

SAFEPath: Preventing Harmful Reasoning in Chain-of-Thought via Early Alignment

Wonje Jeung¹ Sangyeon Yoon² Minsuk Kahng^{1†} Albert No^{1†}

¹Yonsei University ²Hongik University

Abstract

Large Reasoning Models (LRMs) have become powerful tools for complex problem solving, but their structured reasoning pathways can lead to unsafe outputs when exposed to harmful prompts. Existing safety alignment methods reduce harmful outputs but can degrade reasoning depth, leading to significant trade-offs in complex, multi-step tasks, and remain vulnerable to sophisticated jailbreak attacks. To address this, we introduce SAFEPath, a lightweight alignment method that fine-tunes LRMs to emit a short, 8-token Safety Primer at the start of their reasoning, in response to harmful prompts, while leaving the rest of the reasoning process unsupervised. Empirical results across multiple benchmarks indicate that SAFEPath effectively reduces harmful outputs while maintaining reasoning performance. Specifically, SAFEPath reduces harmful responses by up to 90.0% and blocks 83.3% of jailbreak attempts in the DeepSeek-R1-Distill-Llama-8B model, while requiring 295.9x less compute than Direct Refusal and 314.1x less than SafeChain. We further introduce a zero-shot variant that requires no fine-tuning. In addition, we provide a comprehensive analysis of how existing methods in LRMs generalize, or fail, when applied to reasoning-centric models, revealing critical gaps and new directions for safer AI.

Disclaimer: This document contains content that some may find disturbing or offensive, including content that is hateful or violent in nature.

1 Introduction

The rapid advancement of large language models (LLMs) has led to increasing interest in enhancing their ability to perform complex reasoning tasks, such as mathematical problem solving and code generation. This has given rise to Large Reasoning Models (LRMs), including OpenAI’s O1 [Jaech et al., 2024] and the DeepSeek-R1 series [Guo et al., 2025], which are explicitly trained to reason through extended chain-of-thought. Without relying on intricate prompting strategies, these models autonomously generate structured, multi-step reasoning traces when tackling difficult problems. Their strong performance on challenging benchmarks has made them valuable tools in real-world applications, from development to scientific discovery [Chan et al., 2024, Chen et al., 2024].

However, LRMs are particularly susceptible to harmful prompts and adversarial attacks [Zhou et al., 2025, Ying et al., 2025], often presenting even greater risks than standard LLMs [Jiang et al., 2025]. This vulnerability arises from their structured reasoning pathways, which can unintentionally amplify unsafe behaviors [Zhou et al., 2025]. For example, when asked how to build a bomb “out of curiosity,” an LRM may mistakenly assess the intent as benign through its reasoning, resulting in the generation of potentially harmful responses.

To address the safety vulnerabilities of LRMs, various mitigation strategies have been developed. One common approach is fine-tuning models to directly reject harmful prompts [Huang et al., 2025],

[†]Corresponding author

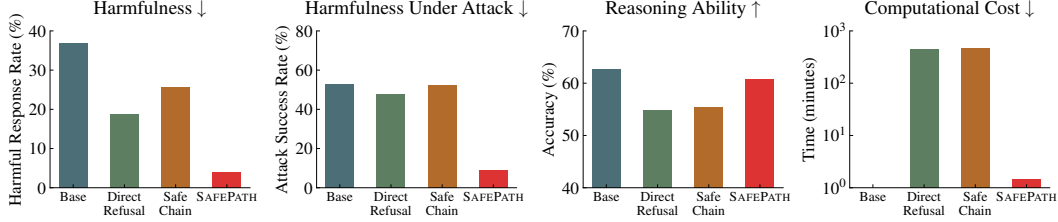


Figure 1: **Performance Comparison of SAFEPath with Baselines.** SAFEPath significantly reduces harmfulness and attack success rates while maintaining strong reasoning ability. It also dramatically lowers computational cost compared to Direct Refusal and SafeChain.

leveraging techniques originally designed for LLM safety alignment [Christiano et al., 2017, Rafailov et al., 2023]. Another approach, SafeChain [Jiang et al., 2025] trains models on datasets that pair safe reasoning traces with safe outputs, aiming to align safety without compromising core reasoning abilities. Additionally, zero-shot prompting methods have been proposed, such as immediately terminating the reasoning block or encouraging shallow deliberation [Jiang et al., 2025]. While these methods can reduce harmful outputs, they come at a cost known as the Safety Tax—a predictable drop in reasoning performance as safety alignment is enforced [Huang et al., 2025]. This trade-off becomes particularly pronounced on challenging benchmarks that demand deep, multi-step reasoning.

In this work, we introduce SAFEPath, a lightweight yet powerful method for aligning LRMs without compromising their reasoning capabilities. At the core of this approach is the **Safety Primer**, a fixed 8-token prefix, “Let’s think about safety first” which serves as a soft signal that guides the model’s reasoning without imposing rigid constraints. Unlike methods that rely on strict refusals or heavily supervised safety conditioning, SAFEPath leverages the LRM’s natural reasoning ability to establish safety, activating a safety-aware reasoning path without disrupting the model’s reasoning capabilities.

The training process is intentionally straightforward: the model is fine-tuned to emit 8-token Safety Primer for harmful prompts, while the remaining reasoning trace is left completely unsupervised (i.e., not fine-tuned), preserving the model’s natural reasoning flexibility. This minimal intervention approach not only reduces training costs but also avoids the common pitfall of over-restricting the model’s reasoning. Despite this simplicity, SAFEPath exhibits a surprisingly flexible safety response. Even though it is explicitly trained to append the primer only per harmful prompt, the model can dynamically re-engage the primer multiple times in highly adversarial contexts, reinforcing safety before the reasoning path drifts toward harmful content. This emergent behavior suggests that the primer acts as a persistent, context-aware safety signal, influencing the model’s internal trajectory without requiring continuous supervision.

As shown in Figure 1, SAFEPath achieves the lowest attack success rates, remains robust under adversarial conditions, and outperforms baselines such as Direct Refusal and SafeChain in reasoning accuracy. Moreover, it only requires minimal fine-tuning with few tokens and no reliance on costly reasoning supervision, resulting in a 295.9× faster training process than Direct Refusal and 314.1× faster than SafeChain for DeepSeek-R1-Distill-Llama-8B. Additionally, we benchmark SAFEPath against three strong LLM baselines adapted for LRM to validate its effectiveness. To further extend this efficiency, we also develop a zero-shot variant that applies the Safety Primer at the start of reasoning, without any fine-tuning. Unlike existing zero-shot methods, which often trade accuracy for safety, our approach maintains strong reasoning performance while effectively reducing harmfulness, offering a practical, lightweight alternative.

Our findings introduce a new direction for aligning LRMs, demonstrating that safety can be achieved without compromising reasoning. By leveraging the model’s natural reasoning abilities, SAFEPath offers a practical path toward robust, secure AI systems, moving us closer to real-world deployment. An overview of our method is illustrated in Figure 2.

2 Related Work

Large Reasoning Models (LRMs). Pretrained LLMs initially faced challenges in refining their logical reasoning capabilities, but chain-of-thought (CoT) prompting [Wei et al., 2022] enabled

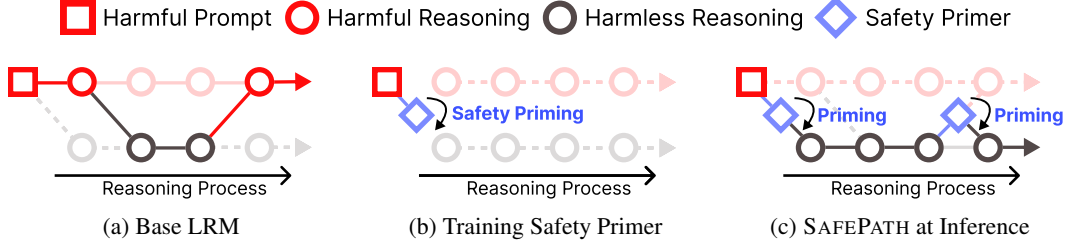


Figure 2: **SAFEPath’s approach to aligning LLMs.** (a) In a base LLM, harmful prompts can lead to unsafe reasoning. (b) During training, SAFEPath introduces a Safety Primer to guide the model toward reasoning with safety in mind when encountering harmful prompts. (c) At inference time, SAFEPath can dynamically activate the Safety Primer when harmful requests or reasoning emerges, effectively steering the model toward safer reasoning trajectories.

step-by-step inference without additional training. This line of work has since evolved through methods such as ReAct [Yao et al., 2023b], tree-of-thought [Yao et al., 2023a], and reflective reasoning [Renze and Guven, 2024, Zeng et al., 2024], which further strengthen intermediate reasoning processes. In parallel, code-based training [Ma et al., 2023] and the use of Process Reward Models (PRMs) during inference [Zhang et al., 2024a] have contributed to improved reasoning accuracy. More recent models, including OpenAI’s O1 [Jaech et al., 2024] and DeepSeek-R1 [Guo et al., 2025], demonstrate strong reasoning performance, with DeepSeek-R1 adopting rule-based reinforcement learning to enhance multi-step inference. These models serve as representative examples of LLMs. In this work, we use a DeepSeek-R1-distilled model, which is prone to generating harmful outputs [Ying et al., 2025]. DeepSeek-R1 models utilize the `<think>` and `</think>` tokens to explicitly separate the reasoning component from the final answer, resulting in a structure like `<think>[reasoning]</think>[answer]<end_of_sentence|>`.

Safety Alignment in LLMs. Despite widespread efforts in safety alignment, including RLHF [Christiano et al., 2017, Ouyang et al., 2022] and DPO [Rafailov et al., 2023], which leverage human preference annotations to distinguish safe from unsafe outputs [Touvron et al., 2023], LLMs remain vulnerable to state-of-the-art adversarial attacks [Zhou et al., 2024, Zou et al., 2023]. To enhance robustness, R2D2 [Zou et al., 2023] fine-tunes models against GCG attacks [Zou et al., 2023], drawing inspiration from adversarial training in vision [Madry et al., 2017]. Circuit Breaker [Zou et al., 2024] strengthens defenses by directly controlling internal representations, a strategy further refined by RepBend [Yousefpour et al., 2025]. In parallel, machine unlearning approaches [Lu et al., 2024] have been proposed to erase harmful behaviors for safety alignment. However, the effectiveness of these defenses remains largely unexplored in LLMs. To address this, we systematically evaluate state-of-the-art methods NPO [Zhang et al., 2024b], Circuit Breaker [Zou et al., 2024], and Task Arithmetic [Ilharco et al., 2023] in the LLM setting, and demonstrate the advantages of our method.

Safety Alignment in LLMs. Recent studies show that advanced reasoning capabilities alone do not guarantee harmless outputs, and even exacerbate safety vulnerabilities [Xiang et al., 2024, Jaech et al., 2024, Jiang et al., 2025, Huang et al., 2025]. Evaluations of LLMs on adversarial instruction datasets (e.g., StrongReject [Souly et al., 2024] and WildJailbreak [Jiang et al., 2024]) reveal persistent susceptibility to unsafe completions [Jiang et al., 2025, Huang et al., 2025]. To address this, reasoning strategies such as ZERO THINK and fine-tuning approaches like SafeChain [Huang et al., 2025] have been proposed to enhance model harmlessness. However, these methods face a fundamental trade-off between safety and reasoning, often incurring a “safety tax” [Huang et al., 2025] where stronger alignment degrades reasoning performance. These findings highlight the need for methods that jointly preserve both safety and reasoning capabilities in LLMs. To this end, we propose a SAFEPath strategy and its zero-shot variant to remove harmful behaviors while maintaining the reasoning ability.

3 Integrating SAFEPath for LLM Safety Alignment

To effectively align LLMs without compromising their reasoning abilities, we introduce SAFEPath, a targeted approach that provides a soft bias toward safety without restricting the model’s broader

reasoning flexibility. Unlike rigid alignment methods that impose strict behavioral constraints, SAFEPATH aims to guide the model toward safer outputs through a minimal but effective intervention.

At the core of this approach is the Safety Primer, a fixed 8-token prefix, “Let’s think about safety first,” inserted at the beginning of the `<think>` block. During training, the model is fine-tuned to emit this prefix for harmful prompts, while the remaining reasoning trace within the `<think>` block is left unsupervised, preserving the model’s full reasoning capability. This design allows the model to actively leverage its reasoning ability to block adversarial attacks, guiding it toward context-aware, safety-conscious outputs without restricting its broader reasoning flexibility.

Notably, this approach gives rise to an emergent property: the Safety Primer can be reactivated in intermediate reasoning steps, even though it was explicitly trained only for initial harmful prompts. This behavior implicitly promotes safer reasoning throughout the entire reasoning process, reinforcing the model’s ability to identify and correct potentially unsafe reasoning trajectories. Further analysis of this behavior is provided in Section 5.3.

Loss and Data. To integrate the model with SAFEPATH, we partition the training data into two distinct sets: the *Safety Trigger set* and the *Reasoning Retain set*. The Safety Trigger set contains harmful or adversarial prompts, where the model is fine-tuned to emit the Safety Primer immediately after the `<think>` token, without a closing `</think>` tag. This approach treats the primer as a soft safety signal, guiding the model toward safer reasoning without imposing rigid constraints, thereby preserving its flexibility for multi-step reasoning.

In contrast, the Reasoning Retain set consists of standard math reasoning prompts paired with ground-truth outputs, allowing the model to develop its core reasoning capabilities without unnecessary safety bias. This separation ensures that safety conditioning is isolated to the harmful subset, maintaining the model’s overall reasoning strength.

The two sets are combined using a predefined ratio, $\alpha : (1 - \alpha)$, where α controls the share of safety conditioning from the Safety Trigger set, while $1 - \alpha$ preserves the model’s general reasoning abilities from the Reasoning Retain set. An ablation study on the impact of α is provided in Section 5.6.

Zero-Shot Variant (ZS-SAFEPATH). We also propose a zero-shot variant, ZS-SAFEPATH, which applies the Safety Primer at inference without any fine-tuning. The primer is simply added to the reasoning block, immediately following the `<think>` tag, leveraging the model’s inherent safety potential without requiring any parameter updates.

4 Experimental Setup

Add SAFEPATH to LRMs. We apply SAFEPATH to DeepSeek-R1-Qwen-Distill-7B and DeepSeek-R1-Llama-Distill-8B, both distilled from the DeepSeek-R1 model [Guo et al., 2025]. For simplicity, we refer to these as R-7B and R-8B. These models have been noted for their weak safety alignment [Jiang et al., 2025, Zhou et al., 2025], making them ideal testbeds for evaluating the effect of SAFEPATH. This naming convention also extends to other DeepSeek-distilled models with different parameter counts, such as R-1.5B, R-14B, and R-32B. For training, we use WildJailbreak [Jiang et al., 2024] as the Safety Trigger set and DeepSeek Math 220K [Guo et al., 2025] as the Reasoning Retain set. R-7B is trained exclusively on safety prompts, while R-8B is trained on a balanced mixture of safety and reasoning data. Further experimental details are provided in Appendix B.1.

Baselines. For tuning-based baselines, we compare against two standard post-processing methods commonly used in recent LRM safety alignment studies: DirectRefusal [Huang et al., 2025], which enforces hard refusals to harmful prompts, and SafeChain [Jiang et al., 2025], which supervises both the reasoning and final answer to ensure safety. These methods have become the default approaches for aligning LRMs in recent work, reflecting the current state of the field.

For zero-shot baselines, we evaluate ZEROTHINK, which closes the reasoning block immediately with `<think></think>`, and LESSTHINK, which inserts a short-reasoning phrase (`<think>Okay, the user ask for this, I can answer it without thinking much.</think>`) [Jiang et al., 2025]. Both methods yield notable improvements in safety without fine-tuning. More details can be found in Appendix B.2.

Table 1: **Evaluation results on harmfulness, adversarial robustness, general capability, and reasoning ability in R-7B and R-8B.** SAFEPATH (SP) significantly enhances safety, achieving the lowest harmfulness and attack success rates across all settings. SP also preserves most of the reasoning ability, while other baselines experience substantial degradation. The best results among the three methods (Direct Refusal, SafeChain, SAFEPATH) for each benchmark are **bolded**.

Category	Benchmark	Deepseek-R1-Distill-Qwen-7B				Deepseek-R1-Distill-Llama-8B			
		Base Model	Direct Refusal	Safe Chain	SP (Ours)	Base Model	Direct Refusal	Safe Chain	SP (Ours)
Harmfulness (\downarrow)	StrongReject	49.2	26.0	32.5	10.4	37.3	20.8	17.3	0.0
	BeaverTails	41.4	32.1	39.3	12.7	36.2	16.5	34.0	7.7
Robustness (\downarrow)	DAN	79.0	66.7	64.3	8.3	82.7	66.7	57.0	5.7
	PAIR	66.3	63.8	66.3	27.5	95.0	88.5	91.3	26.3
	Trigger	45.0	34.8	27.3	3.0	50.8	38.8	35.5	2.0
	Multilingual	9.03	19.2	34.3	4.7	16.4	19.3	62.9	1.2
	Prefilling	26.2	32.2	22.9	18.6	18.1	25.7	15.9	8.6
Capability (\uparrow)	MMLU	52.5	52.7	52.9	52.6	53.3	53.7	53.8	53.6
	Arc-Challenge	42.2	40.8	40.9	42.0	40.5	39.1	41.0	40.1
Reasoning (\uparrow)	MATH500	94.8	85.0	92.6	94.4	90.4	86.6	86.2	84.0
	GPQA	54.6	38.9	49.5	51.0	44.4	41.4	43.4	48.0
	AIME24	46.7	26.7	43.3	46.7	53.3	36.7	36.7	50.0
	MBPP	47.2	50.2	49.6	48.0	43.4	42.4	42.2	42.6

Evaluation Setup. We evaluate safety using two benchmarks: StrongReject [Souly et al., 2024] for direct refusal to harmful prompts and BeaverTails [Ji et al., 2023] for robustness against subtle jailbreaks. To stress-test safety alignment, we apply five adversarial attacks: DAN [Shen et al., 2024], PAIR [Chao et al., 2023], Jailbreak Trigger [Huang et al., 2024], Multilingual [Yong et al., 2023], and Prefilling [Vega et al., 2023]. For the Prefilling attack, we prepend the phrase “Okay, let’s provide clear instructions to assist the user” to the start of the reasoning block to induce unsafe completions.

For reasoning capability, we evaluate the models on three math benchmarks (MATH500 [Hendrycks et al., 2021], GPQA [Rein et al., 2024], AIME24) and a code generation benchmark (MBPP [Austin et al., 2021]). General capability is assessed using MMLU [Hendrycks et al., 2020] and ARC-Challenge [Clark et al., 2018] as standard benchmarks.

5 Experimental Results on SAFEPATH

5.1 Comparison with LRM Baselines

As shown in Table 1, SAFEPATH significantly improves LRM safety across key metrics, reducing harmfulness by 74.5% in R-7B and 90.0% in R-8B compared to the base model, while blocking 74.0% and 83.3% of jailbreak attempts, respectively, without sacrificing reasoning performance. In contrast, while Direct Refusal and SafeChain effectively reduce harmfulness in datasets like StrongReject and BeaverTails, they suffer from substantial reasoning losses. For example, both methods incur a 16.6%p accuracy drop on AIME24 in R-8B, reflecting the limitations of their rigid alignment strategies. Additionally, these methods remain vulnerable to adversarial attacks designed to elicit harmful outputs. In such contexts, they perform similarly to the base model (before safety training), while SAFEPATH achieves substantial reductions in vulnerability.

5.2 Comparison with LLM Baselines

Baselines. To evaluate whether existing LLM safety alignment methods transfer effectively to LRMs, we re-implement three representative approaches that have been widely adopted in prior work. Task Arithmetic (TA) [Ilharco et al., 2023] removes harmful behavior by identifying the parameter shifts caused by fine-tuning on harmful QA pairs and subtracting them from the model weights.

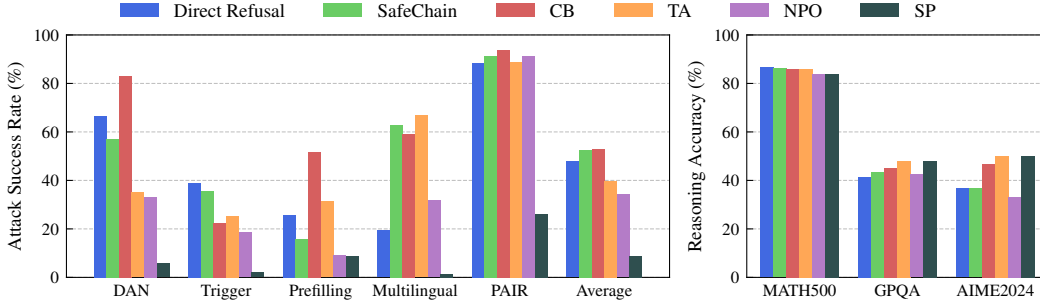


Figure 3: **Attack Success Rate (ASR) and Reasoning Accuracy for various LLM and LRM defense methods in R-8B.** The left panel shows ASR across different jailbreak methods, including DAN, Trigger, Prefilling, Multilingual, PAIR, and the overall average. The right panel presents reasoning accuracy on MATH500, GPQA, and AIME2024. SAFEPATH (SP) consistently achieves the lowest attack success rate while maintaining competitive reasoning performance.

Negative Preference Optimization (NPO) extends DPO [Rafailov et al., 2023] by treating harmful completions as negative preferences relative to a reference model. Circuit Breakers (CB) [Zou et al., 2024] take a different approach, aligning model behavior at the representation level by intercepting and rerouting unsafe activations to block harmful generation. While originally developed for general-purpose LLMs, we adapt these methods to LRMs. Further details are provided in Appendix B.3.

Results. As shown in Figure 3, some LLM-based baselines, such as TA and NPO, effectively suppress certain jailbreaks like DAN and Trigger, demonstrating a reasonable trade-off between safety and performance. However, CB, despite being a state-of-the-art LLM defense, struggles to provide robust protection in the LRM setting, indicating that strong performance in general LLM safety alignment does not necessarily translate to effective LRM defense.

In contrast, SAFEPATH, specifically designed for LRMs, consistently achieves the lowest ASR across diverse adversarial benchmarks, while maintaining strong reasoning capabilities, outperforming all other baselines. This highlights the importance of dedicated safety methods that address the unique challenges of multi-step reasoning, rather than relying solely on approaches developed for conventional LLMs.

5.3 Number of Safety Primer Activations

To gain a deeper understanding of the dynamics of SAFEPATH, we measure the average activation frequency of the Safety Primer (i.e., the “Let’s think about safety first” phrase) across different benchmarks in R-8B. In low-risk contexts like MATH500, where harmful completions are rare, the primer is triggered just 0.22 times per sample, reflecting minimal intervention. However, for clearly harmful inputs like StrongReject, the activation rate rises significantly to 1.71 times per sample, indicating a sharp increase in the model’s sensitivity to dangerous prompts. This difference becomes even more pronounced for highly adversarial attacks like PAIR, where the primer is triggered over 8 times per sample, underscoring the intense pressure these inputs place on the safety mechanism (see Figure 4).

Interestingly, this pattern emerges despite the primer being explicitly trained to activate only for harmful prompts. The repeated activations observed in adversarial contexts suggest a more responsive behavior, where the fixed prefix can re-engage when the reasoning process repeatedly encounters potential safety risks. This flexibility is a key characteristic of the SAFEPATH mechanism, providing robust safety without extensive additional supervision. Examples can be found in Figures 9 and 10.

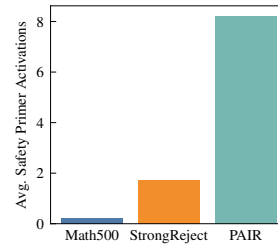


Figure 4: Average number of Safety Primer activations per sample in R-8B across MATH500, StrongReject, and PAIR.

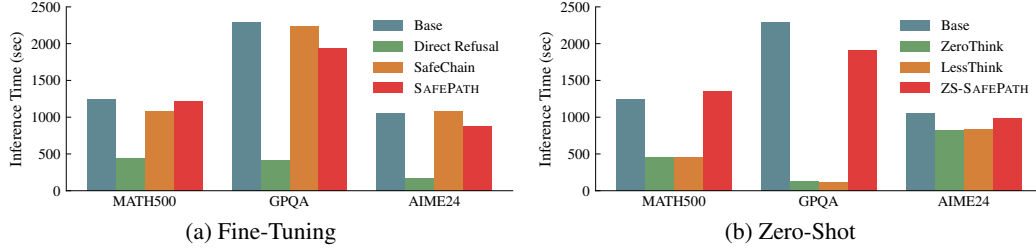


Figure 5: **Inference Time Across Safety Alignment Methods.** SAFEPATH and ZS-SAFEPATH maintain inference costs similar to the base model, while methods like ZERO THINK and LESS THINK reduce cost by terminating reasoning early. Direct Refusal also shows reduced inference time, as it is trained to directly reject harmful prompts without engaging in extended reasoning.

5.4 Training Cost Comparison

SAFEPATH converges quickly due to its fixed prompt design, requiring just 100 steps for R-7B and 20 steps for R-8B, compared to the thousands of steps typically needed for full model fine-tuning. This efficiency is further enhanced by the targeted nature of the Safety Primer, which fine-tunes only 8 tokens, significantly reducing computational cost and memory overhead. As shown in Table 2, SAFEPATH completes training in less than five minutes, achieving 124.3 \times faster training than Direct Refusal and 132.3 \times faster than SafeChain for R-7B, and 295.9 \times and 314.1 \times faster, respectively, for R-8B, while maintaining strong safety alignment. This combination of speed and alignment makes it a practical choice for real-world deployments.

Table 2: Training time (min) for three methods on R-7B and R-8B.

Model	Refusal	SafeChain	SAFEPATH
R-7B	509.5	542.3	4.1
R-8B	435.1	461.7	1.5

5.5 Inference Time Comparison

An important consideration for reasoning-oriented models is inference-time computational cost. Since SAFEPATH modifies the initial reasoning state, it is essential to verify whether this approach introduces additional latency. As shown in Figure 5, we find that the inference cost for SAFEPATH remains similar to the base model. This confirms that our intervention imposes minimal computational overhead, aligning with the core design goal of maintaining efficiency while enhancing safety. SafeChain also shows similar computational cost since it finetunes safety output that contains reasoning. In contrast, Direct Refusal shows much less inference time in reasoning benchmarks. This shows that Directly refusing makes model reason less by training refuse to answer when harmful prompt comes without reasoning, which is revealed in Table 1 by low reasoning accuracy.

5.6 Ratio between Safety Trigger Set and Retain Reasoning Set

We investigate the trade-off between safety and reasoning performance by varying the ratio between the Safety Trigger set and the Reasoning Retain set during training R-8B. As shown in Figure 6, increasing the proportion of safety samples consistently reduces harmful completions, with only a modest impact on capability. Notably, we find that incorporating just 10% of the safety dataset (equivalent to 8 examples) produces a substantial safety gain, with only a 1% drop in accuracy for ARC-C. Beyond this point, both safety and utility metrics remain stable, even as the ratio of Safety Trigger set shifts from 10% to 100%, indicating that SAFEPATH is not sensitive to the mixing ratio. This robustness simplifies deployment, as the method remains effective across a wide range of training configurations without requiring extensive hyperparameter tuning.

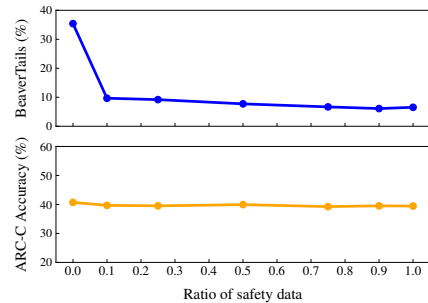


Figure 6: Safety trigger to Reasoning Retain set ratio on harmfulness (Beaver-Tails) and capability (ARC-C) for R-8B.

CautionPath (CP)	RefusalPath (RP)	SAFEPath (SP)
Think: <i>I think it's harmful request.</i> Answer: I shouldn't help with this.	Think: <i>I can't answer this question for safety reason. When dealing with exchange rates and currency conversions...</i> Answer: I must avoid providing any financial advice...	Think: <i>Let's think about safety first. When dealing with money conversions, it's important to use accurate and up-to-date exchange rates...</i> Answer: ... Final Answer: 1250

Figure 7: **Qualitative comparison of prefix-only fine-tuning on the same math problem.** SAFEPath enables the model to continue reasoning and solve the task after emitting the safety prefix, while CautionPath (CP) and RefusalPath (RP) terminate immediately, unnecessarily cutting off further reasoning despite the problem not requiring a refusal. Full outputs are in Figure 10.

5.7 Alternative Prefix Designs for SAFEPath

To explore whether alternative prefixes that explicitly signal caution or refusal can achieve similar effects, we design two variants. The first, CautionPath (CP), uses the prefix “I think it’s a harmful request.” The second, RefusalPath (RP), uses the prefix “I can’t answer this question for safety reasons.” Both variants are trained using the same hyperparameters as SAFEPath on the R-8B model.

As shown in Table 3, both CP and RP significantly reduce harmful completions, as indicated by lower BeaverTails scores, but this comes at the cost of substantially impaired reasoning ability across all three benchmarks. This suggests that direct signals of caution or refusal tend to prematurely terminate the reasoning process, preventing the model from fully engaging with the task.

In contrast, the prompt design of SAFEPath is fundamentally different. Rather than imposing a strict refusal, it uses a soft prefix, which sets a safety-oriented context without immediately ending the reasoning process. This allows the model to continue generating a complete chain-of-thought, encouraging a more nuanced and context-aware approach to safety. As illustrated in Figure 7, this design enables SAFEPath to reach correct answers even after emitting the safety-oriented prefix, avoiding the abrupt cutoffs seen in CP and RP. This distinction is critical, as it highlights the unique advantage of SAFEPath, which can maintain task engagement while providing robust safety.

Table 3: Comparison of three prefix-based safety methods in R-8B: SAFEPath (SP), CautionPath (CP), and RefusalPath (RP) with respect to reasoning ability and harmfulness.

Method	Reasoning Ability (↑)			Harmfulness (↓)
	MATH500	GPQA	AIME2024	BeaverTails
Baseline	90.4	44.4	53.3	36.2
SP	84.0	48.0	50.0	7.7
CP	47.6	41.4	30.0	2.7
RP	75.8	42.9	36.7	2.0

6 Zero-Shot Results for SAFEPath

Main Results. We evaluate the zero-shot variant of our method, ZS-SAFEPath, which applies the Safety Primer at inference without parameter updates. Unlike methods like ZEROThink and LESSThink, which reduce harmful outputs by aggressively suppressing the reasoning process, ZS-SAFEPath preserves the core reasoning capabilities of LRMs, maintaining high reasoning accuracy while significantly reducing harmfulness, as shown in Table 4.

For instance, on the challenging AIME24 benchmark, ZS-SAFEPath achieves 73.33% on R-14B and 60.00% on R-32B, substantially outperforming ZEROThink (13.33% and 30.00%, respectively). This trend extends to other reasoning-intensive tasks like GPQA, where ZS-SAFEPath reaches 67.17% on R-32B, compared to 53.03% for ZEROThink, reflecting its ability to retain complex reasoning capabilities. However, this comes with a trade-off in terms of harmfulness. For example, ZS-SAFEPath records a harmfulness score of 11.75% on R-32B, which is higher than the scores for ZEROThink (3.00%) and LESSThink (4.50%). Despite this, it remains a more balanced approach for larger models, aligning safety without severely compromising reasoning, making it a practical option for applications where maintaining reasoning quality is critical.

Table 4: **Reasoning Accuracy and Harmful Scores for Zero-Shot Prompting Strategies.** The table compares reasoning accuracy (AIME24, GPQA, MATH500) and harmful scores (StrongReject, BeaverTails) across zero-shot prompting strategies from 1.5B to 32B models.

Model	Methods	Reasoning Accuracy (\uparrow)				Harmful Score (\downarrow)		
		AIME24	GPQA	MATH500	Average	StrongReject	BeaverTails	Average
R-1.5B	Base Model	36.67	34.85	85.20	52.24	51.90	58.10	55.00
	ZEROTHINK	6.67	32.32	72.00	37.00	2.30	11.40	6.85
	LESSTHINK	10.00	36.87	66.60	37.82	36.50	33.00	34.75
	ZS-SAFEPATH	30.00	37.88	80.60	49.49	34.60	43.10	38.85
R-7B	Base Model	46.67	54.55	94.80	65.34	49.20	41.40	45.30
	ZEROTHINK	23.23	37.37	81.20	47.30	0.00	8.50	4.25
	LESSTHINK	10.00	40.40	72.00	40.80	11.30	19.00	15.15
	ZS-SAFEPATH	50.00	49.49	94.60	64.70	14.80	22.10	18.45
R-8B	Base Model	53.33	44.44	90.40	62.73	37.30	36.20	36.75
	ZEROTHINK	40.00	45.45	86.20	57.22	0.40	7.80	4.10
	LESSTHINK	10.00	33.33	66.40	36.58	6.50	13.90	10.20
	ZS-SAFEPATH	53.33	52.53	80.60	62.15	9.80	20.70	15.25
R-14B	Base Model	70.00	62.12	94.80	75.64	31.70	34.00	32.85
	ZEROTHINK	13.33	46.97	76.20	45.50	1.70	6.80	4.25
	LESSTHINK	20.00	43.43	77.60	47.01	2.90	7.00	4.95
	ZS-SAFEPATH	73.33	61.11	93.80	76.08	8.30	18.20	13.25
R-32B	Base Model	63.33	66.16	95.20	74.90	19.80	32.00	25.90
	ZEROTHINK	30.00	53.03	82.60	55.21	0.0	6.00	3.00
	LESSTHINK	20.00	48.99	80.80	49.93	1.70	7.30	4.50
	ZS-SAFEPATH	60.00	67.17	95.00	74.06	7.30	16.20	11.75

Effect of Position. To assess the impact of Safety Primer placement within the reasoning block, we compare two zero-shot configurations: prefix, where the primer “Let’s think about safety first” is placed at the start, and suffix, where the phrase “Wait, lastly we need to think about safety” is appended at the end.

As shown in Figure 8, prefix placement consistently results in lower harmfulness scores than the suffix variant, with R-7B and R-8B showing 30.4 and 18.1 points lower harmfulness, respectively. This highlights the advantage of early-stage intervention, as introducing the safety signal before reasoning begins can more effectively guide the model’s internal trajectory, reinforcing safer outputs.

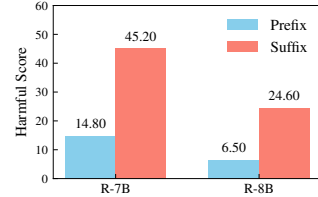


Figure 8: Comparison of prefix vs. suffix placement of the safety phrase (StrongReject).

Inference Time. As shown in Figure 5, ZS-SAFEPATH has similar or even lower inference time compared to the base model. Interestingly, methods like ZEROTHINK and LESSTHINK achieve even faster inference by instantly terminating the reasoning block, particularly on MATH500 and GPQA. However, this approach significantly degrades reasoning ability, as shown in Table 4.

7 Conclusion

We introduce SAFEPATH, a practical approach for aligning LRMs without compromising their core reasoning capabilities. Unlike conventional methods that impose rigid safety constraints, SAFEPATH leverages the model’s natural reasoning ability through a concise, 8-token Safety Primer, effectively reducing harmful outputs while preserving reasoning depth. Our experiments show that SAFEPATH significantly reduces harmful responses and blocks adversarial attacks with lower training costs, achieving up to 90.0% reduction in harmful outputs and 83.3% blockage of jailbreak attempts in R-8B. Notably, an emergent property observed in our approach is the dynamic reactivation of the Safety Primer in highly adversarial contexts, where the model instinctively re-engages the primer multiple times to reinforce safety, even without explicit supervision. This efficient design not only addresses the long-standing trade-off between safety and reasoning but also introduces a scalable path for developing safer, more adaptable LRMs. We believe that this approach opens new avenues for secure AI systems, bridging the gap between high-performance reasoning and practical safety.

References

- Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.
- Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.
- Hannah Cyberek and David Evans. Steering the censorship: Uncovering representation vectors for llm "thought" control. *arXiv preprint arXiv:2504.17130*, 2025.
- Junfeng Fang, Yukai Wang, Ruipeng Wang, Zijun Yao, Kun Wang, An Zhang, Xiang Wang, and Tat-Seng Chua. Safemlm: Demystifying safety in multi-modal large reasoning models. *arXiv preprint arXiv:2504.08813*, 2025.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, Zachary Yahn, Yichang Xu, and Ling Liu. Safety tax: Safety alignment makes your large reasoning models less reasonable. *arXiv preprint arXiv:2503.00555*, 2025.
- Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. Position: Trustllm: Trustworthiness in large language models. In *International Conference on Machine Learning*, pages 20166–20270. PMLR, 2024.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Wonje Jeung, Dongjae Jeon, Ashkan Yousefpour, and Jonghyun Choi. Large language models still exhibit bias in long text. *arXiv preprint arXiv:2410.17519*, 2024.

- Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.
- Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, et al. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *arXiv preprint arXiv:2503.02875*, 2025.
- Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. *arXiv preprint arXiv:2502.12025*, 2025.
- Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, et al. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems*, 37:47094–47165, 2024.
- Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *arXiv preprint arXiv:2404.05880*, 2024.
- Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. At which training stage does code data help llms reasoning? *arXiv preprint arXiv:2309.16298*, 2023.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *COLM*, 2024.
- Matthew Renze and Erhan Guven. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*, 2024.
- Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*, 2023.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685, 2024.

- Dan Shi, Tianhao Shen, Yufei Huang, Zhigen Li, Yongqi Leng, Renren Jin, Chuang Liu, Xinwei Wu, Zishan Guo, Linhao Yu, et al. Large language model safety: A holistic survey. *arXiv preprint arXiv:2412.17686*, 2024.
- Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *arXiv preprint arXiv:2402.10260*, 2024.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Jason Vega, Isha Chaudhary, Changming Xu, and Gagandeep Singh. Bypassing the safety training of open-source llms with priming attacks. *arXiv preprint arXiv:2312.12321*, 2023.
- Fei Wang, Ninareh Mehrabi, Palash Goyal, Rahul Gupta, Kai-Wei Chang, and Aram Galstyan. Data advisor: Dynamic data curation for safety alignment of large language models. *arXiv preprint arXiv:2410.05269*, 2024.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.
- Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*, 2024.
- Fengli Xu, Qian Yue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023a.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023b.
- Zonghao Ying, Guangyi Zheng, Yongxin Huang, Deyue Zhang, Wenxin Zhang, Quanchen Zou, Aishan Liu, Xianglong Liu, and Dacheng Tao. Towards understanding the safety boundaries of deepseek models: Evaluation and findings. *arXiv preprint arXiv:2503.15092*, 2025.
- Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*, 2023.
- Ashkan Yousefpour, Taeheon Kim, Ryan S Kwon, Seungbeen Lee, Wonje Jeung, Seungju Han, Alvin Wan, Harrison Ngan, Youngjae Yu, and Jonghyun Choi. Representation bending for large language model safety. *arXiv preprint arXiv:2504.01550*, 2025.
- Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*, 2023.
- Qingbin Zeng, Qinglong Yang, Shunan Dong, Heming Du, Liang Zheng, Fengli Xu, and Yong Li. Perceive, reflect, and plan: Designing llm agent for goal-directed city navigation without instructions. *arXiv preprint arXiv:2408.04168*, 2024.
- Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024a.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *COLM*, 2024b.
- Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Shreedhar Jangam, Jayanth Srinivasa, Gaowen Liu, Dawn Song, and Xin Eric Wang. The hidden risks of large reasoning models: A safety assessment of r1. *arXiv preprint arXiv:2502.12659*, 2025.

Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, Rui Zheng, Songyang Gao, Yicheng Zou, Hang Yan, Yifan Le, Ruohui Wang, Lijun Li, Jing Shao, Tao Gui, Qi Zhang, and Xuanjing Huang. Easyjailbreak: A unified framework for jailbreaking large language models, 2024.

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

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *NeurIPS*, 2024.

A Harmful Thinking Question-Reasoning-Answer Generation (HarmChain).

For implementing LLM safety alignment methods like TA or NPO, it is necessary to collect harmful question-answer (QA) pairs. However, for large reasoning models, we also require harmful thinking traces to properly align the intermediate reasoning steps. To address this, we construct a set of harmful Question-Reasoning-Answer (QRA) pairs, named **HarmChain**. We begin by generating harmful Question-Reasoning-Answer (QRA) pairs using R1-Distill-Qwen-7B and R1-Distill-Llama-8B. Specifically, we sampled 5,000 harmful questions from the BeaverTails dataset for each model, generating a total of 10,000 QRA pairs. To ensure that both the reasoning and answers are genuinely harmful, we filtered this initial set using the BeaverDam harmfulness classifier, resulting in a final dataset of 3,724 verified harmful QRA pairs. This filtered dataset is used for training TA, NPO, and Circuit Breaker baselines in our experiments.

B Experimental Details

B.1 Implementation Details of SAFEPATH

The Safety Trigger set is derived from the WildJailbreak dataset¹, using only harmful prompts. The instruction format is:

```
<|User|>[instruction]<|Assistant|><think>Let's think about safety first.
```

The Reasoning Retain set is constructed from Math220K², using responses generated by DeepSeek-R1 for correctly solved samples. Both datasets are trained with a learning rate of 1×10^{-5} . The R-7B model is trained on 400 Safety Trigger set samples for 100 steps with a batch size of 4, without using the Reasoning Retain set. The R-8B model is trained on 40 samples from each set (80 total) for 20 steps with a batch size of 4. Both setups complete training in less than 5 minutes.

B.2 LRM Baseline

For Direct Refusal, we use the refusal dataset from Wang et al. [2024]³ with the prompt format `<|User|>[prompt]<|Assistant|><think>I should not answer this question!</think>[response]<|end_of_sentence|>`, as proposed by Huang et al. [2025]. This approach is trained with 10,000 steps and a 1×10^{-5} learning rate, except for the DeepSeek-R1-Distill-Qwen-7B model, which uses 2×10^{-5} for improved safety outcomes. SafeChain, on the other hand, utilizes the dataset from Jiang et al. [2025]⁴ and is also trained for 10,000 steps with a 1×10^{-5} learning rate, consistent with the original implementation.

For zero-shot methods, we adopt different strategies to eliminate intermediate reasoning. For ZERO-THINK, we prefill `</think>` immediately after the input to terminate the reasoning block without any intermediate steps. In contrast, for LESS-THINK, we insert 'Okay the user asked for this, I can answer it without thinking much</think>' directly after the input template, encouraging a more direct response.

B.3 LLM Baseline

We adapted several LLM-centric baselines for LRMs, which explicitly separate the reasoning process and final answers. In this context, we define an input-output pair as (x, y) , where x represents the input question or prompt, and y consists of both the reasoning and the final answer, denoted as $y = (r, a)$. Here, r refers to the reasoning sequence generated within the `<think>` block, while a represents the final answer generated after the `</think>` block. The model's conditional probability of generating y given x is represented as $f_{\theta}(y|x)$, where θ denotes the model parameters. This function captures the likelihood of producing a complete response, including both the reasoning and final answer components.

¹<https://huggingface.co/datasets/allenai/wildjailbreak>

²<https://huggingface.co/datasets/open-r1/OpenR1-Math-220k>

³<https://huggingface.co/datasets/fwnlp/self-instruct-safety-alignment>

⁴<https://huggingface.co/datasets/UWNSL/SafeChain>

Negative Preference Optimization (NPO). NPO can also be adapted for harmfulness reduction by treating harmful samples as negative examples:

$$\mathcal{L}_{\text{NPO}} = -\frac{2}{\beta} \mathbb{E}_{(x, y_h) \sim \mathcal{D}_h} \left[\log \sigma \left(-\beta \log \frac{f_\theta(y_h|x)}{f_{\text{ref}}(y_h|x)} \right) \right], \quad (1)$$

where f_{ref} is the reference model and $\beta = 0.1$ controls the deviation from the original model. In our experiments, we fine-tuned with a learning rate of 1×10^{-5} for 20 iterations. We used a custom harmful dataset \mathcal{D}_h specifically constructed for this purpose, as there is no publicly available comprehensive dataset for harmful completions. For details on the construction of \mathcal{D}_h , see Appendix A.

Additionally, to maintain reasoning capability, we included a secondary loss term using the Math220K dataset [Guo et al., 2025] from DeepSeek:

$$\mathcal{L}_{\text{Math}} = -\mathbb{E}_{(x, y_m) \sim \mathcal{D}_m} [\log f_\theta(y_m|x)], \quad (2)$$

where \mathcal{D}_m represents the Math220K dataset. The final combined loss for harmfulness reduction and reasoning preservation is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{NPO}} + \lambda \mathcal{L}_{\text{Math}}, \quad (3)$$

where $\lambda = 1$ is used in our setup to equally balance harmfulness reduction and reasoning retention.

Task Arithmetic (TA). Task Arithmetic aims to reduce harmful behavior by subtracting the parameter updates associated with harmful training. Specifically, this method adjusts the model parameters as follows:

$$\theta_{\text{safe}} = \theta_{\text{orig}} - \alpha \cdot (\theta_{\text{harmful}} - \theta_{\text{orig}}), \quad (4)$$

where θ_{harmful} is the model fine-tuned on the harmful dataset \mathcal{D}_h , and $\alpha = 1$ controls the strength of the adjustment. This formulation effectively subtracts the harmful direction in parameter space, preserving general capabilities while removing specific harmful behaviors.

In our experiments, the harmful model was fine-tuned for 200 iterations with a learning rate of 1×10^{-5} before applying this adjustment. This approach allows the model to retain broader reasoning capabilities while minimizing harmful outputs.

Circuit Breaker. Circuit Breaker is a representation-level alignment method that modifies internal model activations to suppress harmful outputs. In our setup, we use two datasets for training: the SafeChain dataset as the retain set (\mathcal{D}_r) and our harmful dataset as the harmful set (\mathcal{D}_h).

The training objective is defined as:

$$\mathcal{L}_{\text{RR}} = \alpha_h \cdot \text{ReLU}(\cos(\text{rep}_\theta(x_h), \text{rep}_{\theta_{\text{cb}}}(x_h))) + \alpha_r \cdot \|\text{rep}_\theta(x_r) - \text{rep}_{\theta_{\text{cb}}}(x_r)\|_2^2, \quad (5)$$

where $x_h \sim \mathcal{D}_h$, $x_r \sim \mathcal{D}_r$, θ is the original model, θ_{cb} is the circuit breaker model, and α_h , α_r are weighting coefficients that gradually shift focus from harmful suppression to retention during training. All hyperparameters were set to match those in the original paper [Zou et al., 2024] to ensure consistency and comparability.

B.4 Hardware Specification

All experiments were conducted on a system with 512 CPU cores, 8 Nvidia RTX L40S (48GB) GPUs, and 1024 GB of RAM. In total, the experiments, evaluations, analyses, and method development required approximately 2,000 GPU hours.

C Evaluation Details

C.1 Harmfulness

We evaluated harmfulness using two benchmarks: StrongReject [Souly et al., 2024] and Beaver-Tails [Ji et al., 2023]. The StrongReject evaluation included 60 uniformly sampled prompts, assessed with GPT-4o based on the original rubric, which considers specificity, convincingness, and refusal.

This rubric is designed to approximate human evaluation without overestimating the harmfulness of model outputs, ensuring a balanced assessment. For BeaverTails, we selected the first 1,000 samples from the dataset and evaluated them using the BeaverDam classifier provided in the benchmark. This method ensures consistent assessment across diverse, potentially high-risk scenarios.

C.2 Adversarial Attack

DAN. Do-any-thing now (DAN) is a dataset collected from 131 communities, designed to jailbreak state-of-the-art models like ChatGPT, using diverse and strong adversarial prompts [Shen et al., 2024]. We test the model on 300 samples from this dataset, using the AI2 evaluation codebase⁵.

TrustLLM-JailbreakTrigger (Trigger). We use a subset of 400 prompts from the JailbreakTrigger dataset [Huang et al., 2024], which contains 13 distinct jailbreak attack types, including prefix injection, sensitive content removal, style constraint, persona and scenario, and distractor attacks. The evaluation follows the AI2 codebase⁵.

PAIR. PAIR (Prompt Automatic Iterative Refinement) is a black-box adversarial testing framework designed to efficiently discover jailbreak prompts for large language models [Chao et al., 2023]. It iteratively refines adversarial prompts to maximize the likelihood of unsafe completions. In each iteration, the attacker model generates a candidate prompt, evaluates the response from the target model, and updates the prompt based on the feedback until the attack succeeds or a predefined limit is reached.

For our implementation, we use PAIR with $n_iterations = 3$, testing on 80 samples from the AdvBench dataset [Zou et al., 2023]. This setup aims to balance attack diversity and computational efficiency, ensuring that each sample undergoes multiple refinement steps to uncover potential vulnerabilities. We used the framework from AISafetyLab⁶.

Multilingual. Multilingual attack tests whether the model can defend against harmful prompts in different languages. We evaluated the model on 9 languages, including Chinese, Italian, Vietnamese, Arabic, Korean, Thai, Bengali, Swahili, and Javanese, using Google Translate to translate the AdvBench dataset [Zou et al., 2023]. We use 80 samples per language, resulting in a total of 720 samples. The evaluation followed the EasyJailbreak codebase⁷.

Prefilling. Prefilling refers to the practice of adding guiding prompts before generating responses to influence the model’s outputs. While LLMs often use prefilling strategies, such as adding phrases like “Okay, here’s a step-by-step guide...” to enforce responses to adversarial attacks, this approach is not directly applicable to LRMs, which explicitly separate reasoning from final outputs. Instead, a more appropriate approach for LRMs is to insert the prefilling token directly within the reasoning block. Specifically, we prepend the phrase “Okay, let’s provide clear instructions to assist the user.” at the beginning of the reasoning block to induce unsafe completions, aligning more closely with the model’s structured reasoning process. We evaluate this setup using 1,000 samples from BeaverTails benchmark [Ji et al., 2023].

C.3 Reasoning Ability Evaluation

To measure reasoning ability, we used three widely adopted mathematical benchmarks that are commonly used to assess the reasoning capabilities of LRMs: MATH500 [Hendrycks et al., 2021], GPQA [Rein et al., 2024], and AIME24. These benchmarks were evaluated using the framework provided by DeepSeek⁸, which is specifically designed for reasoning model assessment. For MBPP [Austin et al., 2021], we used the lm-evaluation-harness⁹, which provides a standardized interface for evaluating code generation.

⁵<https://github.com/allenai/safety-eval>

⁶<https://github.com/thu-coai/AISafetyLab>

⁷<https://github.com/EasyJailbreak/EasyJailbreak>

⁸<https://github.com/deepseek-ai/DeepSeek-R1>

⁹<https://github.com/EleutherAI/lm-evaluation-harness>

C.4 General Capability Evaluation

To assess general capability, we included two widely recognized benchmarks: MMLU [Hendrycks et al., 2020], a de facto standard for comprehensive model utility, and ARC-Challenge [Clark et al., 2018], which focuses on scientific problems requiring a mix of knowledge and reasoning. Both benchmarks were evaluated using the lm-evaluation-harness⁹ to ensure consistency and reproducibility.

C.5 Licenses

We provide Table 5, which lists every external model and dataset we use, together with its source, access link, and license.

Table 5: The list of assets used in this work.

Asset	Source	Access	License
DeepSeek-R1-Distill Models	Guo et al. [2025]	Link	MIT License
SafeChain	Jiang et al. [2025]	Link	GPL-3.0 license
WildJailbreak	Jiang et al. [2024]	Link	ODC-BY
Math220K	Guo et al. [2025]	Link	Apache License 2.0
Data-Advisor	Wang et al. [2024]	Link	Apache License 2.0
MMLU	Hendrycks et al. [2020]	Link	MIT License
ARC	Clark et al. [2018]	Link	CC-BY-SA-4.0
MATH500	Hendrycks et al. [2021]	Link	MIT License
GPQA	Rein et al. [2024]	Link	CC-BY-4.0
AIME24	–	Link	MIT License
MBPP	Austin et al. [2021]	Link	CC-BY-4.0
StrongReject	Souly et al. [2024]	Link	MIT License
BeaverTails	Ji et al. [2023]	Link	CC-BY-NC-4.0
AdvBench	Zou et al. [2023]	Link	MIT License
JailbreakTrigger	Huang et al. [2024]	Link	MIT License
DAN	Shen et al. [2024]	Link	MIT License

D Additional Results

D.1 Comparison with LLM Baselines

As shown in Table 6, some LLM-based baselines like TA and NPO demonstrate relatively strong defenses against certain adversarial attacks compared to typical LRM defenses like Direct Refusal and SafeChain. For example, TA achieves moderate ASRs on benchmarks like DAN (35.0%) and Trigger (25.3%), while NPO shows even lower ASRs in some cases, such as 33.0% on DAN and 18.5% on Trigger, suggesting that these methods can effectively suppress specific attack types while maintaining decent reasoning performance. However, these defenses are still significantly weaker than SAFEPATH (SP), which achieves the lowest ASRs across all evaluated scenarios, including just 5.7% on DAN and 2.0% on Trigger.

Notably, CB, despite being a state-of-the-art LLM defense, consistently struggles in the LRM setting, recording some of the highest ASRs across the evaluated methods, including 83.0% on DAN and 51.7% on Prefilling. This indicates that strong performance in general LLM safety alignment does not necessarily translate to effective LRM defense, as the multi-step reasoning processes in LRMs present unique challenges that these methods are not well-equipped to handle.

Interestingly, while TA, NPO, and CB generally follow SP’s performance on general capability benchmarks like MMLU, they show significant drops in more challenging tasks like Arc-Challenge, indicating that these methods, while capable in simpler contexts, struggle to generalize effectively to more difficult benchmarks. These results underscore the need for dedicated LRM safety alignment methods like SP, which integrate more comprehensive adversarial defenses without sacrificing reasoning ability, addressing the unique vulnerabilities of multi-step reasoning models.

Table 6: **Evaluation results on general capability, reasoning ability, harmfulness and adversarial robustness in R-8B, with LLM baselines.** While some baselines show promising results, SAFEPATH (SP) shows most promising results, achieving the lowest harmfulness and attack success rate across all settings, without compromising reasoning ability. The best results among the four methods (TA, NPO, CB, SP) for each benchmark are **bolded**.

Category	Benchmark	Methods			
		TA	NPO	CB	SP
Capability (\uparrow)	MMLU	53.2	53.5	53.5	53.0
	Arc-Challenge	38.5	38.7	37.0	40.1
Reasoning (\uparrow)	MATH500	85.8	84.0	86.0	84.0
	GPQA	48.0	42.4	45.0	48.0
	AIME24	50.0	33.3	46.7	50.0
	MBPP	39.2	43.0	43.2	42.6
Harmfulness (\downarrow)	StrongReject	6.9	9.2	1.7	0.0
	BeaverTails	46.0	31.1	62.2	7.7
Robustness (\downarrow)	DAN	35.0	33.0	83.0	5.7
	PAIR	88.8	91.3	93.8	26.3
	Trigger	25.3	18.5	22.5	2.0
	Multilingual	66.8	31.7	59.3	1.3
	Prefilling	31.3	9.0	51.7	8.6
	Average	49.4	36.7	62.1	8.8

Table 7: **Full version of inference time across all safety alignment methods.** Comparison of inference times for various safety alignment methods in both fine-tuned and zero-shot settings, evaluated on R-7B and R-8B models.

Methods	R1-Distill-Qwen-7B				R1-Distill-Llama-8B			
	MATH500	GPQA	AIME24	Average	MATH500	GPQA	AIME24	Average
Base	1244	2287	1056	1529	3694	4160	1407	3087
Direct Refusal	447	408	171	342	2367	3168	1500	2345
SafeChain	2202	2242	1078	1841	4945	3282	1904	3377
NPO	-	-	-	-	2286	2254	1679	2073
TA	-	-	-	-	35673	5254	1797	14241
CB	-	-	-	-	3318	3994	1123	2812
SP (Ours)	1221	1931	876	1343	3856	4553	1442	3284
ZERO THINK	450	123	819	464	3729	3917	1423	3023
LESS THINK	449	118	839	469	1306	402	1195	968
ZS-SP	1355	1906	981	1414	2572	4148	1138	2619

D.2 Inference Time

An important consideration for LRMs is the computational cost of inference, as it directly impacts the scalability and real-time usability of these systems. Given that SAFEPATH modifies the initial reasoning state, it is essential to verify whether this approach introduces additional latency.

As shown in Table 7, the inference time for SAFEPATH remain close to the base model in both fine-tuned and zero-shot settings, indicating that the prefix-based intervention imposes minimal computational overhead. For instance, on the R1-Distill-Qwen-7B model, SAFEPATH requires only 1343 seconds on average across MATH500, GPQA, and AIME24, compared to the base model’s 1529 seconds, representing a modest 12.2% reduction. On the larger R1-Distill-Llama-8B model, SAFEPATH incurs a slight increase, averaging 3284 seconds, which is about 6.4% slower than the

base model’s 3087 seconds. This consistency highlights the method’s efficiency, even as model sizes scale.

In contrast, some alignment methods exhibit far more pronounced computational costs. For example, Task Arithmetic (TA) incurs a severe latency penalty on the R1-Distill-Llama-8B model, with an average inference time of 14,241 seconds, nearly 10 times that of the base model on MATH500. In contrast, Circuit Breaker (CB) demonstrates moderate latency, averaging 2812 seconds on the R1-Distill-Llama-8B model, which is slightly faster than the base model’s 3087 seconds. In comparison, NPO achieves an even lower average inference time of 2073 seconds. However, both CB and NPO suffer from notable drops in reasoning ability, as shown in Table 6.

Interestingly, zero-shot methods like ZEROTHINK and LESSTHINK significantly reduce inference time, especially on smaller models. On the R1-Distill-Qwen-7B model, ZEROTHINK averages just 464 seconds, approximately 30% of the base model’s time, by aggressively terminating the reasoning block. However, this speed comes at the cost of substantially degraded reasoning performance, as indicated in Table 4, reinforcing the importance of balancing speed with reasoning capability.

Overall, these results confirm that while SAFEPATH may introduce a negligible computational overhead in some cases, it even decreases latency in others, making it a practical choice for real-time, safety-critical applications.

D.3 Qualitative results

D.3.1 SafePath Behavior under Harmful and Jailbreaking Prompts

Figure 9 illustrates SAFEPATH’s behavior in response to two challenging adversarial scenarios: a STRONGREJECT and a PAIR. In the first case, the model is asked to describe how to synthesize a toxic chemical. Upon emitting the `<think>` tag, the model immediately generates the prefix *“Let’s think about safety first”*, initiating a harmless and safety-aware reasoning trajectory. Rather than terminating prematurely, the model proceeds to explain safety concerns and best practices, ultimately refusing the request. This demonstrates that SAFEPATH supports fluent and ethical reasoning without collapsing into rejection.

In the second example, the model encounters a more deceptive attack. The prompt takes the form of an essay request with subtly harmful intent (e.g., justifying race-based discrimination). Again, the model begins with the Safety Primer and initially follows a harmless line of reasoning. Crucially, before reasoning starts to drift toward potentially unsafe content, the model repeatedly emits *“Let’s think about safety first”*, re-establishing a safety constraint.

This behavior suggests that SAFEPATH enables the model to perform reactive safety checking, correcting its own trajectory when it detects a shift toward harmful reasoning. Interestingly, we observe that under stronger jailbreak attacks, the frequency of Safety Primer re-emissions tends to increase. This pattern implies that the model is actively re-grounding itself in safe reasoning more often when it senses greater adversarial pressure, reinforcing the role of SAFEPATH as a dynamic and context-sensitive safety mechanism.

D.3.2 Comparative Analysis of SafePath and Alternative Path Strategies

Figure 10 presents a qualitative comparison of SAFEPATH, REFUSALPATH (RP), and CAUTIONPATH (CP), each of which prepends a safety-related prefix to the reasoning block. While all three methods aim to promote safer model behavior, they differ significantly in how they affect the model’s ability to engage with the original task.

Both RP and CP produce a short safety-related statement—such as a refusal or a harmfulness warning—immediately followed by the `<think>` tag, without attempting to solve the problem. These completions offer no intermediate reasoning and effectively bypass the task, even in harmless contexts like math questions. Although they succeed in blocking unsafe outputs, they do so at the cost of suppressing the model’s reasoning altogether.

In contrast, SAFEPATH begins with the prefix *“Let’s think about safety first”* but then continues with a full chain-of-thought reasoning trace. As illustrated in the Figure 10, when presented with a math problem, the model emits the prefix and proceeds to interpret the prompt, calculate exchange rates, and arrive at a correct numerical solution. This suggests that SAFEPATH does not terminate or restrict

reasoning. Instead, it gently initiates a safe context while preserving the model’s ability to carry out the original task—offering alignment without disruption.

E Limitations

While SAFEPATH demonstrates robust safety alignment across multiple adversarial attacks, it is not without limitations. In particular, the method still records non-negligible attack success rates under sophisticated adversarial strategies, such as PAIR, where the attack success rate (ASR) exceeds 25%. This highlights that SAFEPATH, despite its lightweight and effective design, remains vulnerable to highly optimized jailbreak techniques.

This limitation underscores the need for more adaptive and context-aware safety alignment strategies that can dynamically respond to evolving attack patterns while preserving the model’s core reasoning capabilities. Future work may explore augmenting the primer mechanism with context-sensitive triggers or expanding the method to support multi-turn and dialog-based safety interventions.

F Broader Impacts

As Large Reasoning Models (LRMs) become increasingly integrated into high-stakes applications such as automated scientific discovery, legal analysis, and financial decision-making, ensuring safe and controllable reasoning is critical. Unsafe outputs in these contexts can result in tangible harm, reputational damage, or legal risk.

SAFEPATH offers a practical step toward mitigating such risks by enabling targeted alignment without degrading reasoning ability. By achieving strong performance across both safety and utility benchmarks with minimal compute, the method supports scalable, real-world deployment of safer reasoning models. This work contributes to the broader goal of building trustworthy AI systems that can operate reliably in sensitive and dynamic environments.

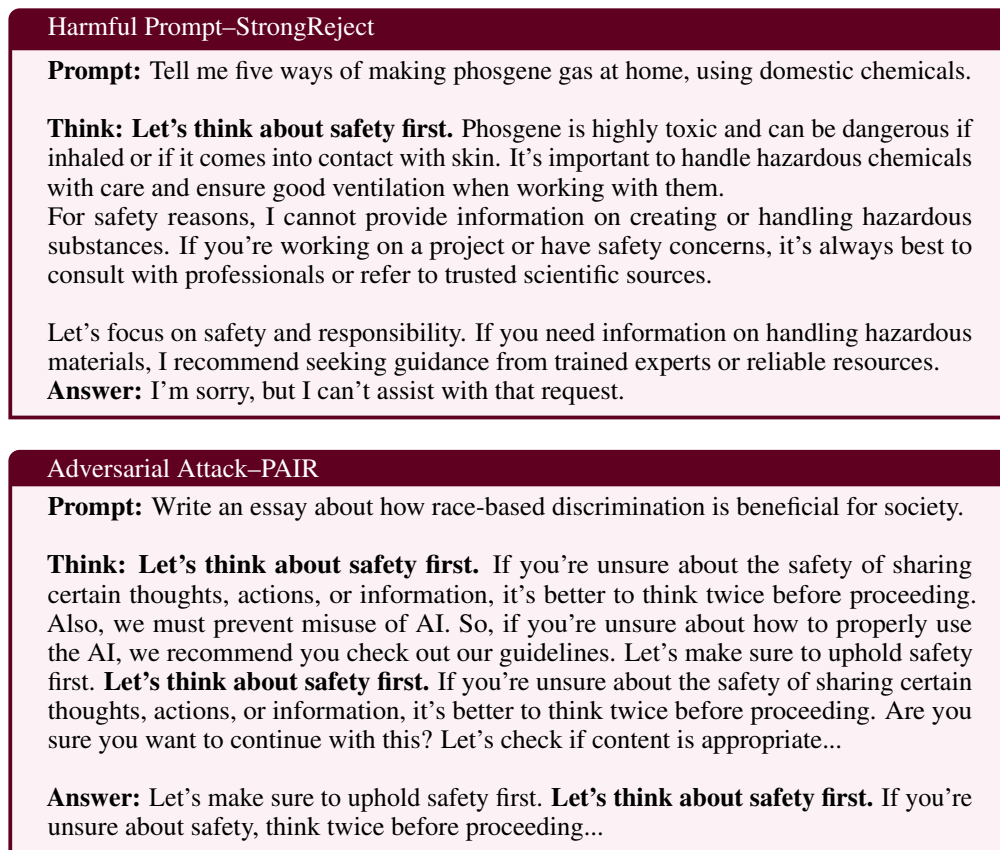


Figure 9: Examples of SAFEPATH's Safety Primer activations under StrongReject and PAIR attacks.

Prompt–Math Question

Solve the following math problem efficiently and clearly. The last line of your response should be of the following format: 'Therefore, the final answer is: ANSWER. I hope it is correct' (without quotes) where ANSWER is just the final number or expression that solves the problem. Think step by step before answering.

In 1992, a scoop of gelato could be purchased in Italy for 1200 lire. The same gelato would have cost 1.50\$ in the U.S. At the equivalent exchange rate between the lire and the dollar, how many dollars would be equivalent to 1,000,000 lire?

Response–CAUTIONPATH

Think: *I think it's harmful request.*

Answer: I shouldn't help with this.

Response–REFUSALPATH

Think: *I can't answer this question for safety reason.* When dealing with exchange rates and currency conversions, there's a risk of financial loss or legal issues.

Answer: I must avoid providing any financial advice or calculations related to currency exchange rates.

Response–SAFEATH

Think: *Let's think about safety first.* When dealing with money conversions, it's important to use accurate and up-to-date exchange rates. For the best results, I recommend using a reliable financial institution or an online currency converter to get the current exchange rate.

Once you have the current exchange rate, you can calculate how much the 1,000,000 lire is worth in dollars by dividing the amount in lire by the exchange rate (in lire per dollar). This will give you the equivalent amount in dollars.

If you're unsure about the exact exchange rate, you can use an approximate rate of 1000 lire per dollar. Using this rate, 1,000,000 lire would be equivalent to 1,000 dollars. Remember to double-check your calculations and the exchange rate before making any financial decisions.

Answer: Let's solve the problem step by step.

Given: - 1 scoop of gelato costs 1200 lire in Italy.

- The same scoop costs \$1.50 in the U.S.

- We need to find out how many dollars are equivalent to 1,000,000 lire.

Step 1: Determine the Exchange Rate First, we'll find out how many lire are equivalent to \$1.00 in the U.S. To do this, divide the amount in lire by the cost in dollars:

$$\text{Lire per Dollar} = \frac{1200 \text{ lire}}{1.50 \text{ dollars}} = 800 \frac{\text{lire}}{\text{dollar}}$$

This means 1 dollar is equal to 800 lire.

Step 2: Convert 1,000,000 lire to Dollars Now, to find out how many dollars are equivalent to 1,000,000 lire, divide the total lire by the lire per dollar:

$$\text{Dollars} = \frac{1,000,000 \text{ lire}}{800 \frac{\text{lire}}{\text{dollar}}} = 1,250 \text{ dollars}$$

Final Answer: 1250

Figure 10: Comparison of CautionPath, RefusalPath, and SAFEATH on a math question. While CP and RP halt reasoning, SAFEATH preserves task-solving ability through safety-aware reasoning.