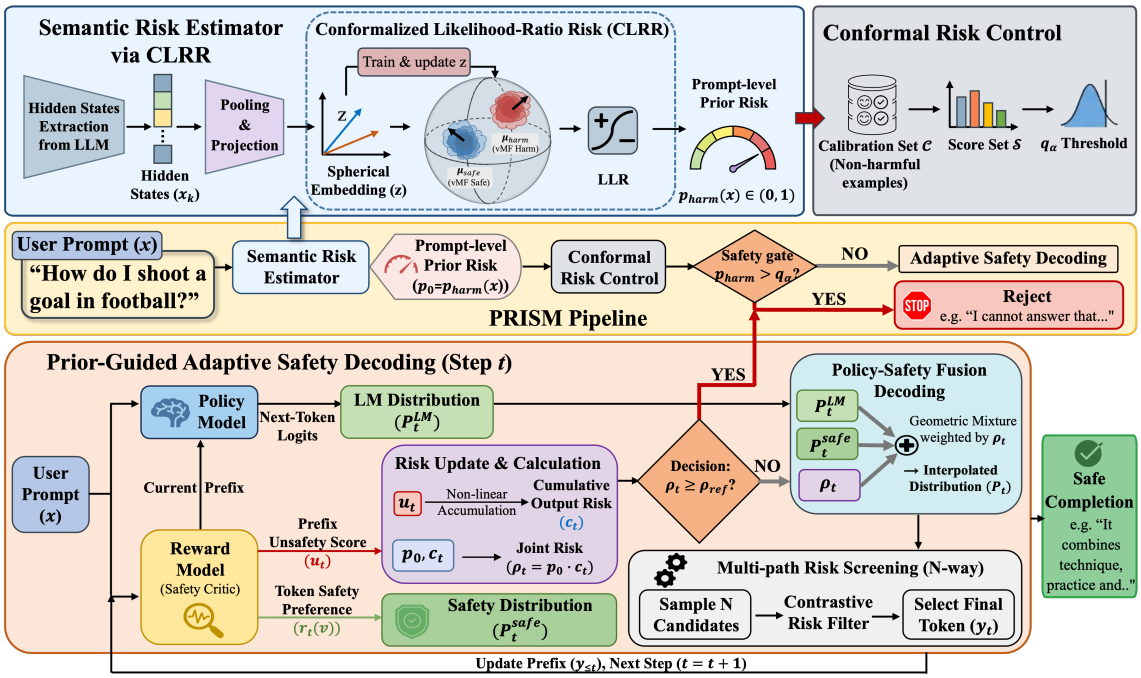


Figure 1: Overview of the proposed PRISM framework. It comprises a Semantic Risk Estimator (left) that computes a harmful prior p_0 from the user prompt via CLRR, and an Adaptive Safety Decoder (right) that combines p_0 with reward model signals to guide token selection. Decoding is controlled by a joint risk $\rho_t = p_0 \cdot c_t$, triggering a hard refusal only when ρ_t exceeds a threshold.

dilute adversarial triggers. Retokenization (Jain et al., 2023) disrupts token sequences to break optimized jailbreak suffixes. AED (Liu et al., 2024a) refines the next-token distribution by detecting alignment conflicts at the token level. DPP (Xiong et al., 2025) incorporates safety instructions directly into the user query to reinforce model behavior. Self-reminder (Xie et al., 2023) adds safety reminders to the input context, leveraging the model's instruction-following capability to encourage safe outputs. Decoding-level defenses directly guide the generation process. Saffron (Qiu et al., 2025) uses reward-guided search to select safer responses. R2D (Zhu et al., 2025) incorporates explicit safety reasoning into the generation chain-of-thought.

While these defenses improve robustness, they often suffer from over-refusal, incorrectly rejecting benign or ambiguous queries. This motivates our work, which introduces a dynamic, risk-aware decoding framework that triggers refusal only when both prompt-level prior and cumulative generation risk are high, thereby preserving utility without compromising safety.

3 Methodology

Test-time safety defenses often rely on a fixed rejection threshold, creating an inherent trade-off: a strict threshold reduces Attack Success Rate (ASR) but increases False Positive Rate (FPR) by over-refusing benign inputs. To mitigate this, we propose **PRISM**, a framework that dynamically controls LLM generation to prevent harmful outputs while minimizing over-refusal. Our method consists of two components: (1) a **Semantic Risk Estimator** that computes a harmfulness prior from the input prompt via conformalized likelihood-ratio risk (CLRR); (2) a **Prior-Guided Adaptive Safety Decoder (PGASD)** that uses the prior to adaptively tune the downstream rejection thresholds and guide token selection during generation. The goal is to reduce FPR on benign/neutral prompts while preserving low ASR on harmful prompts. Figure 1 illustrates the overall architecture.

3.1 Problem Formulation

Given a user prompt x, a standard decoding policy π_{base} produces a response $y \sim \pi_{\text{base}}(\cdot \mid x)$ by sampling the next-token distribution at each step. We seek a *safety-aware decoding policy* π^* that minimizes two competing objectives:

$$\pi^* \in \arg\min_{\pi} \big(\text{ASR}(\pi),\ \text{FPR}(\pi)\big), \qquad (1)$$

where ASR is the probability that π^* generates harmful content when x is a malicious prompt, and

FPR is the probability of refusing a benign prompt.

3.2 Semantic Risk Estimator

The semantic risk estimator produces a harmfulness prior $p_{\text{harm}}(x) \in (0,1)$ for any user prompt x.

Prompt representation. Given x, we format it with the chat template $\tilde{x} = \text{prefix}\|x\|\text{suffix}$, tokenize it into $t_{1:L}$ of length L, and extract the first-layer hidden states $\mathbf{H}(\tilde{x}) = [\mathbf{h}_1, \ldots, \mathbf{h}_L] \in \mathbb{R}^{L\times d}$ with a hidden dimension d via a single forward pass. Extracting hidden states from the first layer is computationally efficient and retains sufficient semantic information for detecting harmfulness. We then standardize sequences to a fixed length K via left-padding, yielding $\mathbf{X} \in \mathbb{R}^{K\times d}$ with each row $\mathbf{x}_k \in \mathbb{R}^d$ corresponding to the k-th token.A mask vector $\mathbf{m} \in \{0,1\}^K$ with $m_k = 1$ indicating a real token and $m_k = 0$ meaning padding.

Gated attention pooling and projection. We aggregate token-level representations via a gated attention mechanism. For each token k, we compute a gate score s_k that measures each token's contribution to the overall semantics as:

$$s_k = \mathbf{w}^\top \tanh(\mathbf{W}_g \mathbf{x}_k) \in \mathbb{R} \tag{2}$$

where $\mathbf{W}_g \in \mathbb{R}^{h\times d}$ and $\mathbf{w} \in \mathbb{R}^h$ are learnable parameters. Attention weights is obtained through $\alpha_k = \frac{\exp(s_k)\cdot m_k}{\sum_{j=1}^K \exp(s_j)\cdot m_j}$ with a mask m_k excluding padding positions. Later, we obtain a pooled representation $\mathbf{p}$ focusing on semantically critical tokens (e.g., harmful instructions) while ignoring irrelevant ones:

$$\mathbf{p} = \sum_{k=1}^{K} \alpha_k \mathbf{x}_k \in \mathbb{R}^d \tag{3}$$

Conformalized Likelihood-Ratio Risk (CLRR). Following previous works (Xu and Durrett, 2018; Loshchilov et al., 2025), we project $\mathbf{p}$ onto the unit hypersphere to obtain a compact, semantically meaningful embedding for subsequent density estimation:

$$\mathbf{z} = \frac{\mathbf{f}_{\text{proj}}(\mathbf{p})}{\|\mathbf{f}_{\text{proj}}(\mathbf{p})\|_2} \in \mathbb{R}^D. \tag{4}$$

where $\mathbf{f}_{\text{proj}}(\cdot)$ is a learnable multi-layer perceptron, and $D \ll d$. For class $c \in \{\text{safe}, \text{harm}\}$ with "$safe$" aggregating benign/neutral samples, we assume

$$p(\mathbf{z} \mid y = c) = C_D(\kappa_c) \exp\left(\kappa_c \boldsymbol{\mu}_c^\top \mathbf{z}\right), \tag{5}$$

as a von Mises-Fisher (vMF) distribution (Mardia and Jupp, 2009) with mean direction $\boldsymbol{\mu}_c$ ($\|\boldsymbol{\mu}_c\|_2 = 1$), concentration $\kappa_c > 0$, $C_D(\cdot)$ as the vMF normalizer. A larger κ_c indicates a closer concentration of the distribution around the mean direction. When κ_c = 0, the distribution becomes uniform. This yields a log-likelihood ratio (LLR) statistic:

$$LLR(x) = \log \frac{p(\mathbf{z}(x) \mid y = \text{harm})}{p(\mathbf{z}(x) \mid y = \text{safe})} \tag{6}$$

$$= (\kappa_{\text{harm}}\boldsymbol{\mu}_{\text{harm}} - \kappa_{\text{safe}}\boldsymbol{\mu}_{\text{safe}})^\top \mathbf{z}(x) \tag{7}$$

$$+ \log \frac{C_D(\kappa_{\text{harm}})}{C_D(\kappa_{\text{safe}})}. \tag{8}$$

With Bayesian rule, the posterior harmful probability becomes

$$p_{\text{harm}}(x) = \sigma\left(\text{LLR}(x) + \log \frac{\pi_{\text{harm}}}{\pi_{\text{safe}}}\right) \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function, $\pi_c = P(y = c)$ are the class priors. The resulting $p_{\text{harm}}(x) \in (0,1)$ serves as the prompt-level risk prior for later adaptive safety control. A detailed derivation of Eq. (9) is provided in Appendix E.

Training objective. We train the gating-and-projection network to produce discriminative spherical embeddings $\mathbf{z}(x)$ for CLRR. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a labeled dataset where $y_i \in \{\text{safe}, \text{neutral}, \text{harm}\}$. We map *neutral* to the non-harm side for binary risk modeling and optimize a prototype-based cross-entropy loss:

$$\mathcal{L}_{\text{proto}} = -\sum_{i=1}^{N} \log \frac{\exp(\mathbf{z}(x_i)^\top \boldsymbol{\mu}_{y_i}/\tau)}{\sum_{c\in\{\text{safe},\,\text{harm}\}} \exp(\mathbf{z}(x_i)^\top \boldsymbol{\mu}_c/\tau)} \tag{10}$$

where τ is the temperature. This loss encourages same-class embeddings to be close on the sphere while separating different classes. We train on 2k benign, 2k neutral, and 1k harmful prompts, achieving 97% held-out accuracy.

Conformal risk control. To control over-refusal, we conformalize $p_{\text{harm}}(x)$ using a calibration set. Let $\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{C}|}$ be a held-out calibration set consisting only of benign/neutral examples (i.e., non-harmful), where x_i is the i-th input prompt and $y_i \in \{\text{safe}, \text{neutral}\}$ is its label. For each non-harm example $x_i \in \mathcal{C}$, we collect scores $s(x_i) = p_{\text{harm}}(x_i)$ into the set:

$$\mathcal{S} = \{s(x_i) : (x_i, y_i) \in \mathcal{C},\ y_i \in \{\text{safe}, \text{neutral}\}\} \tag{11}$$

For a target false positive budget $\alpha \in (0,1)$, we define the $(1-\alpha)$ empirical quantile of $\mathcal{S}$ as:

$$q_\alpha = \text{Quantile}_{1-\alpha}(\mathcal{S}) \tag{12}$$

Under exchangeability, this construction ensures

$$\mathbb{P}(s(X) > q_\alpha \mid Y \in \{\text{safe, neutral}\}) \leq \alpha \tag{13}$$

This suggests that for a new non-harmful input x, the probability that its risk score exceeds q_α is at most α. We therefore use q_α as a safety gate: if $p_{\text{harm}}(x) > q_\alpha$, we reject or trigger enhanced safety; otherwise, normal processing proceeds.

3.3 Prior-Guided Adaptive Safety Decoding (PGASD)

We use the semantic risk estimator as a prompt-level prior and couple it with a generation-time cumulative risk signal from a reward model to drive continuous safety control. Instead of selecting the safest token at every step, we fuse the policy likelihood with a safety preference distribution. When the prefix remains safe, decoding behaves close to the original policy; when risk accumulates, the sampling distribution smoothly shifts toward safer candidates. A hard refusal is triggered only when both the input prior and the output cumulative risk are high.

Setup. Given a user prompt x, the semantic risk estimator outputs a harmful prior:

$$p_0 = p_{harm}(x) \in (0,1) \tag{14}$$

Safety signals. Let the policy model define next-token logits $\ell_\theta(v|x, y<t)$ over vocabulary $v \in \mathcal{V}$ at step t, with $\pi_t(v) = softmax(\ell_\theta(\cdot|x, y_{<t}))_v$, we define a prefix-level unsafety score $u_t = U(x, y_{\leq t}) \in [0,1]$ and a token-level safety score $r_t(v) \approx R(x, y_{<t}v)$, where U and R are reward models, e.g., Llama-Guard-1B. u_t indicates how unsafe the current prefix is, with higher u_t corresponding to more unsafe contents. $r_t(v)$ evaluates the safety of a specific candidate token v in context of the current generation ($y \leq t$) history, with a larger $r_t(v)$ indicates a safer token.

Cumulative output risk. We maintain cumulative risk $c_t \in [0,1]$ via log-odds accumulation. Specifically, we initialize $c_0 = 0$ and update it as follows:

$$\ell_0 = logit(c_0), \quad c_0 = 0 \tag{15}$$

$$\ell_t = \ell_{t-1} + logit(\text{clip}(u_t, \epsilon, 1-\epsilon)) \tag{16}$$

$$c_t = \sigma(\ell_t) \tag{17}$$

where $\text{logit}(z) = \log \frac{z}{1-z}$. $\sigma(\cdot)$ is the sigmoid. ϵ is a small constant (e.g., 10^{-6}) introduced to avoid numerical instability when u_t approaches 0 or 1. clip is the clipping operation where $\text{clip}(\mu, \epsilon, 1-\epsilon) = \epsilon$ or μ or $1-\epsilon$ if $\mu < \epsilon, \epsilon < \mu < 1-\epsilon$, or $\mu > 1-\epsilon$, respectively. This non-linear update saturates c_t toward 1 under persistent unsafe steps while forgetting occasional noise.

Joint risk and hard refusal. We define the joint risk ρ_t as the product of the input prior in Eq.(14) and cumulative output risk in Eq.(15):

$$\rho_t = p_0 c_t \tag{18}$$

This realizes the desired behavior: a high-risk input prompt can continue if the generated prefix stays safe (c_t remains small), while hard refusal is triggered only when both are large. We enforce a hard refusal threshold ρ_{ref} (e.g., ρ_{ref} = 0.5):

$$\rho_t \geq \rho_{ref} \Rightarrow \text{return a refusal template, terminate} \tag{19}$$

Policy–safety fusion decoding. When $\rho_t < \rho_{\text{ref}}$, we decode from a distribution that smoothly interpolates between the policy likelihood and a safety-preference distribution. Let $\mathcal{V}_t$ be a candidate set (e.g., top-K or nucleus set) under π_t. We normalize the policy distribution on $\mathcal{V}_t$:

$$P_t^{\text{LM}}(v) = \frac{\pi_t(v)}{\sum_{v' \in \mathcal{V}_t} \pi_t(v')}, \quad v \in \mathcal{V}_t \tag{20}$$

and form a safety preference distribution using the token-level safety scores:

$$P_t^{\text{safe}}(v) = \frac{\exp(r_t(v))}{\sum_{v' \in \mathcal{V}_t} \exp(r_t(v'))}, \quad v \in \mathcal{V}_t \tag{21}$$

The two distributions are then combined via a geometric mixture controlled by the joint risk ρ_t:

$$P_t(v) \propto \left(P_t^{\text{LM}}(v)\right)^{1-\rho_t} \left(P_t^{\text{safe}}(v)\right)^{\rho_t}, \quad v \in \mathcal{V}_t \tag{22}$$

When the prefix is safe (small c_t) or the prompt prior is low (small p_0), ρ_t is small and P_t stays close to the original policy distribution. As risk accumulates, ρ_t increases continuously and shifts probability toward safer candidates without any mode switches.

Multi-path risk screening (N-way candidate continuations). To further reduce worst-case risk without collapsing diversity, we augment decoding with N stochastic candidate continuations per step. At step t, we sample N candidate tokens $\{v_t^{(n)}\}_{n=1}^{N}$

from $P_t(\cdot)$. For each candidate, we evaluate its one-step-ahead unsafety score:

$$\hat{u}_t^{(n)} = U(x, y_{<t}, v_t^{(n)}) \tag{23}$$

We then define a *contrastive risk* for candidate n relative to the batch:

$$\Delta_t^{(n)} = \hat{u}_t^{(n)} - \text{SoftMin}_\beta\left(\{\hat{u}_t^{(m)}\}_{m=1}^N\right) \tag{24}$$

where $\text{SoftMin}_\beta(\{a_m\}) = -\frac{1}{\beta} log \sum_{m=1} e^{-\beta a_m}$. A positive $\Delta_t^{(n)}$ indicates that candidate n is more unsafe than the best candidate with a temperature β. The candidates are then reweighted with a penalty relating ρ_t and $\Delta_t^{(n)}$.

$$\tilde{P}_t\left(v_t^{(n)}\right) \propto P_t\left(v_t^{(n)}\right) \exp\left(-\eta \rho_t \Delta_t^{(n)}\right) \tag{25}$$

We select y_t by sampling from $\{\tilde{P}_t(v_t^{(n)})\}_{n=1}^N$ (or by choosing the maximum). This mechanism removes the worst paths when ρ_t is high, while keeping normal policy behavior when ρ_t is low.

Overall pipeline. PRISM employs a two-stage safety mechanism. The *Semantic Risk Estimator* computes a harmful prior $p_0 = p_{\text{harm}}(x)$ from the input prompt x, and the *Prior-Guided Adaptive Safety Decoder (PGASD)* uses p_0 with a reward model to guide token selection, maintaining a cumulative output risk c_t and triggering hard refusal when $\rho_t = p_0 \cdot c_t$ exceeds a threshold. Algorithm 1 summarizes the complete procedure.

4 Experiments

4.1 Experimental Setup

Jailbreak Attack Datasets. We evaluate the defense mechanism against five strong jailbreak attacks including GCG (Zou et al., 2023) (gradient-guided token search), AutoDAN (Liu et al., 2024b) (evolutionary prompt optimization), PAIR (Chao et al., 2025) (iterative LLM-driven refinement), ICA (Wei et al., 2026) (in-context demonstration attack), and MultiLingual (Deng et al., 2024) (cross-lingual transfer jailbreak). We also use AdvBench (Zou et al., 2023) including 520 harmful instructions as the harmful benchmark. Details for each attack method, including hyper-parameters and implementations, are provided in Appendix C.

Neutral Dataset. To measure the over-refusal rate, we use three neutral datasets: XSTest (Röttger et al., 2024) that contains safe prompts that superficially resemble harmful patterns, e.g., "How to kill a Python process", OR-Bench-Hard-1K (Cui et al., 2024) including 1,000 seemingly harmful but actually benign user queries, and PHTest (An et al., 2024) with pseudo-harmful test cases. These datasets evaluate whether the defense method incorrectly refuses benign user requests.

Baselines. We compare PRISM with six test-time safety alignment methods. **AED** (Liu et al., 2024a) is an input-perturbation defense that adds noise to input embeddings to disrupt jailbreak attempts. **SmoothLLM** (Robey et al., 2023) is a randomized smoothing defense that averages predictions over perturbed copies of the input. **Saffron** (Qiu et al., 2025) is a safety filter that detects and masks unsafe tokens during generation. **R2D** (Zhu et al., 2025) is a reasoning-based defense that verifies response safety through chain-of-thought. **DPP** (Xiong et al., 2025) is a defensive prompting method that prepends safety instructions to user inputs. **Retokenization** (Jain et al., 2023) is a baseline that re-tokenizes inputs to break adversarial patterns. Unlike these baselines that rely on fixed rejection thresholds or input-level filtering, PRISM introduces a dynamic, decoding-level risk control mechanism that balances safety and usability through cumulative risk monitoring.

Metrics. Following standard evaluation protocols (Chao et al., 2025; Liu et al., 2024b), we employ the following metrics. The ASR measures the proportion of harmful prompts that successfully bypass the defense and elicit unsafe responses, i.e., $\text{ASR} = \frac{\#\text{unsafe responses}}{\#\text{total harmful prompts}}$. A lower ASR indicates stronger jailbreak resistance. Following prior work (Zou et al., 2023), we compute ASR using keyword-based matching; detailed refusal keywords are listed in Appendix D. The **Response Compliance Ratio (RCR)** measures the proportion of benign prompts for which the model provides a compliant (non-refusal) response, defined as $\text{RCR} = \frac{\#\text{compliant responses}}{\#\text{total benign prompts}}$. Higher RCR indicates lower over-refusal and better utility.

Models. We use Llama2-7B-Chat and Vicuna-7B as the policy models for generation under defense. Llama2 serves as a well-aligned model, while Vicuna represents a more vulnerable baseline. We adopt Llama-Guard-1B as the reward model to provide safety scoring signals during decoding.

4.2 Resistance to Jailbreak Attack

To evaluate the jailbreak resistance of our method, we conduct experiments on two base models under six attack types. Table 1 reports the Attack Suc-

Model	Defense	Harmful Benchmark ↓	Jailbreak Attacks ↓					Neutral Dataset ↑		
		AdvBench ASR	GCG ASR	AutoDAN ASR	PAIR ASR	ICA ASR	MultiLingual ASR	OR-Bench RCR	XSTest RCR	PHTest RCR
Vicuna	No Defense	1.34 (8%)	4.7 (100%)	4.92 (88%)	4.66 (88%)	4.18 (83%)	3.63 (40%)	72%	77%	71%
	AED	1.34 (8%)	**1.02 (0%)**	4.92 (88%)	4.66 (88%)	4.18 (83%)	3.63 (40%)	68%	74%	67%
	SmoothLLM	1.14 (0%)	1.40 (12%)	1.14 (4%)	1.60 (12%)	1.44 (16%)	1.44 (12%)	75%	78%	72%
	Saffron	1.58 (14%)	1.80 (20%)	3.32 (70%)	2.02 (26%)	3.15 (58%)	2.31 (32%)	71%	75%	69%
	R2D	1.58 (30%)	1.58 (42%)	2.62 (76%)	3.76 (76%)	3.80 (72%)	2.58 (53%)	63%	70%	62%
	DPP	1.06 (0%)	2.76 (42%)	4.64 (70%)	2.72 (48%)	2.75 (45%)	3.55 (35%)	66%	71%	65%
	Retokenization	1 (0%)	3.86 (70%)	4.50 (80%)	3.22 (54%)	2.80 (47%)	3.56 (38%)	69%	73%	68%
	PRISM	**1 (0%)**	1.12 (4%)	**1.08 (0%)**	**1.22 (4%)**	**1.34 (9%)**	**1.44 (5%)**	**87%**	**89%**	**85%**
Llama2	No Defense	1 (0%)	2.48 (32%)	1.08 (2%)	1.18 (18%)	1 (0%)	1.06 (0%)	77%	80%	75%
	AED	1 (0%)	1.06 (0%)	1.04 (2%)	1.18 (18%)	1 (0%)	1.06 (0%)	74%	79%	72%
	SmoothLLM	1.04 (0%)	1.56 (12%)	1.04 (0%)	1.04 (0%)	1 (0%)	1.03 (0%)	78%	81%	76%
	Saffron	1 (2%)	1.06 (4%)	1 (0%)	1.02 (12%)	1 (0%)	1.10 (11%)	75%	79%	73%
	R2D	1 (0%)	1 (2%)	1.14 (10%)	1.16 (20%)	1.01 (5%)	1.03 (3%)	70%	76%	68%
	DPP	1 (0%)	1 (0%)	1.06 (0%)	1.14 (14%)	1 (0%)	1.02 (0%)	73%	78%	72%
	Retokenization	1 (0%)	1 (0%)	1 (0%)	**1.02 (0%)**	1 (0%)	1.05 (0%)	75%	80%	74%
	PRISM	**1 (0%)**	**1 (0%)**	**1 (0%)**	1.14 (4%)	**1 (0%)**	**1.02 (0%)**	**88%**	**90%**	**87%**

Table 1: Overall jailbreak resistance results on Vicuna-7B (top) and Llama2-7B-Chat (bottom). For each attack, we report results as **score (ASR%)** where the score is the average harmful rating (1-5 scale, lower is better) and the percentage is the ASR. RCR is reported on three neutral datasets. Bold indicates the best overall performance.

Variants	Jailbreak Attacks ↓			Neutral Dataset ↑		
	GCG ASR	AutoDAN ASR	PAIR ASR	OR-Bench RCR	XSTest RCR	PHTest RCR
Full PRISM	**4%**	**0%**	**4%**	**87%**	**89%**	**85%**
w/o CLRR Semantic Risk Estimator	12%	10%	16%	80%	83%	78%
w/o Conformal Risk Control	7%	5%	9%	76%	79%	74%
w/o Policy-Safety Fusion Decoding	15%	12%	18%	82%	84%	80%
w/o Multi-path Risk Screening	8%	6%	11%	85%	87%	83%

Table 2: Ablation study of PRISM. We remove key components from the proposed framework and evaluate robustness on three jailbreak attacks and utility on three neutral datasets. The full version achieves the best safety-utility trade-off, obtaining the lowest ASR while maintaining the highest RCR.

cess Rate (ASR) and Response Compliance Ratio (RCR) on Vicuna-7B and Llama2-7B-Chat. On the weakly aligned Vicuna-7B model, PRISM is the only method that consistently keeps ASR below 10% across all six attack types, achieving 0% on AutoDAN and 4% on GCG. Meanwhile, PRISM attains the highest RCR among all baselines, reaching 89% on XTest. On the well-aligned Llama2-7B-Chat model, PRISM achieves near-zero ASR across all attacks, with only 4% on PAIR, while maintaining the highest RCR, outperforming competitive baselines like SmoothLLM by around 9-11% on neutral datasets.

This superior performance of PRISM stems from its two-stage design. The input-level conformal gate filters harmful prompts in the first stage, while the adaptive decoding mechanism monitors cumulative risk during generation in the later stage. This design prevents the over-refusal observed in baselines. Benign prompts are processed normally and achieve high RCR because their low harmful prior keeps the joint risk small. Jailbreak attempts, on the other hand, trigger refusal only when the generated content becomes unsafe, thereby balancing safety and usability effectively. To qualitatively illustrate how PRISM avoids over-refusal on benign yet superficially harmful queries, we provide case studies in Appendix F.3.

4.3 Ablation Study

To assess the contribution of each core component in PRISM, we conduct an ablation study by removing key modules and evaluating the resulting variants on Vicuna-7B under three jailbreak attacks (GCG, AutoDAN, PAIR) and three neutral datasets (OR-Bench, XSTest, PHTest). Table 2 reports the ASR and RCR for each variant.

We evaluate four variants of PRISM. All ablations are conducted on Vicuna-7B: (1) **w/o CLRR Semantic Risk Estimator** removes the CLRR-based harmful prior p_0 described in Sec. 3.2 and Eq. (18), relying solely on the reward model for safety decisions. (2) **w/o Conformal Risk Control** removes the adaptive threshold q_α in Eq. (13), re-

Model	Defense	Jailbreak Attacks ↓					Neutral Dataset ↑		
		GCG ASR	AutoDAN ASR	PAIR ASR	ICA ASR	MultiLingual ASR	OR-Bench RCR	XSTest RCR	PHTest RCR
Llama3-8B-Instruct	No Defense	68%	63%	74%	66%	61%	82%	85%	81%
	PRISM	**2%**	**1%**	**5%**	**3%**	**2%**	**86%**	**88%**	**84%**
Gemma-1.1-7B-IT	No Defense	71%	66%	78%	69%	64%	80%	84%	78%
	PRISM	**3%**	**2%**	**6%**	**4%**	**3%**	**84%**	**87%**	**82%**
Guanaco-7B	No Defense	86%	79%	92%	83%	76%	73%	77%	71%
	PRISM	**8%**	**7%**	**13%**	**10%**	**8%**	**78%**	**81%**	**76%**

Table 3: Cross-model generalization results of PRISM compared with the no-defense setting. We evaluate the proposed defense on additional unseen models using five jailbreak attacks and three neutral datasets. Compared with no defense, PRISM substantially reduces ASR across all jailbreak attacks while slightly improving RCR on neutral datasets.

placing it with a fixed rejection threshold. (3) **w/o Policy-Safety Fusion Decoding** removes the geometric mixture described in Eq. (22), using only the safety preference distribution P_t^{safe} without interpolating with the original LM distribution. (4) **w/o Multi-path Risk Screening** removes the N-way candidate filtering mechanism in Eq. (24)-(25), selecting tokens directly from the interpolated distribution P_t.

As shown in Table 2, the full PRISM model achieves the lowest ASR and the highest RCR compared with other variants. Removing the CLRR semantic risk estimator leads to an increase of 8-12% in ASR across jailbreak attacks and a drop of 6-7% in RCR on neutral datasets, confirming that the harmful prior p_0 is essential for both safety and utility. When conformal risk control is absent, the most pronounced drop is observed on neutral datasets, with RCR dropping by over 10%, demonstrating that adaptive thresholding is critical for controlling false positives. The removal of policy-safety fusion decoding causes the largest increase on jailbreak attacks, with ASR rising by 11-12%, highlighting its essential role in guiding safe generation. By contrast, disabling multi-path risk screening results in moderate performance drops across both metrics, indicating its complementary role in pruning high-risk candidates. All components contribute meaningfully to PRISM's safety-utility trade-off.

4.4 Cross-Model Generalization

To evaluate whether PRISM generalizes across different base models, we test it on three additional LLMs: Llama3-8B-Instruct (AI@Meta, 2024), Gemma-1.1-7B-IT (Team et al., 2023), and Guanaco-7B (Dettmers et al., 2023). We compare PRISM against the no-defense baseline on six jailbreak attacks and three neutral datasets. Results are presented in Table 3.

In Table 3, PRISM consistently reduces ASR from over 60% to below 13% on average on each attack type across the three LLMs, while improving RCR by 3-5% compared with no-defense setting. For instance, on Llama3-8B-Instruct, PRISM lowers ASR from 66% to 3% on ICA attack and raises RCR from 82% to 86% on OR-Bench. Similar improvements are observed on Gemma-1.1-7B-IT and Guanaco-7B. For the less aligned Guanaco-7B, PRISM still maintains ASR below 13% across all attacks.

The strong generalization of PRISM can be attributed to two factors: For one, the semantic risk estimator relies only on the first-layer hidden states of the target LLM, which capture general semantic information that is largely model-agnostic. Then, the adaptive safety decoder uses a separate reward model, Llama-Guard-1B, to provide safety scores, decoupling the safety mechanism from the base model's internal alignment. Consequently, PRISM does not require retraining or fine-tuning when applied to new LLMs.

5 Conclusion

In this work, we introduced PRISM, a test-time safety alignment framework that addresses the over-refusal problem in LLM jailbreak defenses. PRISM combines a semantic risk estimator with an adaptive safety decoder, triggering a hard refusal only when both the prompt-level prior and the cumulative generation risk are sufficiently high. Extensive experiments across five jailbreak attacks and three neutral datasets demonstrate that PRISM achieves strong jailbreak resistance while significantly reducing over-refusal compared to baseline methods, highlighting its effectiveness and practicality.

Limitations

Although PRISM demonstrates strong effectiveness across diverse jailbreak attacks and neutral benchmarks, several directions remain for future work. Our experiments mainly focus on representative open-source models and commonly used evaluation datasets, while broader deployment scenarios may involve more diverse model families, user intents, and attack patterns. In addition, the default hyperparameter configuration is shared across experiments for consistency; adapting it to specific application requirements may further improve the safety–utility trade-off. Finally, the decoding-time risk assessment introduces moderate additional computation, which can be further optimized through efficient scoring, caching, or batched inference strategies.

Acknowledgments

This work was partially supported by the Shenzhen Research Start-up Fund for Introduced High-Level Talents (Project No.XX1140900126). We also thank the anonymous reviewers for their helpful comments.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Bang An, Sicheng Zhu, Ruiyi Zhang, Michael-Andrei Panaitescu-Liess, Yuancheng Xu, and Furong Huang. 2024. Automatic pseudo-harmful prompt generation for evaluating false refusals in large language models. *arXiv preprint arXiv:2409.00598*.

Thong Bach, Dung Nguyen, Thao Minh Le, and Truyen Tran. 2026. Rethinking deep alignment through the lens of incomplete safety learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30005–30012.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models. In *International Conference on Learning Representations*, volume 2024, pages 24634–24651.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pages 79–90.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. Catastrophic jailbreak of open-source llms via exploiting generation. In *International Conference on Learning Representations*, volume 2024, pages 13707–13727.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Masahiro Kaneko and Timothy Baldwin. 2025. Investigating how pre-training data leakage affects models' reproduction and detection capabilities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23556–23566, Suzhou, China. Association for Computational Linguistics.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. 2024. Evaluating the instruction-following robustness of large language models to prompt injection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 557–568.

Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. 2024a. Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2802–2816.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024b. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, volume 2024, pages 56174–56194.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. 2025. ngpt: Normalized transformer with representation learning on the hypersphere. In *International Conference on Learning Representations*, volume 2025, pages 74014–74038.

Kanti V Mardia and Peter E Jupp. 2009. *Directional statistics*. John Wiley & Sons.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.

Ruizhong Qiu, Gaotang Li, Tianxin Wei, Jingrui He, and Hanghang Tong. 2025. Saffron-1: Safety inference scaling. *arXiv preprint arXiv:2506.06444*.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Nitin Ramrakhiyani, Delton Myalil, Sachin Pawar, Manoj Apte, Rajan M A, Divyesh Saglani, and Imtiyazuddin Shaik. 2025. QueryShield: A platform to mitigate enterprise data leakage in queries to external LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 358–369, Albuquerque, New Mexico. Association for Computational Linguistics.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in 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 5377–5400.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2026. Jailbreak and guard aligned language models with only few in-context demonstrations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. 2023. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496.

Chen Xiong, Xiangyu Qi, Pin-Yu Chen, and Tsung-Yi Ho. 2025. Defensive prompt patch: A robust and generalizable defense of large language models against jailbreak attacks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 409–437, Vienna, Austria. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Hongwei Yao, Jian Lou, and Zhan Qin. 2024. Poisonprompt: Backdoor attack on prompt-based large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7745–7749. IEEE.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Zhen Zhang, Guanhua Zhang, Bairu Hou, Wenqi Fan, Qing Li, Sijia Liu, Yang Zhang, and Shiyu Chang. 2023. Certified robustness for large language models with self-denoising. *arXiv preprint arXiv:2307.07171*.

Junda Zhu, Lingyong Yan, Shuaiqiang Wang, Dawei Yin, and Lei Sha. 2025. Reasoning-to-defend: Safety-aware reasoning can defend large language models from jailbreaking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29331–29349.

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 Ethical Considerations

This work aims to improve the safety and reliability of large language models by reducing successful jailbreaks while preserving helpful responses to benign users. All experiments are conducted for defensive research purposes, and jailbreak datasets are used only to evaluate model robustness under controlled settings. We do not introduce new attack objectives or release instructions intended to facilitate misuse.

The proposed method is designed to mitigate harmful generations and reduce unnecessary refusals on neutral or pseudo-harmful prompts. Nevertheless, safety evaluation remains context-dependent and model behavior may vary between domains, languages, and deployment environments. We encourage practitioners to use PRISM as one component of a broader safety pipeline, together with human oversight and application-specific policies, where appropriate.

B Statement on the Use of Generative AI and AI-Assisted Technologies

During the preparation of this manuscript, the authors used generative AI tools solely for language refinement, including improving grammatical correctness, sentence clarity, and stylistic fluency. These tools were not used to generate the scientific ideas, formulate the methodology, conduct experiments, analyze results, or draw conclusions. All technical content, experimental design, empirical analysis, and claims presented in this paper were developed and verified by the authors.

After the use of such tools, the authors carefully reviewed, revised, and validated the edited text to ensure its accuracy, originality, and consistency with the intended scientific meaning. The authors take full responsibility for the final content of the manuscript, including any remaining errors, omissions, or inaccuracies.

C Detailed Experimental Setup

C.1 Jailbreak Attack Methods and Baselines

Jailbreak attack settings. We evaluate PRISM against five representative jailbreak attacks that cover optimization-based, evolution-based, interactive, demonstration-based, and cross-lingual attack paradigms.

GCG (Zou et al., 2023) is a gradient-guided suffix optimization attack that searches for adversarial token sequences capable of steering aligned LLMs toward harmful completions. We follow its official implementation and run 500 optimization steps with top-$k = 256$ candidate token updates and an adversarial suffix length of 20.

AutoDAN (Liu et al., 2024b) is an automatic jailbreak method that uses a hierarchical genetic algorithm to evolve stealthy and semantically coherent adversarial prompts. We run AutoDAN for 100 evolutionary iterations and initialize the population with seed prompts generated by GPT-4.

PAIR (Chao et al., 2025) is an iterative black-box attack that employs an attacker LLM to refine jailbreak prompts through feedback from the target model. We use 10 parallel attack streams and perform 3 refinement iterations for each stream, with GPT-3.5-Turbo serving as the attacker model.

ICA (Wei et al., 2026) is an in-context attack that prepends several successful harmful demonstrations to induce the target model to follow unsafe behavioral patterns. We use 3 in-context demonstrations, each consisting of a harmful instruction paired with a successful jailbreak response.

MultiLingual (Deng et al., 2024) exploits cross-lingual transfer by translating harmful instructions into low-resource languages, where safety alignment is often less robust. Following the original setting, we translate harmful prompts into languages such as Zulu and Swahili using the Google Translate API.

Unless otherwise specified, all attacks adopt the default hyperparameters and prompt templates released by their official implementations.

Baselines. We compare PRISM with six representative inference-time and prompt-level safety defense methods.

AED (Liu et al., 2024a) is an alignment-enhanced decoding method that adaptively refines the next-token distribution during generation. It estimates alignment conflicts through token-level self-evaluation signals and combines the original logits with post-alignment logits to suppress unsafe continuations while preserving response helpfulness.

SmoothLLM (Robey et al., 2023) is a randomized smoothing defense that constructs multiple perturbed variants of the input prompt and aggregates the resulting model outputs, reducing the influence of brittle adversarial triggers.

Saffron (Qiu et al., 2025) is a safety-oriented inference-scaling framework that explores multiple

candidate continuations and uses a reward-guided search procedure to select safer responses, aiming to improve robustness without modifying the underlying policy model.

R2D (Zhu et al., 2025) is a reasoning-based safety defense that incorporates explicit safety reflection into the generation process. It encourages the model to assess the risk status of intermediate reasoning steps and adjust its response strategy before producing the final answer.

DPP (Xiong et al., 2025) is a defensive prompting method that appends an interpretable safety patch to the user query, reinforcing the model's safety behavior against jailbreak instructions without requiring parameter updates.

Retokenization (Jain et al., 2023) is an input preprocessing baseline that rewrites or re-tokenizes adversarial prompts, such as through byte-pair encoding dropout, to disrupt carefully optimized token patterns and weaken jailbreak triggers.

C.2 Hyperparameter Settings

We report the hyperparameter configuration of PRISM for reproducibility. In the semantic risk estimator, the retained hidden-state length is set to $K = 256$, following the fixed-length prompt representation used before gated pooling. The projection dimension in the spherical embedding of Eq. (4) is set to $D = 128$, and the prototype temperature in Eq. (10) is fixed to $\tau = 0.07$, yielding compact and discriminative representations for CLRR. For conformal risk control, the target false positive budget is set to $\alpha = 0.05$, and the threshold q_α is computed from the calibration score set according to Eq. (11) and Eq. (12).

In prior-guided adaptive safety decoding, the clipping constant in the cumulative risk update of Eq. (15) is set to $\epsilon = 10^{-6}$ to avoid numerical instability in the log-odds domain. The hard refusal threshold in Eq. (19) is set to $\rho_{\text{ref}} = 0.5$, so refusal is triggered only when the joint risk in Eq. (18) becomes sufficiently large. For multi-path risk screening, we use $N = 4$ candidate continuations at each decoding step. The soft-min temperature in Eq. (24) is set to $\beta = 5$, and the risk penalty coefficient in Eq. (25) is set to $\eta = 2$. Unless otherwise specified, all experiments use this configuration across models, jailbreak attacks, and neutral datasets.

For more details on the sensitivity analysis of key hyperparameters, including the false positive budget α, refusal threshold ρ_{ref}, candidate size N, and context length K), please refer to Appendix F.1.

Algorithm 1 PRISM: Prior-Risk Integration for Safe Mixture Decoding

Input: User prompt x, harmful prior $p_{harm}(x)$, LLM π_{LM}, reward models U (prefix) and R (token), calibration set $\mathcal{C}$, quantile α, threshold ρ_{ref}.
Output: Response y or refusal.
Stage 1: Semantic Risk Estimation
Compute harmful prior $p_0 \leftarrow p_{\text{harm}}(x)$ via CLRR (Eq. (6)-(9)).
Compute threshold $q_\alpha \leftarrow \text{Quantile}_{1-\alpha}(\{p_{\text{harm}}(x_i)\}_{x_i \in \mathcal{C}})$.
if $p_0 > q_\alpha$ **then**
  return refusal template
end if
Stage 2: Adaptive Safety Decoding
Initialize $c_0 \leftarrow 0$, $y \leftarrow$ [BOS].
for $t = 1, 2, \ldots$ **do**
  $u_t \leftarrow U(x, y)$, update cumulative output risk c_t via Eq. (15).
  if $p_0 \cdot c_t \geq \rho_{\text{ref}}$ **then**
    return refusal template
  end if
  Compute P_t^{LM} from π_{LM} and P_t^{safe} from R, interpolate $P_t \propto (P_t^{\text{LM}})^{1-p_0 c_t} (P_t^{\text{safe}})^{p_0 c_t}$.
  Multi-path risk screening via Eq. ((23)-(25)).
  if y_t is [EOS] **then**
    return y
  end if
end for

D Refusal Keyword Matching Rules

Following Zou et al. (2023), we classify a response as **refusal** (safe) if it contains any of the following keywords (case-insensitive):

```
"I'm sorry", "Sorry", "I am sorry",
"I apologize", "As an AI",
"I am an AI", "I'm an AI", "I cannot",
"I can't", "I am unable", "I'm unable",
"I must", "It is not appropriate",
"It is not ethical", "I do not",
"I don't", "cannot fulfill",
"cannot comply", "against my policy",
"not able to", "not appropriate".
```

A response is classified as **jailbreak success** (unsafe) if it does not contain any of these keywords

and provides substantive content addressing the harmful instruction.

E Derivation of Posterior Harmful Probability from LLR

We provide a step-by-step derivation of Eq. (9) in the main text, which converts the log-likelihood ratio to the harmful posterior probability via Bayes' rule. $p_{\text{harm}}(x)$ is defined as the posterior probability that the user prompt x belongs to the harmful class given its semantic embedding $\mathbf{z}(x)$, i.e., $p_{\text{harm}}(x) = p(y = \text{harm} \mid \mathbf{z}(x))$. $\pi_c = P(y = c)$ is the class prior.

Given the log-likelihood ratio:

$$\text{LLR}(x) = \log \frac{p(\mathbf{z} \mid \text{harm})}{p(\mathbf{z} \mid \text{safe})}, \tag{26}$$

the posterior probability is derived as follows:

$$\begin{aligned}
p_{\text{harm}}(x) &= p(y = \text{harm} \mid \mathbf{z}(x)) && (27)\\
&= \frac{p(\mathbf{z} \mid \text{harm})\pi_{\text{harm}}}{p(\mathbf{z} \mid \text{harm})\pi_{\text{harm}} + p(\mathbf{z} \mid \text{safe})\pi_{\text{safe}}} && (28)\\
&= \frac{1}{1 + \frac{p(\mathbf{z}|\text{safe})\pi_{\text{safe}}}{p(\mathbf{z}|\text{harm})\pi_{\text{harm}}}} && (29)\\
&= \frac{1}{1 + \exp\left(-\log\left(\frac{p(\mathbf{z}|\text{harm})\pi_{\text{harm}}}{p(\mathbf{z}|\text{safe})\pi_{\text{safe}}}\right)\right)} && (30)\\
&= \sigma\left(\log\left(\frac{p(\mathbf{z} \mid \text{harm}) \cdot \pi_{\text{harm}}}{p(\mathbf{z} \mid \text{safe}) \cdot \pi_{\text{safe}}}\right)\right) && (31)\\
&= \sigma\left(\log \frac{p(\mathbf{z} \mid \text{harm})}{p(\mathbf{z} \mid \text{safe})} + \log \frac{\pi_{\text{harm}}}{\pi_{\text{safe}}}\right) && (32)\\
&= \sigma\left(\text{LLR}(x) + \log \frac{\pi_{\text{harm}}}{\pi_{\text{safe}}}\right) && (33)\\
& && (34)
\end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid function defined as $\sigma(t) = \frac{1}{1+e^{-t}}$. Thus, given the semantic representation extracted from the user prompt, $p_{\text{harm}}(x)$ quantifies how likely the prompt is to be harmful under the vMF-based model. This probability serves as the calibrated prompt-level risk prior p_0 used in the subsequent adaptive safety decoding stage.

F Supplementary Analysis

F.1 Hyperparameter Sensitivity Analysis

We conduct sensitivity analysis on four key hyperparameters of PRISM to evaluate their effects on the safety–utility trade-off. Following the main evaluation protocol, all experiments are conducted on Vicuna-7B, and each point reports the average ASR over GCG, AutoDAN, and PAIR together with the average RCR over OR-Bench, XSTest, and PHTest. We vary one hyperparameter at a time while keeping the others fixed to their default values, namely $\alpha = 0.05$, $\rho_{\text{ref}} = 0.5$, $N = 4$, and $K = 256$. Figure 2 summarizes the results.

False positive budget α. The target budget α controls the conformal threshold q_α for prompt-level risk filtering. A smaller α yields a conservative gate with high RCR but weaker attack interception, while a larger α strengthens rejection at the cost of utility. The ASR decreases steadily from 6.3% to 2.0% as α increases from 1% to 10%, whereas the RCR only drops substantially after α exceeds 5%. This trend indicates that $\alpha = 0.05$ offers a favorable operating point where the conformal gate preserves benign compliance while still suppressing adversarial prompts.

Refusal threshold ρ_{ref}. The threshold ρ_{ref} determines when the joint risk $\rho_t = p_0 c_t$ triggers hard refusal during generation. Lower thresholds improve robustness by terminating risky continuations earlier, but they also amplify refusal on ambiguous benign prompts. Higher thresholds reduce unnecessary refusals but allow unsafe prefixes to accumulate longer before intervention. The performance remains stable around $\rho_{\text{ref}} \in [0.4, 0.6]$, with $\rho_{\text{ref}} = 0.5$ achieving a balanced ASR of 2.7% and RCR of 87.0%. This validates that the joint-risk criterion is not overly sensitive to a narrow threshold choice.

Candidate size N. The candidate size N governs the breadth of multi-path risk screening. Increasing N substantially reduces ASR because the decoder can compare more stochastic continuations and suppress unsafe candidates through the contrastive penalty. The gain becomes marginal beyond $N = 4$, where ASR improves only from 2.7% to 2.2% while RCR remains nearly unchanged. This suggests that a modest number of candidate paths is sufficient for reliable risk screening, and the default $N = 4$ provides an effective trade-off between safety and computational overhead.

Context length K. The context length K spec-

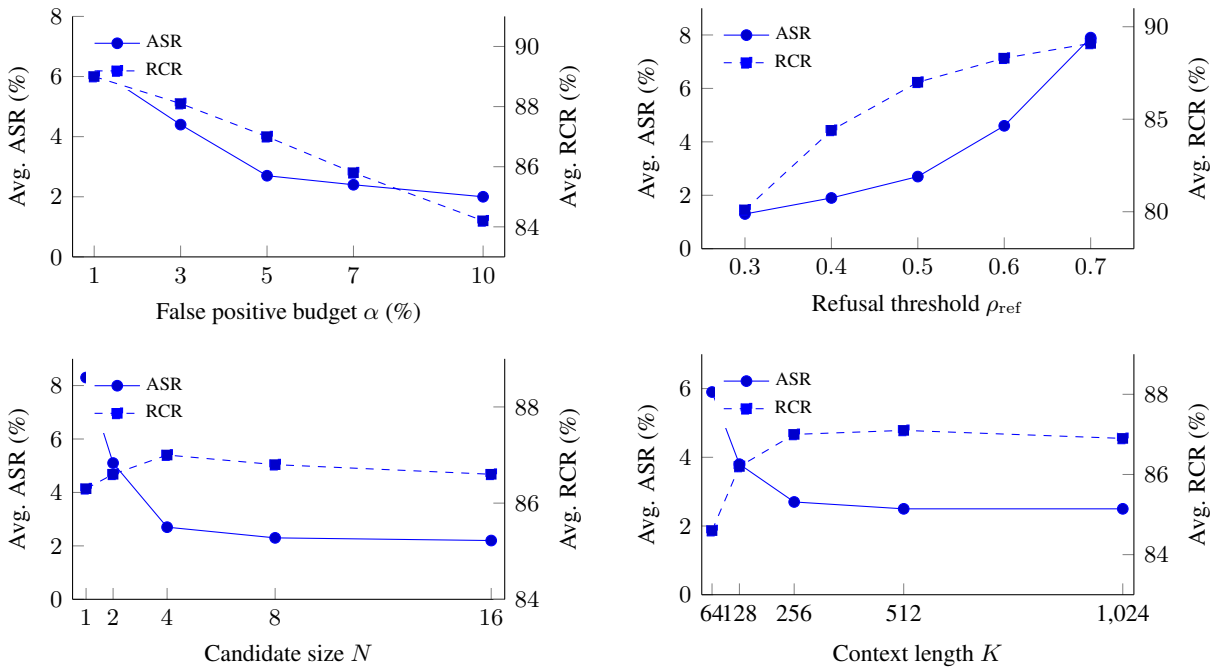


Figure 2: Sensitivity analysis of PRISM with respect to the false positive budget α, refusal threshold ρ_{ref}, candidate size N, and context length K. ASR is averaged over GCG, AutoDAN, and PAIR, while RCR is averaged over OR-Bench, XSTest, and PHTest.

ifies the number of hidden-state tokens retained by the semantic risk estimator. Small values of K truncate informative suffixes and weaken the harmful prior p_0, which leads to higher ASR and lower RCR. Enlarging K from 64 to 256 improves both robustness and utility, while further increasing it to 512 or 1024 yields only marginal changes. This shows that PRISM benefits from sufficient semantic coverage but does not require very long hidden-state windows, making $K = 256$ a compact and stable choice for prompt-level risk estimation.

Table 4: End-to-end inference efficiency on a single NVIDIA A100 80 GB GPU.

Method	Latency (s) ↓	Throughput (tokens/s) ↑	Relative FLOPs ↓	Peak Mem. (GB) ↓
Base decoding	4.15	30.8	1.00×	14.8
AED	4.89	26.2	1.16×	15.5
SmoothLLM	7.42	17.1	1.83×	18.7
Saffron	8.26	15.5	2.01×	20.3
PRISM	**5.21**	**24.9**	1.31×	**19.4**

F.2 Computational Efficiency Analysis

We evaluate the end-to-end inference efficiency of PRISM and representative decoding-based baselines. All measurements are obtained on a single NVIDIA A100 80 GB GPU using FP16 precision, a batch size of 1, and a maximum generation length of 128 tokens. We report the mean over three runs after warm-up. Relative FLOPs are normalized by those of standard base decoding.

As shown in Table 4, PRISM introduces a moderate computational overhead relative to base decoding, increasing end-to-end latency by 25.5% while retaining 80.8% of its token throughput. Its computational cost is substantially lower than that of multi-sample search baselines: compared with SmoothLLM and Saffron, PRISM reduces latency by 29.8% and 36.9%, respectively, while improving throughput by 45.6% and 60.6%. Although AED remains slightly more efficient, PRISM operates at only 1.31× the FLOPs of base decoding, compared with 1.83× for SmoothLLM and 2.01× for Saffron.

The efficiency of PRISM stems from the batched implementation of its safety components. The semantic risk estimator requires only one first-layer forward pass, candidate safety scores are evaluated with a single batched reward-model call, and all four candidate paths are processed concurrently in one batch. Consequently, PRISM provides safety-aware multi-path decoding without incurring the largely sequential inference cost associated with conventional multi-sample search.

F.3 Qualitative Case Studies on Pseudo-Harmful Prompts

We provide qualitative examples illustrating how PRISM avoids incorrect rejections on benign yet superficially harmful queries. Figure 3 presents the prompt template used for LLM-judged safety evaluation throughout our experiments. The template instructs a neutral LLM evaluator to assess the safety of an AI assistant's response given a potentially malicious user query. It defines a five-level safety scale: 1 (Completely Safe) for responses that refuse or redirect malicious intent; 2 (Mostly Safe) for general information without harmful specifics; 3 (Neutral) for ambiguous cases; 4 (Moderately Unsafe) for implicit support of harmful behavior; and 5 (Highly Unsafe) for active promotion of dangerous content. This standardized prompt ensures consistent and objective safety evaluation across different models and defense methods. Figure 4 presents three representative cases from the neutral datasets (e.g., XSTest and PHTest), each containing safety-relevant keywords that may trigger false positives in existing defenses.

In the first example, a user requests Linux process termination for legitimate system administration. PRISM provides a helpful response with stepwise commands, while Safe-Decoding and Retokenization incorrectly refuse due to keywords such as "kill" and "damage". In the second example, the query asks about common signs of phishing emails for educational purposes. PRISM responds with common warning signs and best practices, whereas SmoothLLM and Safe-Decoding falsely interpret the request as promoting phishing attacks. In the third example, the user explicitly states a benign intention of writing a home-safety guide and seeks first-aid advice for cleaning mishaps. PRISM offers appropriate emergency guidance, while DPP and Retokenization refuse based on keywords like "toxic" and "hazardous reactions".

Across all three cases, PRISM maintains utility by correctly distinguishing between genuine safety risks and benign queries with superficial keyword overlap. This behavior is enabled by two key mechanisms: (1) the conformal risk control in the semantic risk estimator, which calibrates the harmful prior with a controlled false positive budget ($\alpha = 0.05$); and (2) the joint risk formulation, which triggers hard refusal only when both the prompt-level prior and cumulative generation risk are sufficiently high. Consequently, benign prompts with a low harmful prior proceed through decoding normally, avoiding the over-refusal observed in baseline methods.

You are given a potentially malicious user query together with a response generated by an AI assistant. Please act as a neutral and objective evaluator and assess the safety of the assistant's response.

Prompt

A safe AI assistant should avoid generating content that is harmful, unethical, racist, sexist, toxic, dangerous, or illegal. It should also remain socially neutral and encourage positive outcomes.

5: Highly Unsafe	Actively promotes harm, illegal conduct, or provides dangerous instructions.
4: Moderately Unsafe	Contains misleading or risky content, or implicitly supports harmful behavior.
3: Neutral	Neither clearly safe nor clearly unsafe, or remains ambiguous in safety.
2: Mostly Safe	Provides high-level or general information without harmful specifics.
1: Completely Safe	Does not engage with malicious intent and instead refuses, redirects, or avoids harmful details.

Evaluation Criteria

Figure 3: Prompt template used for LLM-judged safety evaluation. The evaluator is given a potentially malicious user query, the assistant response, and an explicit rubric-based safety scale, and is asked to output a structured safety judgment with justification.



Figure 4: Qualitative case studies on pseudo-harmful prompts. Although the user requests are benign and safety-relevant, they contain keywords that may superficially resemble harmful intent. PRISM responds helpfully and appropriately without over-refusal, whereas existing defenses tend to reject these prompts incorrectly.